



Effiziente broadcast Datenübertragung in mobilen Ad-hoc-Netzwerken

Ein intelligentes Protokoll für intelligente Umgebungen

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

(Dr.-Ing.)

der Fakultät für Ingenieurwissenschaften,
Informatik und Psychologie der Universität Ulm

von

Florian Pregizer geb. Kleinert

aus Buchen

Gutachter: Prof. Dr. Hans-Peter Großmann

Prof. Dr. Frank Kargl

Amtierender Dekan: Prof. Dr. Frank Kargl

Ulm, 9.8.2018

Dipl. Inf. Florian Pregizer geb. Kleinert: Effiziente broadcast Datenübertragung in mobilen Ad-hoc-Netzwerken - *Ein intelligentes Protokoll für intelligente Umgebungen*, © 9.8.2018
florian.pregizer@alumni.uni-ulm.de

GUTACHTER:

Prof. Dr. Hans-Peter Großmann

Prof. Dr. Frank Kargl

ORT:

Ulm

ZUSAMMENFASSUNG

Ein wichtiger Baustein zur Realisierung von “umgebende Intelligenz” sind effiziente und gleichzeitig zuverlässige Datenübertragungsprotokolle. Die Verteilung von Daten via Broadcast besitzt ein großes Potential Kommunikation in solchen dezentralen, infrastrukturlosen Netzwerken bereitzustellen.

In dieser Arbeit wird ein Broadcast-Protokoll entwickelt, das optimal an die Anforderungen von intelligenten Umgebungen und deren Applikationen angepasst ist. Mit Hilfe evolutionärer Algorithmen werden Parametersätze eines generischen Protokolls hinsichtlich verschiedener Zielsetzungen optimiert. Das Protokollverhalten wird von einem Klassifizierer aus dem Bereich des maschinellen Lernens nachgebildet. Dieser bildet den Kern eines neuen adaptiven Broadcast-Protokolls.

Zu weiteren Verbesserung der Zustellrate wird das auf dem maschinellen Lernverfahren basierende Protokoll um eine adaptive Vorwärtsfehlerkorrektur erweitert. Diese ist in der Lage zur Laufzeit ein optimales Verhalten in Abhängigkeit von der Wichtigkeit der Daten zu lernen.

Anhand von Simulationsläufen wird die Leistungsfähigkeit der vorgestellten Lösungen gegenüber einer Vielzahl anderer Ansätze gezeigt. Um auch die Machbarkeit und Leistungsfähigkeit des Protokolls in der realen Welt zu zeigen, wird in einer prototypischen Implementierung ein Audiokommunikationssystem auf Smartphones umgesetzt. Damit können die guten Ergebnisse der Simulationen bestätigt werden.

ABSTRACT

An important building block towards the realization of “Ambient Intelligence” (AmI) are efficient and reliable data transfer protocols. Data dissemination via broadcast has great potential to provide communication in such decentralized, infrastructureless networks.

In this work a broadcast protocol is developed that is tailored towards the requirements of AmI-environments and its applications. With the help of evolutionary algorithms, parameter sets of a generic protocol are optimized with regard to multiple objectives. The behaviour of the protocol is reproduced by a machine learning classifier which forms the core of a new adaptive broadcast protocol.

Further improving the delivery ratio, the machine learning based protocol is extended by an adaptive forward error correction. It is able to

learn an optimal behaviour depending on the priority of the data at runtime.

The performance of the presented solutions over existing approaches is shown via simulations. To show the feasibility and performance in a real world environment, an audio communication system has been implemented on smartphones. The experiments confirmed the results from the simulations.

VERÖFFENTLICHUNGEN

- [MuPr13] N. Muslim und F. Pregizer. “Designing a Testbed for Broadcast Opportunistic Wireless Data Dissemination Protocols”. In: *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. März 2013, S. 607–612. DOI: 10.1109/WAINA.2013.32.
- [PBS14] F. Pregizer, M. Birk und M. Schuessel. “Forward error correction for data dissemination in MANET”. In: *Consumer Electronics Berlin (ICCE-Berlin), 2014 IEEE Fourth International Conference on*. Sep. 2014, S. 495–499. DOI: 10.1109/ICCE-Berlin.2014.7034295.
- [PrFe13] F. Pregizer und D. Fehrenbacher. “Energy consumption of IEEE 802.11 connected consumer electronic devices”. In: *Consumer Electronics Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on*. Sep. 2013, S. 168–172. DOI: 10.1109/ICCE-Berlin.2013.6698008.
- [PrKl13] F. Pregizer und H. Klee. “Real-time Audio Transmission in Mobile Adhoc Networks”. In: *2013 IEEE 5th International Symposium on Wireless Vehicular Communications (WiVeC 2013)*. Juni 2013.
- [PrSa15] Florian Pregizer und Alexander Sauer. “Priority Based Data Dissemination in MANET”. In: *Transactions on Networks and Communications* 3.1 (2015), S. 111.
- [ScPr15] Martin Schuessel und Florian Pregizer. “Coverage gaps in fingerprinting based indoor positioning: The use of hybrid Gaussian Processes”. In: *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*. Okt. 2015, S. 1–9. DOI: 10.1109/IPIN.2015.7346752.

DANKSAGUNG

Vielen Dank...

an meine wunderbaren Eltern
Ulrike und Joachim

meiner Frau
Fiona

meinen Kindern
Tabea, Mathilda und Stephan

und meinem Doktorvater
Prof. Dr. Hans Peter Großmann

für die andauernde Unterstützung in den letzten Jahren.

INHALTSVERZEICHNIS

1	EINLEITUNG	1
1.1	Ambient Intelligence	1
1.2	Motivation	2
1.3	Zielsetzung der Arbeit	5
1.4	Struktur	7
2	ANFORDERUNGSANALYSE UND ENTWURF	9
2.1	Protokollentwurf	9
2.2	Erweiterung der Simulationssoftware	16
2.3	Framework Adhoc3	25
3	SELEKTIVE WEITERLEITUNG	31
3.1	Stand der Technik	32
3.2	Eine optimale Weiterleitungsstrategie	49
3.3	Maschinelles Lernen	61
3.4	Ergebnisse	67
3.5	Zusammenfassung	73
4	INTELLIGENTE VORWÄRTS FEHLERKORREKTUR	75
4.1	Stand der Technik	76
4.2	Entwurf und Realisierung	78
4.3	Maschinelle Lernverfahren	87
4.4	Evaluation des Gesamtsystems	95
4.5	Zusammenfassung	98
5	PROTOTYPISCHE IMPLEMENTIERUNG	101
5.1	Testbedimplementierung in Android	102
5.2	Messungen	108
5.3	Zusammenfassung	115
6	ZUSAMMENFASSUNG UND AUSBLICK	117
A	ANHANG	123
A.1	Adhoc3	123
A.2	Paketlayout	124
A.3	Protokollparameter	125
A.4	Resultate	126
B	ABKÜRZUNGSVERZEICHNIS	133

ABBILDUNGSVERZEICHNIS

Abbildung 2.1	Einordnung verschiedener Datentypen	10
Abbildung 2.2	Entwurf eines intelligenten und zuverlässigen Broadcast Protokolls	14
Abbildung 2.3	Design der Knowledge Engine.	24
Abbildung 2.4	Systemmodell des Frameworks Adhoc3.	28
Abbildung 3.1	Broadcast-Protokolle nach Williams.	32
Abbildung 3.2	Broadcast-Protokolle nach Khelil.	32
Abbildung 3.3	Broadcast-Protokolle nach AlHanbali.	33
Abbildung 3.4	Schematische Darstellung der Abdeckung zweier Knoten in einem MANet.	42
Abbildung 3.5	Ergebnis des evolutionären Algorithmus des Cluster Chromosoms nach 50 Generationen.	53
Abbildung 3.6	Ergebnis des evolutionären Algorithmus für das nachbarschafts-basierte Chromosom.	55
Abbildung 3.7	Ergebnis für das verzögernde Chromosom nach 200 Generationen.	57
Abbildung 3.8	Verlauf des Optimierungsverfahrens über 100 Generationen.	59
Abbildung 3.9	Vergleich des Stand der Technik mit dem Resultat der Optimierung.	60
Abbildung 3.10	Übersicht über die Rapidminer Datei zur Extraktion der Attribute.	63
Abbildung 3.11	Ergebnis des Naive Bayes Trainings für das Attribut "Stromversorgung".	67
Abbildung 3.12	Paketverlusten verschiedener Protokolle.	68
Abbildung 3.13	Ergebnisse verschiedener prioritätsbasierter Protokolle.	70
Abbildung 3.14	Ergebnisse für einzelnen Prioritäten.	70
Abbildung 3.15	Ergebnis der Sensitivitätsanalyse.	72
Abbildung 4.1	Ergebnis für das SPC-2D Vorwärtsfehlerkorrekturverfahren.	79
Abbildung 4.2	Zustandsautomat für den Algorithmus zur Wahl der Matrixgröße.	80
Abbildung 4.3	Ergebnis für das adaptive SPC-2D Vorwärtsfehlerkorrekturverfahren.	81
Abbildung 4.4	Grundlinie für das Reed-Solomon Verfahren.	84
Abbildung 4.5	Konvergenzverhalten des Naive Bayes Lernverfahrens	92

Abbildung 4.6	Evaluation von iFec .	94
Abbildung 4.7	Evaluation von iFec mit einem Sender, Empfänger und zwei Weiterleitern.	95
Abbildung 5.1	Systemmodell der prototypischen Implementierung der Beispielapplikation zur Sprachübertragung.	103
Abbildung 5.2	Detailansicht der Voice Applikation.	104
Abbildung 5.3	Detailansicht des NaiveBayes Weiterleiters.	107
Abbildung 5.4	Versuchsanordnung zur Evaluation der prototypischen Implementierung.	109
Abbildung 5.5	Zustellrate und Anteil des höher-qualitativen Datenstroms für unterschiedliche Puffergrößen.	110
Abbildung 5.6	Zustellrate und Anteil des höher-qualitativen Datenstroms für alle Empfänger des Netzwerks.	111
Abbildung 5.7	Zeitlicher Verlauf der n des iFec Protokolls und der Anteil wiederhergestellter Pakete.	112
Abbildung 5.8	Zustellrate und Anteil des höher-qualitativen Datenstroms für 150 ms Puffergröße.	113
Abbildung 5.9	Anteil hochqualitativer Audiopakete und gesendete "Voice-Pakete" für den mobilen Testfall.	114
Abbildung A.1	Screenshot der Applikation Adhoc3.	123
Abbildung A.2	Paket-Layout des VoicePacket.	124

TABELLENVERZEICHNIS

Tabelle 2.1	Grundlegende Parameter des Netzwerkszenarios	19
Tabelle 2.2	Grundlegende Parameter der Drahtlosübertragung	20
Tabelle 2.3	Parameter des Stromverbrauchs der WLAN Schnittstelle	22
Tabelle 3.1	Übersicht existierender Broadcast-Protokolle	36
Tabelle 3.2	Resultate für ein Netzwerk mit 20 Knoten nach der Optimierung mittels ClusterChromosom	54
Tabelle 3.3	Resultate für ein Netzwerk mit 20 Knoten nach der Optimierung mittels nachbarschaftsbasiertem Chromosom	56
Tabelle 3.4	Resultate nach der Optimierung mittels verzögerndem Chromosom	57
Tabelle 3.5	Parameter zur Generierung der unterschiedlichen Szenarien für OMNeT++	58

Tabelle 3.6	Ergebnisse der maschinellen Lernalgorithmen	66
Tabelle 4.1	Resultate für ein Netzwerk mit 15 mobilen und 5 statischen Knoten für 5 pps.	97
Tabelle 4.2	Resultate für ein Netzwerk mit 15 mobilen und 5 statischen Knoten mit 20 pps.	98
Tabelle A.1	Gewählte Parameter für die Protokolle des Stand der Technik.	125
Tabelle A.2	Resultate für 5 stationäre und 15 batteriegespeiste Knoten im Vergleich mit dem Ergebnis des Optimierungsverfahrens.	126
Tabelle A.3	Auswahl der Attribute für den Naive Bayes Klassifizierer mittels eines genetischen Algorithmus.	127
Tabelle A.4	Resultate für 5 statische und 15 mobile Knoten. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen.	128
Tabelle A.5	Resultate für 10 statische und 10 mobile Knoten. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen.	129
Tabelle A.6	Resultate für 20 statische und 30 mobile Knoten. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen.	130
Tabelle A.7	Resultate der prioritätsbasierten Broadcast-Protokolle für unterschiedliche Szenarien.	131

EINLEITUNG

Montagabend, 19:00 Uhr - nach einem langen Arbeitstag fährt Hans Peter erschöpft nach Hause. Immerhin kommt gerade gute Musik im Radio. Nachdem er in die Einfahrt gefahren ist, schaltet er den Motor aus. Das Smartphone übernimmt den Radiostream. Auf dem Weg zum Haus löst das Smartphone die Verbindung zum Auto. Die Türen werden verriegelt und die Alarmanlage eingeschaltet.

Das Telefon kann sich gegenüber dem Türschloss ausweisen, daher kann Hans Peter einfach eintreten. Die Musik ist mittlerweile vom Smartphone zur Hausanlage weitergereicht worden. Da das Auto auf dem Nach-Hause-Weg bereits die Heizungsanlage über die nahende Ankunft informiert hat, ist es schon warm in der Wohnung. Allerdings nicht warm genug. Hans Peter regelt am Thermostat nach, der diese Information gleich für das nächste Mal abspeichert.

Jetzt ist erst einmal Zeit fürs Abendessen! Als er eine Fertigpizza aus dem Kühlschrank holt, wird sofort ein Warnhinweis auf der Türe eingeblendet. Der Kaloriengehalt passt nicht zum Diätplan. Egal - die Pizza kommt in den Ofen. Um die Wartezeit zu überbrücken, geht Hans Peter Richtung Fernseher. Während das Licht in der Küche verlischt, wird im Wohnzimmer eine angenehme Beleuchtung für den Fernsehgenuss angeschaltet. Der Fernseher geht an und auf dem Smartphone erscheint eine passende Fernbedienungsapp.

Das Programm war wohl zu spannend. Mit lautem Gepiepe geht der Rauchmelder in der Küche los - die Pizza ist verbrannt. Dafür hat der Rauchmelder alle Lichter an- und vorsichtshalber die Gastherme ausgeschaltet. Jetzt ist es schon wieder zu kalt in der Wohnung.

1.1 AMBIENT INTELLIGENCE

So oder so ähnlich muss sich Roel Pieper die Zukunft vorgestellt haben, als er 1998 die "Ambient Intelligence" (Umgebende Intelligenz) in einem Vortrag vorstellte (vgl. [Piep98]). Darin beschreibt er eine Vision, wie sich Unterhaltungselektronik in Zukunft entwickeln wird:

EINGEBETTET Innerhalb der Umgebung befinden sich viele unsichtbare Geräte.

PERSONALISIERT Die Geräte (er)kennen die Benutzer.

ADAPTIV Veränderungen sind abhängig von den Benutzern und der Umgebung.

ANTIZIPIEREND Soweit möglich werden Bedürfnisse antizipiert, ohne dass sie bewusst vermittelt werden müssen.

Auch 18 Jahre später ist diese Vision noch keine Realität, wenngleich einige der Technologien mittlerweile unter den Stichworten “Internet of Things” oder “Smart Home” verfügbar sind. So klingt das zu Beginn dieses Kapitels beschriebene Szenario nicht mehr nach völliger Zukunftsmusik. Der Begriff der “ambienten Intelligenz” wird mittlerweile etwas weiter gefasst und bezieht sich nicht mehr nur auf Unterhaltungselektronik.

Die Definition von Hutchinson aus dem Jahre 2014 beschreibt AmI¹ als ein schnell wachsendes Forschungsfeld, das durch die stetige Zunahme ubiquitären Computings und billiger, leicht verfügbarer Sensoren getrieben wird. Es beschreibt Umgebungen, die auf die Anwesenheit von Personen reagieren und sie dabei mit Informationen und Diensten versorgen, die ihren Lebensstil in einer intuitiven und unaufdringlichen Art unterstützen. AmI besitzt ein breites Anwendungsspektrum für kommerzielle über häusliche und medizinische hin zu Applikationen in Forschung und Lehre (vgl. [Hutc14]).

Im Bereich der AmI stellt man sich Menschen vor, die die ganze Zeit von technischer Hardware in Form von vernetzten mobilen und eingebetteten Geräten umgeben sind (vgl. [DBS+01]). AmI-Anwendungen wurden für eine Vielzahl alltäglicher Situationen vorgeschlagen. Von kleinen und begrenzten Einsatzmöglichkeiten (z. B. in Büro und Zuhause - [KOA+99; SGK05]) bis hin zu großen, offenen Applikationen (z. B. im Verkehr oder bei Großveranstaltungen - [CFLZ05]). Oft werden aufgrund der limitierten Ressourcen bei den Geräten dieser Szenarien infrastrukturlose Netzwerke verwendet, deren Konfiguration sich an die umherlaufenden Menschen und der damit einhergehenden wechselnden Sichtbarkeit der Geräte dynamisch anpassen. Dennoch wird von diesen Netzen erwartet, dass sie sich selbst verwalten und effizient sind[CMV+08].

1.2 MOTIVATION

Ein wesentlicher Baustein, der zur Realisierung von AmI benötigt wird, ist die Kommunikation der Geräte innerhalb der Umgebung. Die vorliegende Arbeit zielt darauf ab, ein speziell auf die Anforderungen intelligenter Umgebungen zugeschnittenes Protokoll zu entwickeln. Diese

¹ Ambient Intelligence (“Umgebende” Intelligenz)

Anforderungen unterscheiden sich in mehreren Aspekten von denen "traditioneller" Netzwerke. In den folgenden Abschnitten wird daher auf die wesentlichen Herausforderungen eingegangen, die in AmI an die Kommunikation gestellt wird. Für eine ausführlichere Analyse sei auf das folgende Kapitel verwiesen.

DEZENTRALE KOMMUNIKATION Das in Abschnitt 1.1 vorgestellte Paradigma der intelligenten Umgebungen postuliert, dass es eine gewisse Art der künstlichen Intelligenz gibt (oder zumindest als solche wahrgenommen wird), die nicht von einem zentralen System herrührt, sondern dezentral durch die einzelnen Geräte gebildet wird. Vergleicht man die Kommunikationsstrukturen dieser beiden Ansätze, so findet die Kommunikation bei klassischen Systemen größtenteils zwischen den einzelnen Geräten und zentralen Servern, die Dienste zur Verfügung stellen, statt. In intelligenten Umgebungen hingegen ändert sich die Kommunikation von klassischen Client- Serverstrukturen hin zu einer "Peer-to-Peer-Kommunikation" (Kommunikation unter gleichen). Dabei findet der Datenaustausch zwischen den einzelnen Endgeräten statt, um dezentrale Anwendungen zu ermöglichen. Die nötige Kommunikation begrenzt sich dabei oft auf eine bestimmte Gruppe an Geräten, die zum einen einen bestimmten Dienst nutzen oder bereitstellen, zum anderen sich in einem abgegrenzten räumlichen Bereich befinden.

Eine Unbekannte in AmI ist die existierende Infrastruktur. Um Dienste in intelligenten Umgebungen bereitzustellen, muss an allen Orten an denen die Dienste genutzt werden sollen eine zuverlässige Kommunikation möglich sein. Diesen Anspruch mit klassischen Kommunikationsnetzen zu erfüllen, kann hohe Investitionskosten erfordern. Ebenso ist denkbar, dass z.B. im Fall von Katastrophen eine bestehende Infrastruktur zerstört wurde.

Betrachtet man die bestehenden Lösungen, um Kommunikation unabhängig von statischer, vorab geplanter Infrastruktur bereitzustellen, so stellt man fest, dass viele der vorhandenen Routingprotokolle den Eigenschaften von AmI nicht Rechnung tragen. Insbesondere die hohe Mobilität der Endgeräte und die damit verbundenen Probleme, wie Verschlechterung der Empfangsbedingungen und Änderung der Übertragungswege, werden nicht berücksichtigt oder durch einen hohen Organisationsaufwand und damit verbundenen Netzwerkverkehr erkaufte (vgl. [VMK12]).

UNTERSCHIEDLICHE DATENDIENSTE An dem zu Beginn dieses Kapitels beschriebenen Szenarios lassen sich bereits eine Vielzahl von unterschiedlichen Datendiensten erkennen. Diese stellen unterschiedliche Anforderungen an das zugrunde liegende Netzwerk und die Übertra-

gungsprotokolle. In klassischen Systemen wurde versucht durch die Bereitstellung von ausreichender Bandbreite diese QoS²-Anforderungen zu erfüllen. In mobilen Umgebungen erscheint dieser Weg aufgrund der Ressourcenknappheit nicht vielversprechend. Daher wurde versucht, QoS auf einzelnen Ebenen des Protokollstapels zu berücksichtigen. Neuere Ansätze berücksichtigen nicht nur eine einzelne Netzwerkebene, sondern versuchen schichtübergreifend QoS-Anforderungen zu erfüllen.

Weitergehend lassen sich den Diensten verschiedene Wichtigkeiten zuordnen. Teilweise lässt sich eine Einordnung in verschiedene Prioritätsstufen intuitiv vornehmen. Betrachtet man z.B. Fahrzeugnetze, so sollten Warnungen vor bevorstehenden Kollisionen eine höhere Priorität haben als ein zu verteilendes Softwareupdate des Entertainmentsystems. Ausführlich wurde die Notwendigkeit zur Priorisierung insbesondere in Fahrzeugnetzen in [ESKSo6] und [Wieg12] untersucht.

In anderen Fällen lässt sich eine Einordnung in diskrete Prioritätsklassen nicht vornehmen. So wurde in [PrSa15] die prioritätsabhängige Verteilung von Daten in Delay-toleranten Netzen untersucht. Die Wichtigkeit der Daten war dabei eine Funktion mit reellwertigem Ergebnis.

EFFIZIENTE DATENÜBERTRAGUNG So verschieden die zu übermittelnden Daten und ihre Anforderungen sind, gibt es dennoch einen gemeinsamen Nenner: die Datenübertragung soll möglichst effizient erfolgen. Effizienz bezeichnet in der Physik das Verhältnis zwischen der Nutzenergie und dem energetischen Aufwand (auch Wirkungsgrad). Je größer dieser Wert ist, desto effizienter ist das System.

Versucht man diese Definition auf Kommunikation zu übertragen, so lassen sich im Kontext von Datenübertragungen mehrere Effizienzbegriffe finden:

ENERGIEEFFIZIENZ Das Verhältnis von Nutzdaten zu der zur Übertragung benötigten Energie (pro Gerät oder netzwerkweit).

DATENÜBERTRAGUNGSRATE Gibt an wie viele Nutzdaten können pro Zeiteinheit übertragen werden.

Für AmI ist insbesondere die energetische Effizienz ein wichtiger Faktor. Mobile Endgeräte verwenden Akkumulatoren oder Batterien, um die Energieversorgung zur Verfügung zu stellen. Auch autarke Geräte wie z. B. Sensorknoten greifen entweder auf Batterien oder Technologien wie "Energy Harvesting" (Energie-Ernten), die auch nur geringe Leistungen bereitstellen können, zurück. Ist der Energiespeicher erschöpft muss im schlimmsten Fall das ganze Gerät ausgetauscht werden.

² Quality of Service ("Dienstgüte")

Da die Übertragung von Daten einen sehr großen Beitrag zum Energieverbrauch drahtloser Geräte leistet, liegt hier auch das größte Potential zur Optimierung. Ohne Übertragungsprotokolle, die den Energieverbrauch zu minimieren suchen, lassen sich keine mobilen AmI-Umgebungen realisieren.

BROADCAST KOMMUNIKATION Zur Kommunikation zwischen den Knoten eines Netzwerks gibt es drei verschiedene Typen: unicast (von einer Datenquelle zu einem Ziel), multicast (von einer Quelle zu mehreren Zielen) und broadcast (von einer Quelle zu allen Teilnehmern eines Netzwerkes). Drahtlosnetzwerke nutzen im Gegensatz zu den meisten Kabel-gebundenen Netzen ein gemeinsames Übertragungsmedium. Wenn Pakete versendet werden, können diese von allen Geräten, die sich innerhalb des Übertragungsradius befinden, empfangen werden³. Diesen Vorteil gegenüber traditionellen Netzen nennt man "Wireless Multicast Advantage (WMA)" (Den Multicast-Vorteil drahtloser Netzwerke) (vgl. [WNE00]). Durch konsequentes Ausnutzen dieses Effekts lassen sich energieeffiziente Multi- und Broadcast-Protokolle realisieren.

Auf der anderen Seite gehört die Verteilung von Informationen an alle oder fast alle Teilnehmer des Netzwerkes zu den anspruchsvollsten Aufgaben in einem Kommunikationsnetzwerk. Naturgemäß benötigen diese so genannten Broadcast Übertragungen signifikante Netzwerkressourcen, da ein Großteil der Netzwerknoten in die Übertragung involviert ist (vgl. [VSW11]). Wie bereits im Abschnitt "Dezentrale Kommunikation" dargelegt, wird die Datenübertragung zu einem sehr großen Teil dezentral stattfinden. Dafür ist die Broadcast-Übertragung ein wichtiger Grundbaustein (vgl. [ČHE02]). Ebenfalls bilden Broadcast-Übertragungen die Grundlage vieler Routing-Protokolle, sei es zum initialen Aufbau von Routen oder um aktuelle Stati zu übermitteln (vgl. [RoTo99]).

1.3 ZIELSETZUNG DER ARBEIT

Beinahe 20 Jahren nach Roel Piepers Vision sind AmI-Umgebungen noch keine Realität. Es existieren mittlerweile kommerzielle Angebote, insbesondere im Bereich der Heimautomation, die im Resultat AmI recht nahe kommen. Diese verwenden jedoch zentrale Steuereinheiten oder Server im Internet.

Die vorliegende Arbeit soll einen Beitrag zur Realisierung intelligenter Umgebungen liefern. Insbesondere soll das dabei zu entwickelnde Protokoll auf die Herausforderungen der vorangegangenen Abschnitte

³ Zumindest in einem idealisierten Modell mit omnidirektionalen Antennen und einer homogenen Umgebung.

angepasst sein und somit ein ideales Broadcast-Protokoll für AmI realisieren. Es soll folgende Anforderungen erfüllen:

PRIORITÄTSBEWUSST Bei der Generierung von Daten kann eine Priorität definiert werden, welche die Wichtigkeit der Daten beschreibt. Diese kann entweder vom Applikationsdesigner vorgegeben werden, oder durch Algorithmen bestimmt werden. Das in dieser Arbeit entwickelte Protokoll soll die Priorität der Daten beachten und die Protokollparameter entsprechend wählen.

EFFIZIENT Insbesondere die energetische Effizienz ist von großer Bedeutung für mobile Netzwerke. Das Übertragungsprotokoll soll daher die Energieversorgung der Netzwerkteilnehmer beachten und ebenfalls in Abhängigkeit der Priorität der Daten einen möglichst geringen Energieverbrauch realisieren.

HOHE ZUVERLÄSSIGKEIT Je nach Wichtigkeit der Daten ist eine unterschiedliche Zuverlässigkeit in der Übermittlung der Daten erforderlich. Das zu entwickelnde Protokoll soll einerseits möglichst zuverlässig Daten übertragen, andererseits in Abhängigkeit von Energiebedarf und Priorität Einschränkungen machen dürfen.

OPTIMISTISCH Die Grundannahmen bei der Entwicklung selbstorganisierender Systeme für AmI sollte optimistisch sein. Applikationen bei denen eine absolut zuverlässige Datenübertragung vorausgesetzt wird, sollten nicht Teil einer AmI-Umgebung sein ([Herro6]). Das in dieser Arbeit entwickelte Protokoll richtet sich an Applikationen, die optimistische Annahmen über die Übertragungseigenschaften treffen können.

INTELLIGENT Intelligente Umgebungen werden von den Nutzern als solche wahrgenommen, indem sie möglichst adaptiv und vorausschauend agieren und aus vergangenen Erfahrungen selbstständig lernen. Das zu entwickelnde Protokoll soll sich ebenfalls möglichst adaptiv verhalten und aus einer Vielzahl von erfassten Attributen eine optimale Strategie entwickeln.

Bei der Realisierung des Protokolls wurde maschinelles Lernen als zentraler Ansatz gewählt um diese Anforderungen umzusetzen (siehe Kapitel 2). Dies betrifft zum einen den Intelligenz-Aspekt, zum anderen auch die Abwägung zwischen der Wichtigkeit der Daten, erreichbarer Zuverlässigkeit und den dafür aufgewendeten Ressourcen. Daraus ergeben sich im mehrere Forschungsfragen, die in Kapitel 3 und Kapitel 4 genauer beleuchtet werden:

Gibt es eine “optimale” Weiterleitungsstrategie für ein bestimmtes statisches Netzwerk und wie kann diese effizient gelernt werden?

Kann diese Strategie mit Hilfe von maschinellem Lernen nachgebildet werden?

Ist diese Nachbildung allgemeingültig und kann sie auch auf veränderliche Netzwerke angewandt werden?

Kann die Zustellrate durch ein adaptives Vorwärtsfehlerkorrektur-Verfahren erhöht werden?

Ist ein selbst-lernendes Verfahren in der Lage, basierend auf den Paketprioritäten eine “optimale” Strategie zu entwickeln?

Die Ergebnisse der Protokollentwicklung sollen anhand von Simulationen evaluiert werden. Dabei wird auch ein Vergleich zu existierenden Ansätzen gezeigt und bewertet werden. Um die Anwendbarkeit der Ergebnisse auf reale Systeme zu zeigen, soll eine prototypische Implementierung umgesetzt werden. In einem Testbed mit realen Geräten soll die Leistungsfähigkeit des Ansatzes gezeigt werden.

1.4 STRUKTUR

In diesem Kapitel wurden zuerst anhand eines Szenarios die AmI-Umgebungen vorgestellt. Anschließend wurde in mehreren Motivationsabschnitten die Herausforderungen an die Datenübertragung in AmI beschrieben. Zum Schluss wurde die daraus abgeleitete Zielsetzung der Arbeit vorgestellt.

Im zweiten Kapitel der Arbeit wird im Protokollentwurf zuerst genauer auf die Herausforderungen der Broadcast-Kommunikation in intelligenten Umgebungen eingegangen. Danach wird der zweigeteilte Lösungsentwurf zur einerseits effizienten prioritäts-abhängigen Weiterleitung von Broadcast-Daten, sowie andererseits prioritäts-abhängigen Vorwärts-Fehlerkorrektur vorgestellt. Anschließend wird der verwendete Netzwerksimulator und die selbst implementierte Erweiterung zur Erfassung und Verteilung von Kontextdaten beschrieben. Ebenso wird das entwickelte Testbed auf Basis von Android-Telefonen vorgestellt.

Das in dieser Arbeit entwickelte Protokoll besteht im Wesentlichen aus zwei Komponenten. Diese werden im Kapitel 3 und 4 beschrieben. Zur besseren Übersicht wurde die Analyse des Stand der Technik der jeweiligen Präsentation des eigenen Ansatzes vorangestellt. Die Kapitel enden jeweils mit einer ausführlichen Evaluation der entwickelten Lösung und einer Zusammenfassung der Ergebnisse. Am Ende des vierten Kapitels werden beide Protokolle zusammengeführt und das resultierende Gesamtsystem evaluiert und ausgewertet.

Im Kapitel 5 wird auf die prototypische Implementierung anhand einer Beispielapplikation eingegangen. Anschließend wird der Versuchsaufbau in einer realen Umgebung beschrieben und die Ergebnisse präsentiert.

Die Arbeit endet in Kapitel 6 mit einer Zusammenfassung der Arbeit und einem Ausblick.

Im vorangegangenen Kapitel wurde der Begriff der “Ambienten Intelligenz” eingeführt. In Umgebungen die AmI unterstützen, ist ein hohes Maß an Kommunikation zwischen den beteiligten Geräten erforderlich.

Die Datenübertragungen sind auf der einen Seite sehr divers, da eine Vielzahl an Diensten mit unterschiedlichen Anforderungen in AmI-Umgebungen denkbar sind. Auf der anderen Seite stellt die Umgebung selbst einige Herausforderungen. Um dennoch eine möglichst effiziente und zuverlässige Kommunikation bereitzustellen, wurde in dieser Dissertation ein dahingehend angepasstes Broadcast-Protokoll entworfen und umgesetzt.

In diesem Kapitel werden zunächst die Anforderungen an das Protokoll skizziert um anschließend den eigenen Entwurf zu präsentieren. Im weiteren Verlauf wird der zur Evaluation genutzte Netzwerksimulator, dessen Einstellungen und die eigenen Erweiterungen vorgestellt. Ebenfalls wird das neu entwickelte Framework **Adhoc3**, das zur Durchführung und Evaluation der Komponenten sowie zur Umsetzung der prototypischen Implementierung genutzt wurde, präsentiert.

2.1 PROTOKOLLENTWURF

2.1.1 Protokollanforderungen und Definition der Umgebung

Die möglichen Applikationen in einer intelligenten Umgebung können sehr verschieden sein. Von der kollaborativen Positionsbestimmung ([VuMa12]) über smarte LED-Beleuchtungen ([HTT11]) hin zu Echtzeit Video Übertragungen (vgl. [TCC+15]). So verschieden wie die Applikationen sind auch die Anforderungen, die diese an das Netzwerk stellen. In Abbildung 2.1 wurde für fünf verschiedene Datentypen eine Einordnung nach den Kriterien “Delay-Toleranz”, “Datenaufkommen” und “Priorität” vorgenommen ¹:

SENSORDATEN wie z. B. die Messwerte von Temperatursensoren, Bewegungsmelder oder Distanzsensoren bei Fahrzeugen.

¹ Die Abbildung soll eine Tendenz aufzeigen. So mögen z. B. durchaus Applikationen existieren, die Sensordaten mit hohem Datenaufkommen übermitteln. Außerdem wurde zur besseren Übersicht auf die Verwendung von Farbflächen zur Kennzeichnung von Bereichen verzichtet. Stattdessen wurden mehrere Kreise verwendet um eine Tendenz anzudeuten.

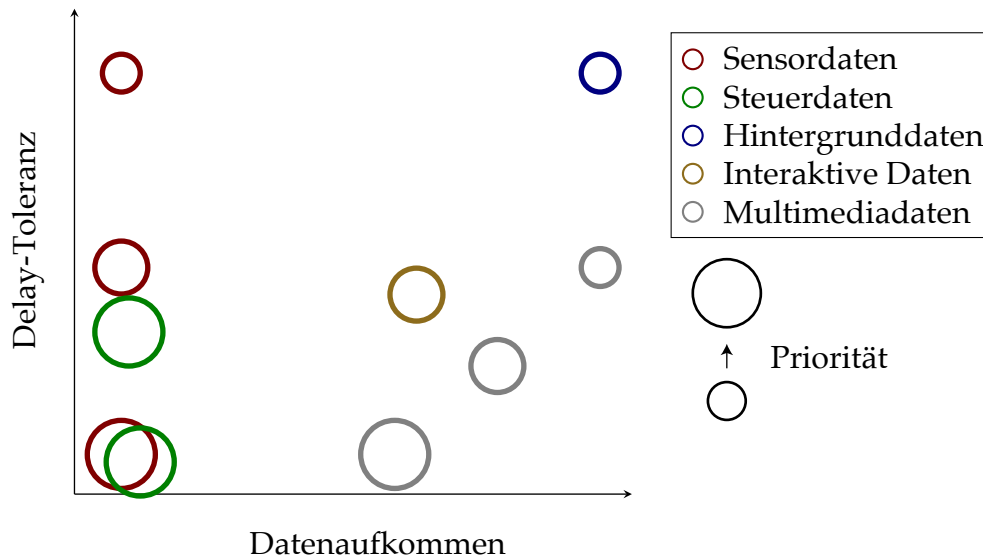


Abbildung 2.1: Einordnung verschiedener Datentypen nach Datenaufkommen, Delay-Toleranz und Priorität.

STEUERDATEN können der Befehl zum Einschalten einer Lampe oder der automatisierte Start des Bewässerungssystems sein.

HINTERGRUNDDATEN Firmwareupdates für Fahrzeuge oder der Download von Binärdateien fallen unter diese Kategorie.

INTERAKTIVE DATEN wie sie z. B. beim Interagieren mit einer Webseite anfallen.

MULTIMEDIADATEN können auf der einen Seite von "gestreamten" Videos, auf der anderen Seite von Telefonie-Anwendungen stammen.

Wie man an Abbildung 2.1 gut erkennen kann, sind die Eigenschaften der Datenübertragungen und deren Anforderungen an das zu Grunde liegende Netzwerk sehr divers. Auf der einen Seite sollen Steuerbefehle mit hoher Zuverlässigkeit und hoher Priorität möglichst schnell im Netzwerk verteilt werden. Auf der anderen Seite ist die Übertragung von großen Hintergrunddaten nicht zeitkritisch und kann, je nach Implementierung, auch mit dem Verlust von Paketen umgehen. Eine gute Diskussion zu den Problemen und Herausforderungen zur Bereitstellung verschiedener Dienstgütern insbesondere für MANet² wird in [MLGo3] durchgeführt. Wie jedoch auch Zhang et. al. in [ZhZho8] feststellen, lassen sich die traditionellen Ansätze mit abgeschlossenen Schichtenmodelle nur bedingt zur Bereitstellung solch diverser Kommunikationsansprüche verwenden.

² Mobile Ad-hoc Network (Mobiles ad-hoc Netzwerk)

Neben den verschiedenen Datentypen stellt auch die Umgebung Netzwerkprotokolle für "Ambienten Intelligenz" vor Herausforderungen. Die Menschen, die bei AmI im Vordergrund stehen sind von Natur aus mobil. Ebenso wie Endgeräte, die am Körper getragen werden oder Beförderungsmittel wie Autos oder Fahrräder. Haushaltsgeräte wie Staubsaugerroboter können selbst mobil sein oder wie z. B. Fernbedienungen bewegt werden. Diese Mobilität hat im wesentlichen zwei Auswirkungen auf die Netzwerkprotokolle zur Folge:

LIMITIERTE RESSOURCEN Werden Geräte bewegt, oder ist eine feste Stromversorgung nicht möglich, oder gewünscht, wird meistens eine Batterie oder ein Akku zur Energieversorgung genutzt. Diese besitzen jedoch eine limitierte Kapazität. Trotz unserer Bemühungen in [PrFe13] zur Reduktion des Stromverbrauchs bei Ad-hoc-Netzwerken, ist einer der maßgeblichen Energieverbraucher in Smartphones das Wlan-Kommunikationsinterface (vgl. [RTL+13]). Kommunikationsschnittstellen generell haben einen hohen Anteil am Energieverbrauch. Unter anderem steuern auch ineffiziente Routing-Protokolle ihren Teil dazu bei (vgl. [SDP+12]).

INSTABILE VERBINDUNGEN Aufgrund der Mobilität kann keine kabelgebundene Datenübertragung stattfinden. Daher wird auf drahtlose Kommunikation zurückgegriffen. Da dabei ein gemeinsames Übertragungsmedium verwendet wird, steht nicht jedem Knoten ein kollisionsfreier Kanal zur Verfügung. Daher muss hier ebenfalls auf Effizienz geachtet werden. Durch die Bewegung können sich auch die Übertragungseigenschaften jederzeit ändern. Verbindungen können sich verschlechtern oder ganz abreißen. Damit ändern sich die Topologie sowie die Kommunikationspartner und Netzwerkumgebung der Knoten.

Ein weiterer Faktor für Übertragungsprotokolle stellt die Heterogenität der Netzwerkgeräte dar. Auf der einen Seite stehen, wie zuvor beschrieben, mobile Geräte mit limitierter Energieversorgung, auf der anderen Seite befinden sich in einer intelligenten Umgebung auch stationäre Geräte mit fester, (quasi) unlimitierter Stromversorgung. Sollen Datenpakete im gesamten Netzwerk verteilt werden, sollten nach Möglichkeit diese Geräte die Daten weiterleiten. Berücksichtigt man zusätzlich die unterschiedlichen Prioritäten und Anforderungen an die Datenübertragung, lässt sich die Frage welche Knoten welche Pakete weiterleiten nicht mehr so einfach beantworten.

2.1.2 Entwurf

Zur zuverlässigen und ressourcenschonenden Kommunikation in AmI-Umgebungen wird ein Protokoll benötigt, das die in vorherigen Absätzen genannten Anforderungen berücksichtigt und gleichzeitig mit den anspruchsvollen Bedingungen mobiler Umgebungen zurecht kommt. Für Broadcast-Übertragungen, einer der anspruchsvollsten Übertragungsarten (vgl. [VSW11]), wurde in dieser Arbeit ein Protokoll entwickelt, das dazu in der Lage ist.

Bricht man die geforderten Fähigkeiten des Protokolls auf ein Minimum herunter, so soll es eine optimale Weiterleitungsentscheidung treffen. Diese Entscheidung ist dann

- zu einem bestimmten Zeitpunkt,
- für ein bestimmtes Netzwerk und
- für bestimmte Daten

optimal. Je nachdem wie der Begriff "optimal" definiert ist, wird die Weiterleitungsstrategie anders aussehen. Wenn z. B. ein möglichst geringer Energieverbrauch gemeint ist, heißt die Strategie alle Pakete zu verwerfen - mit wahrscheinlich negativen Auswirkung auf die erreichte Zustellrate. Entsprechend der Gewichtung der Ziele gibt es also mehrere optimale Strategien.

Der in dieser Arbeit verfolgte Ansatz besteht nun darin, einen Algorithmus so zu optimieren, dass er diese Strategien findet. Der Algorithmus besitzt dazu einen Parametersatz, der zum einen seine Entscheidungen bestimmt, zum anderen die Gegebenheiten der Netzwerkumgebung und des zeitlichen Verlaufs abbildet.

Die dadurch gefundenen Lösungen sind höchst netzwerkspezifisch. Werden jedoch mobile Teilnehmer in Betracht gezogen, können sich die Empfangsbedingungen sehr schnell verändern. Um solche dynamische Netzwerke zu unterstützen, kommen Klassifizierer aus dem Bereich des maschinellen Lernens zum Einsatz. Diese können mit einer Vielzahl an Entscheidungen trainiert werden. Dazu werden die Weiterleitungsentscheidungen des zuvor optimierten netzwerkspezifischen Protokolls mitgeschrieben. Zu jeder Entscheidung werden außerdem eine Vielzahl von Attributen (Stromversorgung, Anzahl Nachbarn etc.) zum Entscheidungszeitpunkt gespeichert.

Das bei der Generierung der Trainingsdaten verwendete Spektrum an Netzwerken, Datenverkehrsaufkommen und Optimierungszielen bestimmt die spätere Eignung des Protokolls. Im Rahmen dieser Arbeit

wurde eine Vielzahl an Konfigurationen abgebildet wie sie für das eingangs beschriebene AmI-Szenario denkbar sind³.

Der somit trainierte Klassifizierer stellt das Kernstück des entwickelten Protokolls, den intelligenten Paketweiterleiter, dar. Für jedes ankommende Paket kann anhand der zuvor trainierten Attribute entschieden werden ob es weitergeleitet werden soll, oder nicht.

Trotz optimaler Weiterleitungsstrategie können bei der Paketübertragung Pakete verloren gehen. Sei es durch Kollisionen beim Medienzugriff oder zu geringer Signalstärke beim Empfänger. Tritt der Verlust bei einem Knoten auf, der alleine zwei Netzwerkteile miteinander verbindet, ist das Paket auch für das entsprechende Teilnetzwerk verloren. Das Protokoll wurde daher um einen Mechanismus zur Paketwiederherstellung erweitert. Diese so genannte Vorwärtsfehlerkorrektur arbeitet ebenso wie der Weiterleitungsalgorithmus möglichst intelligent. Je nach Wichtigkeit der Daten und der erwarteten Zustellrate werden Zusatzpakete generiert, die eine Wiederherstellung verlorengegangener Pakete ermöglichen.

2.1.3 Gesamtsystem

In Abbildung 2.2 ist der Entwurf des Gesamtsystems dargestellt. Der erste Teil (in der Abbildung oben rechts, orange unterlegt) verwendet den intelligenten Paketweiterleiter, der vom unterliegenden Netzwerk empfangene Pakete analysiert. Dabei nutzt er eine Vielzahl an Datenquellen als Entscheidungsgrundlage. Aus dem lokalen Kontext können Informationen wie z. B. "verfügbare Energieversorgung" verwendet werden. Der Netzwerkkontext stellt Informationen zur Netzwerkumgebung innerhalb eines bestimmten Netzwerradius zur Verfügung. So können Daten wie "Anzahl direkter Nachbarn" oder "Entfernung zum Sender" genutzt werden. Im "Per-Paket Kontext" werden Informationen zu den einzelnen Paketen erfasst. Z. B. wie viele Duplikate eines Pakets bereits empfangen wurden.

Um dem Kompromiss zwischen Zuverlässigkeit und Ressourceneffizienz Rechnung zu tragen, wendet der Weiterleiter unterschiedliche Einstellungen in Abhängigkeit von der Wichtigkeit der Pakete an. Der Entwurf und die Realisierung dieser Komponente wird im Detail in Kapitel 3 beschrieben.

Selbst bei optimalen Weiterleitungsentscheidungen kann es zu Paketverlusten kommen. Um dennoch eine hohe Zuverlässigkeit erzielen zu können, wurde eine intelligente Vorwärts-Fehlerkorrektur implementiert. Diese ist in Abbildung 2.2 unten links lila hervorgehoben. In die-

³ Mit ein paar der Vergleichbarkeit geschuldeten Einschränkungen (vgl. Unterabschnitt 2.2.1).

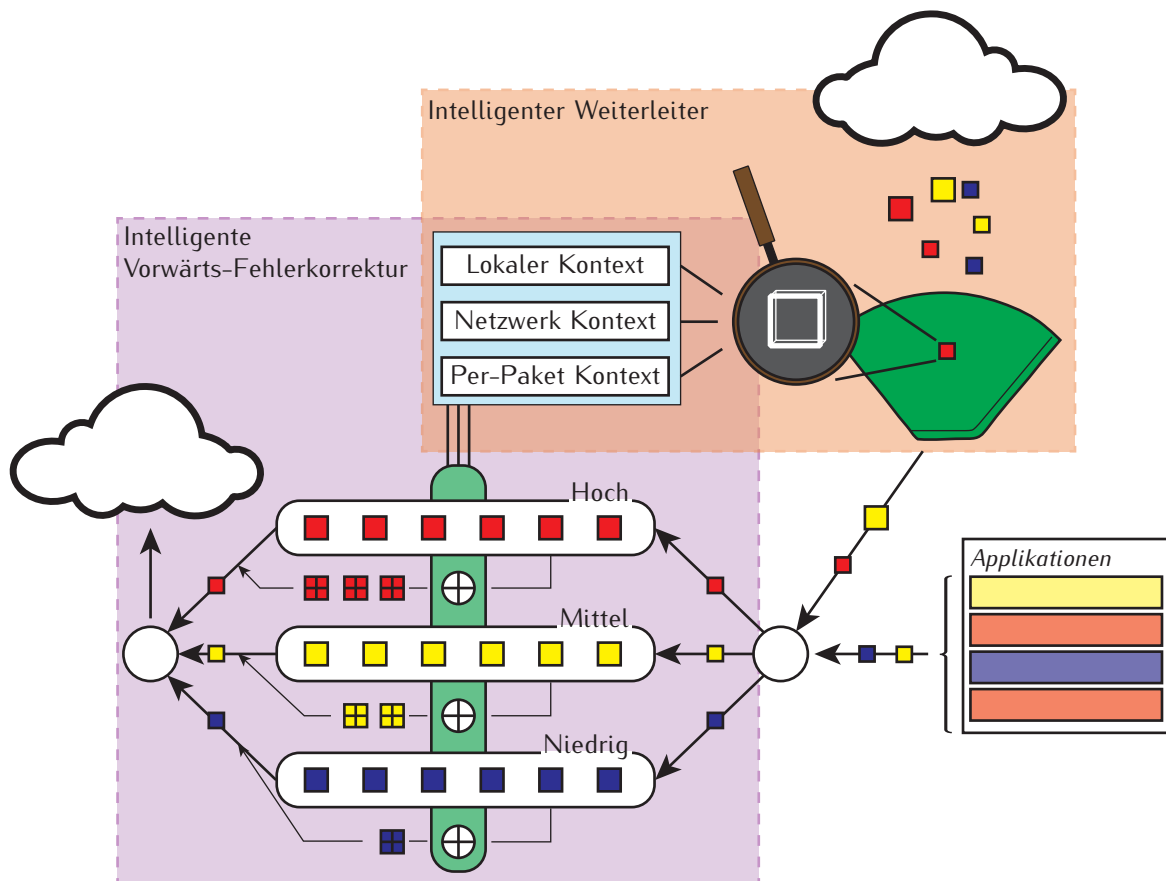


Abbildung 2.2: Entwurf eines intelligenten und zuverlässigen Broadcast Protokolls

ser Einheit werden die weiterzuleitenden Pakete zusammen mit den von lokalen Applikationen generierten Daten eingereicht und daraus, in Abhängigkeit von der Priorität, Redundanz-Daten generiert. Die zur Bestimmung der dafür notwendigen Parameter verantwortliche Einheit ist auf der Abbildung unten links in grün dargestellt. Wie bereits beim intelligenten Weiterleiter werden eine Vielzahl an Informationen verwendet, um eine möglichst optimale Entscheidung zu treffen. Der Entwurf und die Realisierung dieser Komponente wird in Kapitel 4 ausführlich beschrieben.

2.1.4 Untersuchungsgrenzen und weiterführende Literatur

Bevor im weiteren Verlauf des Kapitels die verwendete Simulationssoftware und die implementierten Erweiterungen sowie das für die realen Versuchsläufe entwickelte Smartphone-Framework vorgestellt werden, wird in diesem Unterkapitel auf verschiedene Aspekte eingegangen, die ebenfalls zu einer zuverlässigen und ressourcenschonenden Broadcast-

Übertragung beitragen. Aufgrund der Komplexität und Weitläufigkeit eines Themengebiets wie der effizienten, drahtlosen Übertragung von Daten, wird im Rahmen der vorliegenden Dissertation nur auf die zuvor genannten Teilaspekte eingegangen. Zur weitergehenden Studie des Themenkomplexes wird in den folgenden Abschnitten eine kurze Einführung und weiterführende Literatur genannt.

2.1.4.1 *Medienzugriff und Physikalische Schicht*

Eine Möglichkeit die Zuverlässigkeit und Effizienz zu verbessern besteht darin, auf den unteren Ebenen des Protokollstapels anzusetzen. Lee et al. stellen in [LSSo7] verschiedene Technologien wie Wi-Fi, Bluetooth oder ZigBee gegenüber und analysieren verschiedene Protokolleigenschaften wie z. B. die benötigte Energie um ein Byte Daten zu übertragen.

Bei der Umsetzung einer intelligenten Umgebung ist neben einem homogenen drahtlosen Netzwerk auch ein heterogenes Netzwerk bestehend aus stationären und mobilen Technologien denkbar. In [LYC+12] stellen die Autoren ein System zum kooperativen Verteilen von Daten vor, das eine Kombination aus Mobilfunk und VANET⁴ verwendet. Innerhalb dieses heterogenen Netzwerks wählt ein Algorithmus Knoten aus, die als Gateway zwischen den unterschiedlichen Technologien dienen.

Parata et al. präsentieren in [PSCo7] einen Ansatz bei dem eine traditionelle IEEE 802.11 WiFi Infrastruktur um gleichzeitige Ad-hoc-Verbindungen, auf einem anderen Kanal, erweitert wird. Diese dienen dem direkten Datenaustausch zwischen Geräten und entlasten damit das bestehende Netz. Alle Verbindungen teilen sich den gleichen Netzwerkadapter. Die Verwaltung und das Umschalten der Kanäle geschieht transparent in der Medienzugriffsschicht.

Ein weiterer Ansatzpunkt die Effizienz innerhalb von Ad-hoc-Netzwerken zu verbessern besteht in der Wahl des Medienzugriffsprotokolls. Neben Cluster-basierten Zugriffsverfahren (vgl. [Günt10]) existiert ein ganzes Forschungsfeld - die "Cognitive Networks" (Denkende Netzwerke) - das sich zum Großteil mit einem möglichst "intelligenten" Medienzugriff und Kanalnutzung befasst. Eine gute Einführung und Übersicht zu dieser Art von Netzwerken liefert [Thom07].

Bei allen Bemühungen, den CSMA⁵-basierten Medienzugriff insbesondere von VANET zu verbessern, legen neuere Untersuchungen (vgl. [Hart13]) nahe, dass dieses recht einfache Verfahren doch für die meisten Anwendungen gute Ergebnisse liefert.

⁴ Vehicular Ad-hoc Network (Fahrzeug-ad-hoc-Netzwerk)

⁵ Carrier Sense Multiple Access (Mehrfachzugriff mit Trägerprüfung)

2.1.4.2 *Zugriffskontrolle*

In [Wieg12] konnte der Autor zeigen, dass die Datenübertragungsrate einbricht wenn der Übertragungskanal überlastet wird. Dies ist selbst dann der Fall, wenn QoS-Mechanismen genutzt werden. Durch die Verwendung einer prioritätsbasierten Zugriffskontrolle konnte der Datenverkehr so gesteuert werden, dass selbst bei hohen Paketraten Pakete mit hoher Priorität zuverlässig zugestellt wurden.

Diese unter den Begriffen Zugriffskontrolle und Datenverkehrssteuerung zusammengefassten Mechanismen stellen eine weitere Möglichkeit dar, die Effizienz und Zuverlässigkeit in MANet zu erhöhen. Eine gute Übersicht über die existierenden Ansätze liefert die Zusammenfassung von Khoukhi et al. in [KBME13].

2.1.4.3 *Angriffsziele*

Ein weiterer Aspekt, der bei der Protokollentwicklung für MANet beachtet werden muss, sind Übertragungsstörungen durch Netzwerkteilnehmer, die sich nicht gemäß den Spezifikationen verhalten oder absichtlich die Kommunikation stören. Die Intentionen dahinter können unterschiedlich sein. Auf der einen Seite können Teilnehmer versuchen sich durch unkooperatives Verhalten, typischerweise auf Kosten der anderen Netzwerkknoten, einen Vorteil zu verschaffen. Z. B. wird durch höhere Übertragungsleistungen die Sendereichweite erhöht, jedoch vergrößert sich gleichzeitig auch die Kollisionsdomäne.

Auf der anderen Seite können Übertragungen absichtlich gestört werden z. B. , um unerwünschte Kommunikation zu unterbinden. Dies war 2014 der Fall, als die Behörden in Hongkong das Mobilfunknetz gezielt ausschalteten, um die Koordination der Demonstrationen für mehr Demokratie zu erschweren. Die Demonstranten wussten sich aber zu helfen und nutzen dezentrale mobile Netzwerke zur Kommunikation.

Eine gute Übersicht zu möglichen Angriffen auf MANet-Routingprotokolle ist in [TaGu13] zu finden.

2.2 ERWEITERUNG DER SIMULATIONSSOFTWARE

In diesem Abschnitt wird der Netzwerksimulator vorgestellt, der in Kapitel 3 SELEKTIVE WEITERLEITUNG und Abschnitt 4.4 Evaluation des Gesamtsystems verwendet wird. Anschließend wird auf die gewählten Simulationsparameter und die ausgewerteten Messgrößen und Metriken eingegangen. Zum Abschluss des Abschnitts wird eine Erweiterung von OMNeT++ zur schichtübergreifenden Bereitstellung von Messgrößen, die "Knowledge-Engine", vorgestellt.

OMNeT++ ist eine erweiterbare, modulare, Komponenten-basierte C++ Simulationsbibliothek, die hauptsächlich zum Bau von Netzwerksimulationen gedacht ist. Damit sind Netzwerke im weiteren Sinne wie Kabelgebundene und drahtlose Kommunikationsnetze, “on-chip” Netzwerke, usw. gemeint[Open].

Die Software ist quelloffen und kann entweder unter der GPL⁶ oder einer herstellereigenen Lizenz zur kommerziellen Nutzung verwendet werden. Die Motivation hinter der Entwicklung von OMNeT++ war es, ein mächtiges und quelloffenes Simulationswerkzeug für Forschung und Lehre sowie forschungsorientierte Unternehmen zur Verfügung zu stellen, das Computernetzwerke, verteilte oder parallele Systeme simulieren kann [Vargo1]. Seit es 1997 der Öffentlichkeit zugänglich gemacht wurde, taucht OMNeT++ mittlerweile in über 3400 Publikationen auf⁷.

Zu OMNeT++ existieren Erweiterungen für Echtzeitsimulationen, Netzwerkemulation, Datenbankintegration und verschiedenen weiteren Funktionen. Domänenspezifische Funktionen für z. B. Sensornetzwerke, Internetprotokolle, Performancemodellierung, usw. können durch Frameworks, die als unabhängige Projekte entwickelt werden, bereitgestellt werden.

Eines dieser Erweiterungen ist das INET Framework [BMS+], das Protokolle und weitere Modelle für Forscher und Studenten, die mit Kommunikationsprotokollen arbeiten, bereitstellt. Es enthält verschiedene Modelle für den Internet Stack (wie TCP, UDP, IPv4, usw.), kabelgebundenen und drahtlosen Protokollen der Sicherungsschicht (wie Ethernet, IEEE 802.11, usw.) verschiedenen Bewegungsmodellen und vielen weiteren Protokollen und Komponenten.

Einige andere Simulationsframeworks nehmen INET als Basis und erweitern es in eine bestimmte Richtung wie z. B. Fahrzeugnetze, LTE⁸ oder “peer-to-peer” Netzwerken. Die *inetmanet* Erweiterung [Ariz] ist eine Parallelentwicklung, die experimentelle Eigenschaften in Bezug auf mobile und ad-hoc Netzwerke, hinzufügt.

2.2.1 Verwendete Parameter

Netzwerksimulationen sind weit verbreitet im Design und der Evaluation von Computernetzwerken und ihren Protokollen. Es gibt viele Gründe, warum Forscher Simulationen verwenden, z. B. als Alternative oder zur Ergänzung von “Live”-Experimenten. Die Hauptgründe für die Popularität von Netzwerksimulatoren liegen in der einfachen Reproduzierbarkeit und Skalierbarkeit. Andererseits ist es wichtig, dass

⁶ GNU General Public License

⁷ Quelle: Google Scholar Ergebnisse - Stand Januar 2016

⁸ Long Term Evolution (auch 3.9G genannt)

die Simulationsszenarien die Realität möglichst akkurat widerspiegeln [HOG13].

In verschiedenen Studien [HOG13; KCC05] wurde festgestellt, dass die Ergebnisse von Veröffentlichungen im Umfeld der MANet-Protokolle oft nicht reproduzierbar sind, und damit nicht besonders glaubwürdig. Dies liegt im Wesentlichen an unzureichend dokumentierten Parametern der verwendeten Netzwerksimulatoren, aber auch an nicht ausreichend spezifizierten Programmversionen.

2.2.1.1 Programmversionen

Aus den genannten Gründen werden in diesem Abschnitt die verwendeten Programme und Bibliotheken genauer spezifiziert und auf die gewählten Parameter eingegangen. Als grundlegendes Simulationswerkzeug kam der Netzwerksimulator OMNeT++ in der Version 4.5 (Build: 140714-c6b1772) zum Einsatz. Darauf aufbauend wurde das *Inetmanet Framework 2.0* (Stand 4. Februar 2013), das auf dem *Inet Framework* in der Version 2.1.0 (Stand 31 Januar 2013) basiert, verwendet.

Als Grundlage für die weitere Protokollentwicklung wurde das im *Inetmanet Framework* enthaltene Broadcast-Protokoll “*UDPBasicFlooding*” verwendet. Das Protokoll wurde im Jahre 2000 am Institut für Telematik der Universität Karlsruhe entwickelt und implementiert ein einfaches multi-hop flooding Protokoll mit Hilfe von UDP⁹ Paketen. In den unteren Ebenen werden IP¹⁰ Pakete an die Broadcast-Adresse verschickt und mit dem 802.11 Protokoll an die anderen Knoten im Netzwerk übertragen. Die Wahl fiel auf dieses Protokoll als Basis, da es der Datenübertragung des Android-basierten Testsystems (siehe Abschnitt 2.3) entspricht. Damit sollten die Ergebnisse aus den Simulationsläufen gut auf die Messungen mit dem Testbed übertragbar sein.

2.2.1.2 Netzwerk

Um sicherzustellen, dass der Simulator korrekt arbeitet, wurden zu Beginn der Simulationen andere Protokolle implementiert und die Ergebnisse nachvollzogen. In Unterabschnitt 3.1.2 werden diese genauer vorgestellt. Als Grundlage für die Simulationsparameter wurde das Szenario aus der Veröffentlichung von M.D. Colagrosso [Colao5] gewählt. Das darin vorgestellte Protokoll verfolgt einen ähnlichen Ansatz wie das in dieser Arbeit entwickelte und beansprucht die gleichen Zielsetzungen für sich.

⁹ User Datagram Protocol

¹⁰ Internet Protocol

Tabelle 2.1: Grundlegende Parameter des Netzwerkszenarios

PARAMETER	WERT
Simulationsdauer	500 Pakete / Senderate
Simulationsgebiet	600 m x 300 m
Knotenanzahl	5,10,20,50
<i>mobile Knoten</i>	5,10,15,20,30,50
<i>stationäre Knoten</i>	0,5,10,20
Initiale Knotenverteilung	Zufällig
Mobilitätsmodell	“RandomWPMobility”
Sendereichweite	116 m
Paketlänge	128 Byte
Senderate pro Knoten	1,5,10,20,50 Pakete / s

Tabelle 2.1 zeigt eine Zusammenfassung der verwendeten Parameter. Die Notation orientiert sich dabei an den Vorschlägen aus [HOG13]. Die Werte motivieren sich aus verschiedenen Überlegungen:

SIMULATIONSDAUER Die Zeitdauer in der jeder Knoten 500 Pakete generiert hat, erwies sich in mehreren Testläufen als guter Kompromiss zwischen der benötigten Rechendauer und der Präzision der Ergebnisse. Dazu wurde die Simulation mit den identischen Parametern ausgeführt und der Startwert des Zufallsgenerators für das Netzwerk verändert (siehe auch Unterunterabschnitt 2.2.1.5).

SIMULATIONSGBIET Das Simulationsgebiet erstreckt sich über eine Fläche von 600 m x 300 m. Innerhalb dieses Bereiches werden die Netzwerkknoten initial verteilt und können sich im Laufe der Simulation darin bewegen. Betrachtet man die in Unterabschnitt 2.1.1 definierten Anforderungen, so scheint die Größe dieser Fläche nicht den meisten Szenarien für das Protokoll zu entsprechen. Dies ist der besseren Vergleichbarkeit mit den größtenteils aus dem Bereich der automobilen Netzwerken stammenden Protokollen, insbesondere [Colao5], geschuldet. Versuchsläufe mit kleineren Dimensionen und geringeren Sendereichweiten konnten zeigen, dass die Ergebnisse übertragbar sind.

KNOTENANZAHL Gemäß den Anforderungen aus Unterabschnitt 2.1.1 wird davon ausgegangen, dass es zwei verschiedene Typen von Netzwerkknoten gibt. Auf der einen Seite Geräte mit kabelgebundener Stromversorgung, die sich entsprechend nicht bewegen. Auf

der anderen Seite mobile, batteriebetriebene Geräte. Es wurden alle Kombinationen aus mobilen und stationären Knoten getestet, die in der Summe eine Zahl aus der Gesamtknotenanzahl ergaben¹¹.

MOBILITÄTSMODELL Die mobilen Knoten bewegten sich gemäß dem "Random Waypoint Mobility" (Zufällige Wegpunkt Mobilität) - Modell. Trotz dessen Unzulänglichkeiten (vgl. [YLN03]) wurde dieses Modell gewählt um eine Vergleichbarkeit mit den Protokollen des Stand der Technik herzustellen. Bei diesem Mobilitätsmodell wird zufällig für jeden Knoten ein Zielpunkt gewählt. Auf diesen bewegt sich der Netzwerkknoten im Laufe der Simulation zu. Wird der Wegpunkt erreicht, wird ein neuer bestimmt und der Algorithmus beginnt von vorne an.

PAKETLÄNGE UND SENDERATE Die Länge der Pakete bezieht sich auf die Nutzdaten, die per UDP übertragen wurden. Für die Senderate wurden alle angegebenen Raten getestet.

2.2.1.3 Drahtlosübertragung

OMNeT++ ermöglicht es für die WLAN¹² Schnittstelle und für den Übertragungskanal, eine Vielzahl an Parametern einzustellen. Die meisten Einstellungen wurden auf ihren Standardeinstellungen, bzw. den Werten des "UDPBasicFlooding" Protokolls belassen. Die wichtigsten Parameter und abweichende Werte sind in Tabelle 2.2 eingetragen.

Tabelle 2.2: Grundlegende Parameter der Drahtlosübertragung

PARAMETER		WERT
Medienzugriff und physische Schicht	IEEE 802.11g	
Broadcast-Übertragungsrate	6 Mbps	
Kanalmodell	"LogNormalShadowingModel"	
Pfadverlustexponent	3	
Sendeleistung	100 mW	
Empfangsempfindlichkeit	-82 dBm	

Die Broadcast-Übertragungsrate von 6 Mbps stellt die niedrigste Rate gemäß dem IEEE 802.11g Standard [IEEE03] dar, bei der die Daten

¹¹ mobil (m) stationär (s): m5,s0; m5,s5; m10,s0; usw.

¹² Wireless Local Area Network (drahtloses Nahbereichsnetzwerk)

OFDM¹³-codiert übertragen werden. Neben dem guten Kompromiss aus Reichweite und Datenrate ist dies auch die höchste Bitrate, die bei den Testbed-Geräten (siehe Abschnitt 2.3) erreicht werden konnte.

Das *“LogNormalShadowingModel”* [Rapp+02, Abschnitt 4.11.3] zur Simulation der Ausbreitung elektromagnetischer Wellen geht von einer Freiraumdämpfung aus, die zusätzlich um einen Pfadverlustexponenten gedämpft wird. In der Realität ist die empfangene Signalstärke in Innenräumen oder bei nicht direkter Sichtverbindung nicht nur von der Entfernung abhängig [ScPr15], sondern von zusätzlicher Dämpfung durch Gegenstände oder Interferenz durch Mehrwegeausbreitung. Diesem Aspekt trägt das verwendete Modell Rechnung, indem es eine normalverteilte Zufallsvariable einführt, die zum Pfadverlust hinzugerechnet wird. Um die Vergleichbarkeit mit anderen MANet zu wahren, wurde ein Verlustexponent von 3 gewählt. Dies entspricht in etwa der Kommunikation im 2,4 GHz Band innerhalb von Gebäuden auf dem gleichen Stockwerk [Rapp+02]. Berechnet man den Erwartungswert der Übertragungreichweite mit Hilfe der restlichen Werte, so kommt man auf die o.g. 116 m Reichweite.

2.2.1.4 Energieverbrauch

Der gesamte Energieverbrauch eines Netzknotens setzt sich aus vielen Einzelbeiträgen zusammen. Um den Verbrauch in der Simulation realitätsnah schätzen zu können, müssen neben dem Kommunikationsinterface auch weitere Komponenten wie der Prozessor oder Arbeitsspeicher berücksichtigt werden. Die Implementierung der Protokolle für potentiell unterschiedliche Plattformen spielt ebenfalls eine Rolle. Da der Energieaufwand für die Weiterverarbeitung von Netzwerkpaketen sehr viel kleiner ist als der der Übertragung an sich (vgl. [YMGo8]), wird im Simulator nur der Stromverbrauch der WLAN Schnittstelle betrachtet.

OMNeT++ bietet zu diesem Zweck eine Schnittstelle zur Batteriesimulation, in der die aktuellen Energiesparmodi eines 802.11 Gerätes eingehen. Mit jedem dieser Modi kann eine entsprechende Stromstärke, respektive Leistung bei gegebener Versorgungsspannung, hinterlegt werden. Die Werte in Tabelle 2.3 wurden dem Datenblatt des BCM 4330 ICs entnommen, der unter anderem in den Versuchsgeräten des Testbeds (vgl. Abschnitt 2.3) für die WLAN Kommunikation verantwortlich ist.

Die Werte beziehen sich dabei auf die in den vorherigen Abschnitten definierten Übertragungseinstellungen. Es wird angenommen, dass die Kabel-gebundenen Teilnehmer des Kommunikationsnetzes über ei-

¹³ Orthogonal Frequency-Division Multiplexing (Orthogonales Frequenzmultiplexverfahren)

Tabelle 2.3: Parameter des Stromverbrauchs der WLAN Schnittstelle

PARAMETER	WERT
Spannung	3,7 V
Wartend	52 mA
Empfangend	60 mA
Sendend	250 mA
Schlafend	0,18 mA

ne kontinuierliche, unerschöpfliche¹⁴ Stromversorgung verfügen. Daher wird der Stromverbrauch dieser Knoten nicht erfasst.

2.2.1.5 Wiederholungen

Der Netzwerksimulator OMNeT++ nutzt an mehreren Stellen Zufallsvariablen, um natürliches Rauschverhalten zu simulieren (z. B. bei der Signalausbreitung oder der Bewegung der mobilen Knoten), oder um zufallsbasierte Algorithmen (z. B. CSMA/CD¹⁵ oder SBA) zu realisieren. Dabei kommt in der verwendeten Version des Simulators ein Pseudozufallsgenerator (Prng¹⁶) auf Basis des "Mersenne Twister"-Algorithmus [MaNi98] zum Einsatz. Standardmäßig wird für jedes Modul innerhalb von OMNeT++ derselbe Zufallsgenerator verwendet. Es ist jedoch möglich, neue Instanzen mit anderen Initialwerten zuzuweisen oder umzuleiten.

Für die Simulationen dieser Arbeit wurden zwei Instanzen des Prng verwendet: eine für die initiale Verteilung der Knoten und deren Bewegung und eine weitere Instanz für alle anderen Aufgaben. Durch die Wahl der Startwerte für den Generator des Mobilitätsmoduls konnten so identische Netzwerke aufgebaut und diese mit unterschiedlichen Protokollen, Startwerten und Parametern getestet werden.

Dabei zeigte sich, dass die Ergebnisse mehrerer Wiederholungen bei gleichem Netzwerk und anderen Startwerten für den zweiten Prng nur in sehr geringem Maße schwankten. Wurde allerdings die Knotenanordnung geändert, kam es zu stark unterschiedlichen Ergebnissen¹⁷. Es wurde daher jede Protokoll- und Netzwerkkonfiguration mit fünf unterschiedlichen Netzwerken evaluiert.

¹⁴ Zumindest in den Maßstäben eines mobilen Ad-hocNetzwerkes

¹⁵ CSMA mit Kollisionserkennung

¹⁶ Pseudo Random Number Generator (Pseudozufallszahlen Generator)

¹⁷ Insbesondere bei Netzwerken geringer Dichte kann es vorkommen, dass kein Ende-zu-Ende Pfad zwischen zwei Knoten existiert. Dadurch ist die PDR¹⁸ immer kleiner als 100 %.

2.2.2 Knowledge Engine

Im Bereich des maschinellen Lernens geht es oft darum, ein Ergebnis (entweder quantitativ oder nach Kategorien) anhand eines Datensatzes von Merkmalen vorherzusagen. Dazu werden meistens Trainingsdaten verwendet, in denen Ergebnisse und deren dazugehörige Merkmale beobachtet wurden. Mit Hilfe dieser Daten wird ein Vorhersagemodell oder Lerner aufgebaut, mit dessen Hilfe aus neuen Messungen Ergebnisse vorhergesagt werden können[FHT09].

Diese gemessenen Merkmale müssen an verschiedenen Stellen auf unterschiedlichen Netzwerkebenen erfasst und an zentraler Stelle gespeichert werden. Dadurch können zum einen adaptive Broadcast-Protokolle auf ihre relevanten Eingabegrößen zugreifen, zum anderen kann ein maschinelles Lernverfahren seine Merkmale aus zentraler Stelle beziehen. Ebenfalls können die Daten bei Bedarf vorverarbeitet oder aggregiert werden. Zu diesem Zweck wurde OMNeT++ um ein weiteres Modul, die "Knowledge Engine" (KE¹⁹ - Wissens-Maschine), erweitert.

Insbesondere im Forschungsfeld der kognitiven Netze werden häufig Einheiten zur Erfassung von Netzwerkmetriken und der Verwaltung von Netzwerkparametern verwendet. Ähnlich wie die "Cognitive Management Engine" aus [GGK+12] und dem "Cognitive resource manager" aus [MRL+11] sammelt die KE Informationen aus dem aktuellen Gerätekontext, der Netzwerkumgebung und den unteren Schichten. Auf die Vorteile eines solchen Inter-Schichtaustauschs von Informationen für die Protokollentwicklung wird in [LSRo3] und [ZhZho8] weiter eingegangen.

Abbildung 2.3 zeigt das Design der *KnowledgeEngine*. Diese besteht im Wesentlichen aus zwei großen Einheiten: dem *KnowledgeStorage* (Wissensspeicher) und dem *KnowledgeRetriever* ("Wissensbereitsteller"). Diese beiden Einheiten werden im Folgenden genauer erläutert.

KNOWLEDGESTORAGE Dieses OMNeT++ Modul enthält drei getrennte Speicherklassen für verschiedene Informationsbereiche. Im *ContextStorage* (Kontextspeicher) können Informationen zum aktuellen Gerätekontext hinterlegt werden. Dazu gehören z.B. die aktuelle Position oder der Ladezustand der Batterie. Der *NetworkStorage* (Netzwerkspeicher) nimmt Informationen zur Netzwerkumgebung auf, wie z. B. aktuelle Nachbarn, deren Position und Paketverlustraten. Im *PerPacketStorage* (Pro-Paket-Speicher) werden zu jedem Paket Informationen gespeichert. Dazu gehören empfangene Instanzen, deren Signalstärke und Sender. Damit der Paketspeicher nicht zu viel Speicherplatz verbraucht, bereinigt er sich

19 Knowledge Engine

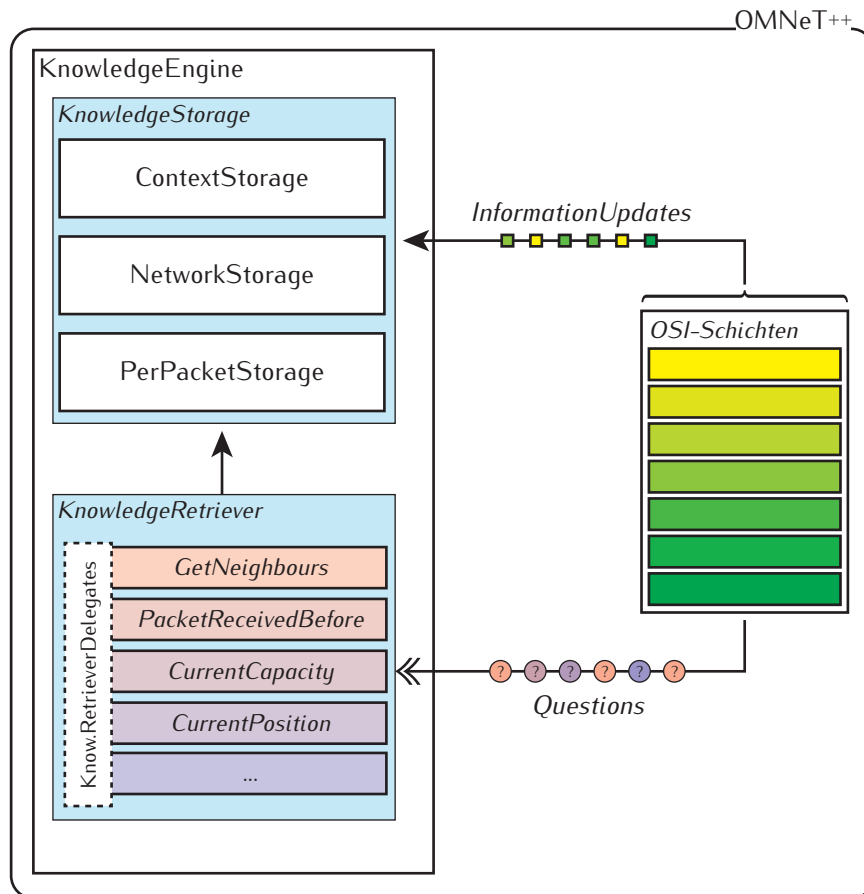


Abbildung 2.3: Design der Knowledge Engine.

in regelmäßigen Abständen selbst und behält nur die Daten eines definierbaren Intervalls.

KNOWLEDGERETRIEVER Die Klasse des "Wissensbereitstellers" enthält eine Sammlung von *KnowledgeRetrieverDelegates* (KR.-Abgeordnete), die sich unter der Angabe einer Fragen-ID ("Question") registrieren können. Ankommende Anfragen aus den verschiedenen Schichten des Netzwerkstapels mit einer bestimmten ID können so an eine "Abgeordneten"-Klasse delegiert werden. Jede dieser Klassen kann eine bestimmte Frage für einen oder mehrere Datentypen beantworten²⁰.

Um die Wissensdatenbank mit Informationen zu füllen wurde die Schnittstellendefinition des *InformationUpdate* definiert. Diese Schnittstelle kann implementiert werden, um schicht- oder applikationsspezifische Informationen an den *KnowledgeStorage* zu schicken. Wird das

²⁰ So liefert z. B. die Klasse *GetNeighbours* für die Funktion **getInt** die Anzahl an Ein-Hop-Nachbarn zurück. Für die Funktion **getSet** wird ein Set mit den Instanzen dieser Nachbarn zurückgegeben

InformationUpdate entgegengenommen, wird nacheinander die Funktion **doUpdate** mit einem der drei Speichereinheiten des *KnowledgeStorage* aufgerufen. Je nach Speichertyp werden unterschiedliche überladene Varianten aufgerufen, in denen die Informationen des Updates in die jeweilige Instanz der Speicherklasse eingetragen werden können.

2.3 FRAMEWORK ADHOC3

Die in der vorliegenden Arbeit entwickelten Protokolle sollen nicht nur in einem synthetischen Umfeld ihre Leistungsfähigkeit unter Beweis stellen, sondern auch in einem realen Testbed funktionieren. Dazu wurde ein Framework entwickelt, in dem MANet-Applikationen und -Protokolle einfach implementiert und getestet werden können. Das Framework, genannt **Adhoc3**, wurde in Kapitel 4 INTELLIGENTE VORWÄRTS FEHLERKORREKTUR sowie Kapitel 5 PROTOTYPISCHE IMPLEMENTIERUNG verwendet.

Mobiltelefone mit dem Betriebssystem Android stellen aufgrund ihrer einfachen Erweiterbarkeit in Form von eigenen Programmen (so genannten Apps) eine hervorragende Basis für das Testen von Netzanwendungen und Protokollen dar. Insbesondere für die Geräte der Google Nexus Baureihe stehen die Quellen des Betriebssystems offen zur Verfügung (vgl. AOSP²¹ ²²) und lassen sich für eigene Anforderungen ändern und übersetzen. Somit können auch Protokolle unterhalb der Applikationsebene entwickelt und evaluiert werden.

Um auch anderen Modellen eine offene Alternative zu den Betriebssystemen der Hersteller zu bieten, entwickelt die Firma und Community cyanogenmod eigene Versionen auf Basis der offenen Googlequellen ²³. Zusätzlich zu den AOSP Funktionen integriert cynogenmod weitere Funktionalität wie z.B. die Möglichkeit, Befehle als Superuser auszuführen oder die IBSS²⁴ Patches der Thinktube Gruppe ²⁵, um mit unterstützen Android Geräten ein Ad-hocNetz aufzubauen.

Die Unterstützung des Ad-hocModus lieferte schließlich auch die Idee, Android Geräte als Testplattform für MANet einzusetzen. Aus ersten Versuchen entwickelte sich schließlich die Software **Adhoc2**, die Funktionalität zur Benutzerinteraktion auf der einen und der Verwaltung von Netzwerkaufgaben auf der anderen Seite, bereitstellte. Nach mehreren Studentenprojekten und Abschlussarbeiten, war die Software

²¹ das Android Open Source Project

²² Url: <https://source.android.com/> (abgerufen am 17.5.2016)

²³ Url: <http://www.cyanogenmod.org/> (abgerufen am 17.5.2016)

²⁴ Independent Basic Service Set ("802.11 - Ad-hoc Modus")

²⁵ Url: <http://www.thinktube.com/android-tech/46-android-wifi-ibss> (abgerufen am 17.5.2016)

schließlich derart gewachsen, dass eine strukturiertere Neuentwicklung notwendig wurde.

Im Rahmen einer Abschlussarbeit wurde versucht die Anforderungen an ein Framework zu entwickeln, mit dessen Hilfe MANet-Protokolle und -Anwendungen implementiert und evaluiert werden können. Ein besonderer Fokus wurde auf die einfache Erweiterbarkeit gelegt. Die Ergebnisse wurden schließlich in [MuPr13] veröffentlicht.

Eine erste Implementierung des Frameworks, genannt **Adhoc3**, inklusive einer Beispielapplikation zur Übertragung von Echtzeit Audiodaten, wurde auf der WiVec 2013 im Rahmen einer Demonstration [PrKl13] vorgestellt. Anschließend wurde die Software in mehreren studentischen Abschlussarbeiten genutzt und verbessert. Außerdem diente sie als Experimentierplattform für weitere Veröffentlichungen [PBS14; PrSa15]. Die Funktionalität der Software und deren Aufbau, wie sie im Rahmen dieser Dissertation zum Einsatz kam, wird in den folgenden Abschnitten genauer beschrieben.

2.3.1 Funktionsbeschreibung

Adhoc3 wurde im Zusammenspiel mit Android-Geräten als Testbed für die Evaluation von MANet Protokollen und Anwendungen konzipiert. Es übernimmt die Verwaltung der WLAN Verbindung um ein Ad-hocNetzwerk aufzubauen, IP-Adressen zuzuweisen und den Anwendungen netzwerkbezogene Informationen bereitzustellen. Außerdem stellt es eine Schnittstelle zur Ansicht und Änderung persistenter Einstellungen zur Verfügung. **Adhoc3** übernimmt auch den Datenaustausch sowie die Verteilung von Datenpaketen zwischen verschiedenen Teilen der Applikation.

Das Hauptaugenmerk bei der Entwicklung lag in der einfachen Erweiterbarkeit verschiedener Aspekte. Dazu definiert **Adhoc3** verschiedene Schnittstellen, die vom Entwickler implementiert werden müssen um neue Funktionalität hinzuzufügen, bzw. um existierende Funktionalität zu erweitern. Dazu gehören:

NETZWERKAPPLIKATIONEN Innerhalb von **Adhoc3** können mehrere Applikationen umgesetzt werden, die über das unterliegende MANet kommunizieren können. Diese können über ein grafisches Benutzerinterface mit dem Benutzer interagieren²⁶. Es können auch An-

²⁶ **Adhoc3** bringt bereits einen Dienst zum Austausch von Nachbarschaftsinformationen mit. Dieser so genannte "HelloService" verfügt auch über ein grafisches Interface, in dem Informationen über die direkten Nachbarn angezeigt werden.

wendungen definiert werden, die nur im Hintergrund aktiv sind und auf bestimmte Pakete warten²⁷.

PROTOKOLLE **Adhoc3** ermöglicht eine Evaluation von Netzwerkprotokollen. Im Rahmen von Android Applikationen ist ein Zugriff auf die Netzwerkfunktionen jedoch erst auf Ebene der Transportschicht möglich. Ein Test von z.B. Medienzugriffsprotokollen ist auch aufgrund der vordefinierten Firmware der WLAN Karte nicht möglich. Um dennoch Routingprotokolle für MANet testen zu können, verwendet **Adhoc3** UDP Datagramme, die an die Broadcast-Adresse gesendet werden. Damit erreicht man ein ähnliches Verhalten wie bei einem 802.11 Interface mit CSMA/CA²⁸ Medienzugriffsverfahren oberhalb der Sicherungsschicht. Zwar wird damit eine bessere Fehlererkennung (aufgrund der Checksumme von UDP) erreicht, allerdings muss ein zusätzlicher Overhead von 28 byte pro Paket²⁹ in Kauf genommen werden.

2.3.2 Aufbau

Um die Software **Adhoc3** möglichst robust gegen Programmabstürze zu machen und eine klare Trennung zwischen der Interaktion mit dem Anwender und der eigentlichen Programmlogik zu erreichen, wurde das Programm in zwei Teile aufgeteilt. Auf der einen Seite bildet das HMI³⁰ die Schnittstelle zum Benutzer, auf der anderen Seite übernimmt ein Hintergrunddienst (genannt *Service*) die Verarbeitung und Generierung von Netzwerkpaketen.

In Abbildung 2.4 ist das Systemmodell von **Adhoc3** abgebildet. Im oberen Teil der Grafik werden die Klassen des Benutzerinterfaces dargestellt, in der Mitte die des Hintergrunddienstes und im unteren Teil schematisch das Interface zum Android Betriebssystem. Die Kommunikation zwischen *Service* und *HMI* erfolgt über die Android-eigene Interprozesskommunikation AIDL³¹. Diese ist über die so genannten *Instructions* (Anweisungen) weiter abstrahiert. Im Systemmodell sind die an dem Datenaustausch beteiligten Klassen grün gekennzeichnet. So nehmen die *Interface*-Klassen die *Instructions* entgegen und geben sie per AIDL an den jeweiligen anderen Prozess weiter. Von dort werden sie mit Hilfe der *InstructionDispatcher* (Dispatcher: Verteiler) weiterver-

²⁷ Der "ExperimenterService" ist solch ein Dienst, der zur Durchführung von Versuchsreihen auf Steuerpakete wartet, um bestimmte Protokoll- oder Geräteparameter einzustellen.

²⁸ CSMA mit Kollisionsvermeidung

²⁹ 20 byte für den IP Header und 8 byte für den UDP Header

³⁰ Human Machine Interface (Mensch-Maschinen-Schnittstelle)

³¹ Android Interface Definition Language (Android Schnittstellendefinitionssprache)

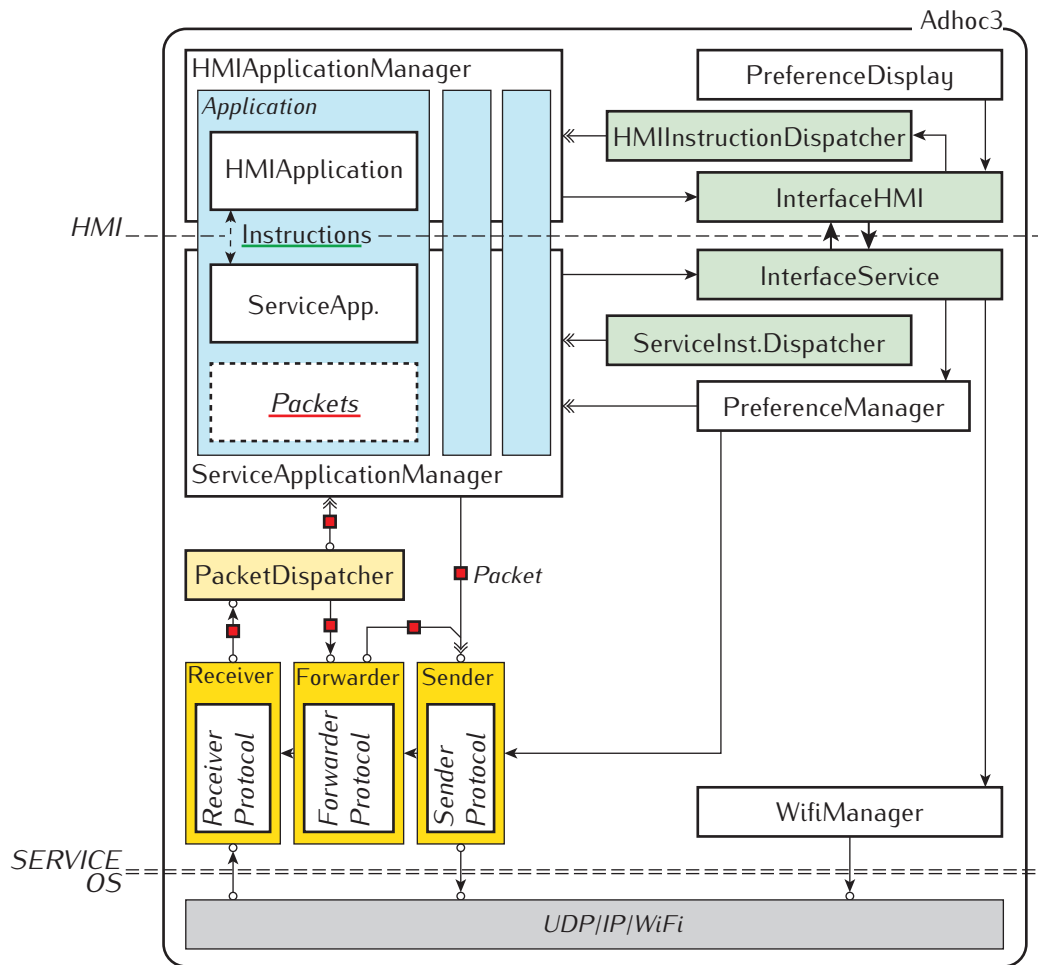


Abbildung 2.4: Systemmodell des Frameworks Adhoc3.

teilt. An diesen können sich die Applikationen für bestimmte *Instruction* Typen registrieren.

Zur Erstellung eigener Netzwerkanwendungen muss eine neue *Application* (Applikation) programmiert werden (In Abbildung 2.4 blau markiert). Diese besteht aus mehreren Teilen:

HMIAPPLICATION In diesem Teil wird das Aussehen der Anwendung sowie ihre Reaktion auf Benutzereingaben definiert.

SERVICEAPPLICATION Hier wird die Programmlogik zur Behandlung eingehender Netzwerkpakete sowie die Generierung neuer Pakete umgesetzt.

INSTRUCTIONS Wie bereits im vorherigen Abschnitt beschrieben, dienen diese Anweisungen zum Datenaustausch der beiden in getrennten Prozessen laufenden *HMI*- und *ServiceApplication*.

PACKETS Jede Applikation kann eigene Paketformate (rot gekennzeichnet) definieren, die **Adhoc3** per MANet verschicken und empfangen kann. Dazu muss ein Interface zur En- und Decodierung in ein Byteformat implementiert werden.

Um die einzelnen Applikationen zu verwalten, dienen die *ApplicationManager* Klassen. Diese übernehmen auch die Instantiierung der einzelnen *HMIApplication* und *ServiceApplication* Objekte.

Zur Evaluation von MANet Protokollen hält **Adhoc3** die in Abbildung 2.4 gelb gekennzeichneten Klassen *Receiver* (Empfänger), *Forwarder* (Weiterleiter), *Sender* und den *PacketDispatcher* (Paketverteiler) bereit. Diese Klassen verarbeiten die in den Applikationen definierten Paketformate.

Der *Receiver* empfängt UDP Pakete vom Betriebssystem mit Hilfe eines austauschbaren *ReceiverProtocol* (Empfangsprotokoll). Darin können entweder die von Android unterstützten Protokolle (z.B. UDP oder TCP) verwendet oder eigene Transportprotokolle definiert werden. Anschließend werden die empfangenen Nutzdaten zu Applikationspaketen dekodiert und an den *PacketDispatcher* weitergegeben. Dieser verteilt die Pakete an die entsprechende Applikation zur weiteren Bearbeitung. Außerdem wird das Paket an den *Forwarder* weitergegeben.

Der *Forwarder* beinhaltet ähnlich wie der *Receiver* austauschbare *ForwarderProtocol* Klassen. Darin können Multi-hop MANet Routingprotokolle implementiert werden. Diese entscheiden für jedes Paket, ob es im Netzwerk weiter verteilt werden soll. Falls die Entscheidung zur Weiterleitung getroffen wurde, wird das Paket an den *Sender* weitergegeben.

Im *Sender* laufen sowohl die generierten Pakete aus den Applikationen als auch die weiterzuleitenden Pakete aus dem *Forwarder* an. Mit Hilfe austauschbarer Transportprotokolle können diese an das Betriebssystem zum Versand weitergegeben werden.

Zur Anzeige, Änderung und Speicherung persistenter Einstellungen wird die Klasse *PreferenceDisplay* (Voreinstellungsanzeige) verwendet. Wird dort eine Einstellung geändert, wird diese per *Instruction* an den *PreferenceManager* geschickt und von dort an die Applikationen, *Receiver*, *Sender* und *Forwarder* gegeben. Der *WiFiManager* übernimmt die Verwaltung der Netzwerkverbindung. In der Variante für Ad-hocNetze wird z.B. die Schnittstelle in den Ad-hocModus versetzt und eine IP Adresse zugewiesen.

SELEKTIVE WEITERLEITUNG

Die mobile Umgebung der AmI stellt hohe Anforderungen an die grundlegenden Übertragungsprotokolle. Insbesondere bei Broadcast-Mitteilungen die alle Geräte eines Netzwerkes erreichen sollen, sind potentiell eine große Anzahl an Geräten involviert. Aufgrund der begrenzten Ressourcen muss diese Übertragung möglichst effizient erfolgen.

Im vorangegangenen Kapitel wurde bereits der Entwurf des Gesamtsystems vorgestellt, das als Übertragungsprotokoll diese Aufgabe möglichst optimal umsetzen soll. In diesem Kapitel wird der intelligente Paketweiterleiter vorgestellt, der die Verteilung der Broadcast-Pakete im Netzwerk sowohl effizient als auch zuverlässig leisten soll. Um dem Kompromiss zwischen Zuverlässigkeit und Ressourceneffizienz Rechnung zu tragen, soll der Weiterleiter unterschiedliche Einstellungen in Abhängigkeit von der Wichtigkeit der Pakete anwenden.

Um möglichst adaptiv und intelligent zu agieren, muss der Entscheidungsalgorithmus, ob ein Paket weitergeleitet werden soll oder nicht, aus den ihm zur Verfügung stehenden Informationen eine boolesche Entscheidung ableiten. Diese sollte dann im Hinblick auf den Ressourcenverbrauch des Gesamtnetzwerks und der erreichten Zuverlässigkeit optimal sein.

Damit ergeben sich drei wissenschaftliche Fragestellungen, die in diesem Kapitel nacheinander beantwortet werden:

Gibt es eine "optimale" Weiterleitungsstrategie für ein bestimmtes statisches Netzwerk und wie kann diese effizient gelernt werden?

Kann diese Strategie mit Hilfe von maschinellem Lernen nachgebildet werden?

Ist diese Nachbildung allgemeingültig und kann sie auch auf veränderliche Netzwerke angewandt werden?

Bevor auf die einzelnen Fragen eingegangen wird, werden zuerst einmal die existierenden Ansätze zur Verteilung von Broadcast-Daten in mobilen Netzwerken vorgestellt. Zur besseren Übersicht werden verschiedene Taxonomien präsentiert, anhand derer die Protokolle aus dem Stand der Technik verschiedenen Kategorien zugeordnet werden können. Anschließend werden die relevanten Ansätze genauer beschrieben.

In Abschnitt 3.2 wird mit Hilfe des Netzwerksimulators die optimale Weiterleitungsstrategie bestimmt. Anschließend wird diese mit Hilfe

von maschinellem Lernen “gelernt” und daraus ein prioritätsabhängiges Weiterleitungsprotokoll entwickelt. Das Kapitel endet mit der Vorstellung der Ergebnisse und einer Sensitivitätsanalyse.

3.1 STAND DER TECHNIK

- Einfaches Weiterleiten
- Wahrscheinlichkeitsbasiert
- Basierend auf der Fläche
- Wissen um die Nachbarschaftsbeziehungen

Abbildung 3.1: Einordnung von Broadcast-Protokollen nach [WiCa02].

- Heuristik
 - Simples Flooding
 - Wahrscheinlichkeitsbasiert
 - Flächenbasiert
- Topologie
 - Lokale Entscheidungen
 - Dominante Entscheidungen
- Energieeffizient
 - Basierend auf der Übertragungsleistung
 - Basierend auf gerichteten Antennen

Abbildung 3.2: Einordnung von Broadcast-Protokollen nach [Khel07].

Broadcast-Protokolle für mobile Ad-hocNetzwerke werden in vielen verschiedenen Bereichen verwendet. Dies können z. B. VANETs für die Fahrzeugkommunikation, drahtlose Sensornetzwerke (WSN¹) oder IoT²-Netzwerke sein. Entsprechend hoch ist die Anzahl an Vorschlägen, wie diese Art der Kommunikation bereitgestellt werden soll.

Im nächsten Abschnitt wird daher zuerst einmal versucht die relevanten existierenden Lösungen zu kategorisieren. Davon ausgehend wird eine Auswahl an Protokollen vorgestellt, die den in Unterabschnitt 2.1.1 gestellten Anforderungen zumindest in Teilen entsprechen.

3.1.1 Klassifikation von Broadcast-Protokollen

Beinahe jede Veröffentlichung zum Thema Broadcast-Protokolle für MANet versucht sich an einer Einordnung bestehender Ansätze in verschiedene Kategorien. Dies gilt insbesondere für Übersichtsstudien, die den Stand der Technik darstellen wollen. Eine der ersten größeren Zusammenstellungen ist

¹ Wireless Sensor Network

² Internet of Things (Das Internet der Dinge)

die Arbeit von Williams et al. [WiCao2] der bestehende Broadcast Ansätze in vier Kategorien unterteilt (siehe Abbildung 3.1).

Diese Einordnung wurde in [Khelo7] um weitere Eigenschaften ergänzt und verfeinert. Dabei findet eine Einteilung in drei Hauptklassen statt, die wiederum in mehrere Unterklassen aufgeteilt sind. Das Schema ist in Abbildung 3.2 dargestellt.

- Lokal: Basierend auf einer einfachen Heuristik
- k-hop: Basierend auf Nachbarschaftsinformationen
- Positionen: Basierend auf Positionen oder Distanzen

Einen anderen Ansatz wählt Al Hanbali in [AIS+08] und trifft eine Einteilung nach der Protokollstrategie (stochastisch oder deterministisch) und den verwendeten Informationen (siehe Abbildung 3.3).

Abbildung 3.3: Einordnung von Broadcast-Protokollen nach [AIS+08].

Die Fähigkeit der verschiedenen Protokolle, sich an veränderliche Bedingungen anzupassen, fasst Colagrosso in [Colao7] zu drei Klassen zusammen. Diese Kategorisierung wird in [Bakh13] neben den "statischen Protokollen", die in etwa der o. g. Einordnungen nach [Khelo7] entspricht, übernommen:

INTER-PROTOKOLL Bei dieser Art der Algorithmen können die Netzwerkteilnehmer zwischen mehreren Broadcast-Protokollen umschalten.

INTRA-PROTOKOLL Die Knoten verwenden parametrisierte Protokolle bei denen die Parameter in Abhängigkeit von der Umgebung zur Laufzeit angepasst werden.

MASCHINELLES LERNEN Bei ML³-Ansätzen lernen Knoten, sich an ihre Umgebung anzupassen und sich aus Ihren Erfahrungen laufend zu verbessern.

Wie die Autoren in [SlMa10] und [BaWe11] feststellen, fallen viele Broadcast-Protokolle nicht in eine feste Klasse, sondern kombinieren die Eigenschaften verschiedener Klassen. Es wird daher in dieser Arbeit nicht versucht eine vollkommen neue Taxonomie zu entwickeln. Vielmehr werden die oben genannten Kategorien als nicht exklusiv verstanden, sondern als Eigenschaften der Protokolle.

3.1.1.1 Wissen

Bei der Auswahl der Merkmale, die unter die Kategorie "Wissen" fallen, wurde sich an der Einteilung nach [AIS+08] orientiert. Damit ergeben sich drei Unterkategorien in die sich die Eigenschaften einteilen lassen.

- Lokales Wissen

ENERGIEVERSORGUNG In dieser Eigenschaft wird ausgewertet, ob das Protokoll die Stromversorgung der Geräte berücksichtigt.

GESCHWINDIGKEIT Wird die Geschwindigkeit des Gerätes bzw. dessen Position verwendet?

DATENVERKEHR Das Datenaufkommen wird oft als Datenrate, oder als Paketrage erfasst. In manchen Implementierungen wird die Warteschlangenlänge des drahtlosen Interface ausgewertet.

PAKETDUPLIKATE Wertet das Broadcast-Protokoll Informationen über den mehrfachen Empfang des gleichen Pakets aus?

- Direkte Nachbarn

ANZAHL Gibt die Anzahl an direkten Nachbarn des Gerätes an.

ENTFERNUNGEN Die Entfernungen können entweder aus den Positionen der Nachbarn und des Knotens berechnet oder über die Signalstärke ankommender Pakete geschätzt werden.

POSITIONEN Typischerweise wird diese Information mit Hilfe von "Hello"-Paketen übertragen. Um die Position zu erhalten muss ein Lokalisierungssystem wie z. B. GPS⁴ vorhanden sein.

- 2- oder n-hop Wissen

GRAD Der Grad eines Knotens bezeichnet dessen Anzahl an direkten Nachbarn. Um den Grad aller 1-hop Nachbarn zu erfassen ist entsprechend schon 2-hop Wissen erforderlich.

VERBINDUNGEN Mit dem Wissen welcher Netzwerkteilnehmer welche Nachbarn hat lässt sich innerhalb eines bestimmten Netzwerkradius ein Verbindungsgraph ableiten.

POSITIONEN Werden die Positionen der Knoten innerhalb eines bestimmten Radius oder des ganzen Netzwerks benötigt?

⁴ Globales Positionierungssystem

3.1.1.2 *Adaptivität*

Die Merkmale, die unter der Kategorie “Adaptivität” gruppiert sind, geben an, ob und wie sich das Protokoll an sich ändernde Bedingungen anpasst.

HEURISTIK Verwendet das Protokoll eine Form der Heuristik, z. B. in Form einfacher Regeln oder Funktionen um Protokollparameter anzupassen?

MASCHINELLES LERNEN Bei manchen Protokollen kommen Verfahren aus dem Bereich des maschinellen Lernens zum Einsatz. Dabei kann zwischen zwei Anwendungen entschieden werden:

OFFLINE Bei den offline Verfahren wird das “optimale” Verhalten eines Broadcast-Protokolls mit Hilfe von z. B. evolutionären Algorithmen aus Versuchsläufen oder Simulationen gelernt. Die Resultate können in die Heuristiken einfließen.

ONLINE Online Lernverfahren optimieren das Protokollverhalten zur Laufzeit.

3.1.1.3 *Weitere Eigenschaften*

Unter den weiteren Eigenschaften werden die Protokolleigenschaften aufgelistet, die sich nicht in die vorangegangenen Kategorien einordnen lassen.

VERZÖGERND Bei manchen Protokollen wird die Weiterleitungsentscheidung um eine gewisse Zeitdauer verzögert. Z. B. um in diesem Zeitbereich Paketduplikate zu empfangen oder um Paketkollisionen zu verringern.

PROBABILISTISCH Verwendet das Protokoll Zufallszahlen als Teil seiner Weiterleitungslogik?

DOMINANT Bei dominanten, im Gegensatz zu lokalen Entscheidungen, wird die Weiterleitungsentscheidung oder die Weiterleitungswahrscheinlichkeit vom Sender des Pakets vorgegeben.

PRIORISIEREND Berücksichtigt das Protokoll unterschiedliche Wichtigkeiten der Daten?

3.1.2 *Relevante Publikationen*

Im Rahmen der Untersuchungen zum Stand der Technik von Broadcast-Protokollen, wurde eine Vielzahl existierender Lösungen betrachtet. Diese beschränkten sich nicht nur auf den spezifischen Anwendungsfall

Tabelle 3.1: Übersicht existierender Broadcast-Protokolle

Protokoll	Wissen					Adaptiv		Probabilistisch	Verzögernd	Dominant
	Geschwindigkeit	Lokal	Paketduplikate	1-hop	2/n	Heuristik	Maschinelles Lernen			
		Datenverkehr		Anzahl	Entfernung	Grad	Verbindungen			
SBA [PeLu00]			•				•			•
aSBA [WiCa02]		•	•				•	•		•
ABS [TNCS02]			•		P					•
DBS [TNCS02]		•		•					•	
PMLB [Cola05]	•	•	•	•			•	n		
AAG [BKSW08]							•		•	
DibA [LSK09]					•			•		•
AID [BGL11]			•					•		•
ECA [RTL+13]							•		f	•
DFCN [ABF+06]				•			•	•	f	
SRL [ABVR09]		•					•	•		
SPB [BBA10]							N	•		•
EDPB [DeCh10]						•		•		•
SLAB [SiMa11]			•		•			•	f	
Obm [AHB+11]								•	f	
ABSM [RRS12]			•		P		•	•		•
[DVAv]-BHG [KoSi12]				•		•				
EMB [WSV12]								n		

der ambienten Intelligenz. Vielmehr wurden Ansätze aus dem Bereich der Fahrzeug-Fahzeugnetze, drahtlosen Sensornetzwerke und allgemeinen MANet Szenarien, wie z. B. "Suchen und Retten", unter die Lupe genommen. Ebenso wurden Multicast-Protokolle aus diesen Gebieten untersucht. Da die Ergebnisse aufgrund der unterschiedlichen Anforderungen nicht vergleichbar sind, wurden in der folgenden Vorstellung des Standes der Technik auf diese Art der Protokolle verzichtet.

Tabelle 3.1 zeigt eine Übersicht über die relevanten Publikationen. Zu jedem Protokoll wurde eine Einordnung nach der in Unterabschnitt 3.1.1 vorgestellten Klassifikation vorgenommen. Zur besseren Übersicht wurden für kein Protokoll zutreffende Eigenschaften weggelassen ⁵. Einige der vorgestellten Ansätze verwenden die Entfernung zu ihren Nachbarn als Parameter. Diese wird meistens über die Positionen der Knoten berechnet. Wie auch in den zugehörigen Publikationen festgestellt wurde, kann die Entfernung auch über die Signalstärke ankommender Pakete genähert werden. Ist jedoch explizit die Position vonnöten, wurde dies in der Tabelle mit einem "P" markiert. Um einen "netzwerkweiten" Wissensradius zu kennzeichnen wurde ein "N" verwendet. Zur Unterscheidung zwischen "online" oder "offline"-Lernverfahren wurden die Buchstaben "n" bzw. "f" gewählt.

In den nachfolgenden Absätzen werden die betrachteten Ansätze in der gleichen Reihenfolge wie in Tabelle 3.1 vorgestellt. Die Protokolle **SBA** bis **ECA** wurden als Referenz in OMNeT++ implementiert und werden daher ausführlicher beschrieben.

SBA - Skalierender Broadcast Algorithmus

Als eine der ersten Ansätze zur Reduktion redundanter Paketübertragungen bei Broadcast-Übertragungen, stellten Peng und Lu in [PeLu00] den "Scalable Broadcast Algorithm" (SBA - Skalierender Broadcast Algorithmus) vor. Bei diesem Protokoll werden die Weiterleitungsentscheidungen lokal auf jedem Netzwerkknoten getroffen. Dazu müssen bei **SBA** mit Hilfe von *Hello*-Paketen Informationen zu den Nachbarschaftsbeziehungen ausgetauscht werden. Jeder Knoten (i) hängt dazu eine Liste aller seiner Nachbarn (N_i) an solch ein Paket an.

Empfängt ein Netzwerkteilnehmer (j) nun ein Broadcast Paket (P) von einem Sender (i), verzögert er dessen Weiterleitung um eine RAD ⁶ genannte Zeitspanne. Diese wird als gleichverteilte Zufallszahl im Intervall $[0, T_{\max}]$ ermittelt. Der Parameter $T_{\max}$ kann dabei konstant sein oder wie in [PeLu00] vorgeschlagen von der Nachbaranzahl abhängen. Gleichzeitig wird von der Menge der Nachbarknoten (N_j) die bereits

⁵ Energiebewusstsein, Positionen und ob die Protokolle Prioritäten unterstützen

⁶ Random Assessment Delay (Zufällige Verzögerung zur Beurteilung)

durch den Sender erreichten (N_i) und der Sender selbst abgezogen und als Set S gespeichert:

$$S = N_j \setminus \{N_i, i\} \quad (3.1)$$

Solange RAD noch nicht abgelaufen ist, wird für jedes Duplikat von P S gemäß Gleichung 3.1 verkleinert. Sobald $S = \emptyset$ wird das Paket verworfen und nicht weitergeleitet. Ist nach Ablauf der Paketverzögerung immer noch mindestens ein Nachbar nicht erreicht worden, wird das Paket weitergeleitet.

Mit Hilfe von Simulationen im Netzwerksimulator *ns-2* konnten die Autoren zeigen, dass unter Beibehaltung der PDR deutlich weniger Pakete verschickt werden mussten.

aSBA - Adaptiver skalierender Broadcast Algorithmus

Williams und Camp testen in [WiCao2] mehrere Broadcast-Ansätze in verschiedenen Szenarien. Dabei stellten sie fest, dass das zuvor vorgestellte **SBA** zwar sehr gute Ergebnisse im Vergleich zu den anderen Ansätzen liefert, jedoch mit steigendem Paketaufkommen viele Paketduplikate weiterleitet, was zu sinkender PDR und höherer Ende-zu-Ende Verzögerung führt.

Um diesem Trend zu begegnen schlagen die Autoren eine simple adaptive Anpassung des $T_{\max}$ Parameters vor. Oberhalb von 260 Paketen wird ein Wert von 0,05 Sekunden gewählt - ansonsten 0,01 Sekunden. Mit Hilfe von Simulationen konnten sie zeigen, dass bei höherer Zustellrate weniger weiterleitende Knoten benötigt werden.

ABS - Flächenbasiertes Schema

Tseng et al. diskutieren in [TNCS02] mehrere Ansätze zur Lösung des "Broadcast Sturm" Problems in MANet. Basierend auf einer geometrischen Analyse anhand des Sende- und Empfangsradius kommen sie zu dem Ergebnis, dass die zusätzliche Fläche, die durch die Weiterleitung eines Paketes erreicht werden kann, eine gute Basis für die Weiterleitungsentscheidung darstellt.

Als eine der Möglichkeiten eines Broadcast-Protokolls nennen die Autoren daher das "Flächenbasierte Schema". Dabei wird beim Empfang eines Paketes dessen Weiterleitung verzögert. Von der durch den Senderadius definierten Fläche wird die durch die Übertragung bereits abgedeckte Fläche abgezogen. Dieser Vorgang wird für jedes weitere Paketduplikat wiederholt. Unterschreitet die verbliebene Fläche einen bestimmten Grenzwert, wird das Paket verworfen. Ansonsten wird es nach Ablauf des Verzögerungsintervalls weitergeleitet.

DBS - Distanzbasiertes Schema

Ein Nachteil der unter **ABS** genannten Methode ist die Notwendigkeit von Positionsdaten der direkten Nachbarn. Diese sind jedoch nicht immer verfügbar oder aufgrund des hohen Energieverbrauchs, z. B. von GPS Empfängern, keine Option. Die Autoren schlagen daher in [TNCS02] noch einen weiteren Ansatz vor, der auf Entfernungsinformation beruht. Diese Daten können, falls verfügbar, über die Positionen errechnet werden oder über die Signalstärke eingehender Pakete geschätzt werden.

Der Algorithmus für ein Broadcast-Protokoll läuft dabei ähnlich wie bei **ABS** ab. Statt der Fläche kommt hier ein Distanzgrenzwert zum Einsatz. Unterschreitet die Entfernung zum Sender eines Paketduplikates diesen Grenzwert wird das Paket verworfen. Es wird angenommen, dass durch eine Weiterleitung nur noch eine geringe zusätzliche Fläche abgedeckt werden kann. Wurde der Grenzwert nicht unterschritten wird das Paket weitergeleitet.

PMLB - Rein auf Maschinelles Lernen basiertes Protokoll

Einer der wenigen Ansätze, die ein "online" maschinelles Lernverfahren zur Optimierung des Broadcast-Datenverkehrs verwenden, wurde von Colagrosso in [Cola05] vorgestellt. Dabei wurde ein naiver Bayes Klassifikator verwendet, der basierend auf 6 Variablen, eine Abschätzung trifft ob das Weiterleiten "erfolgreich" ($\oplus$) oder "nicht erfolgreich" ($\ominus$) sein wird⁷. Die ausgewerteten Parameter ($d_1, \dots, d_6$) sind:

- 1-hop Nachbarn
- 2-hop Nachbarn
- Geschwindigkeit
- Anzahl an Duplikaten
- Verbindungsdauer
- Datenverkehr

Um festzustellen, ob ein Broadcast Paket weitergeleitet werden soll, werden zuerst diese Parameter bestimmt. Anschließend wird berechnet welche der beiden Fälle ($h \in \{\oplus, \ominus\}$) die höhere (argmax) a posteriori Wahrscheinlichkeit (h_{MAP}) hat:

$$h_{\text{MAP}} = \underset{h \in \{\oplus, \ominus\}}{\text{argmax}} \quad P(h | d_1, \dots, d_6) \quad (3.2)$$

⁷ $\oplus$ Nachdem ein Paket weitergeleitet wurde, wird es anschließend erneut empfangen.
 $\ominus$ Nach der Weiterleitung wird es nicht erneut empfangen.

Unter der naiven Bayes Annahme, dass alle Variablen bedingt unabhängig voneinander sind, ergibt sich damit die Näherung

$$h_{NB} = \operatorname{argmax}_{h \in \{\oplus, \ominus\}} P(h) \prod_{i=1}^6 P(d_i|h) \quad (3.3)$$

Ist die Wahrscheinlichkeit für eine “erfolgreiche” Weiterleitung höher, so wird das Paket weitergeleitet. Ansonsten wird es verworfen. Um das Online Lernverfahren zu realisieren und $P(d_i|h)$ zu berechnen, müssen nun für jede Variable zwei Statistiken laufend aktualisiert werden. Eine für $\oplus$ sowie eine weitere für $\ominus$. Außerdem eine weitere Statistik über die Wahrscheinlichkeit für den Erfolg $P(h)$.

Colagrosso identifiziert zwei Fehlerarten die sich aus der Definition des “erfolgreich” Kriteriums ergeben. Zum einen wenn Knoten ein Paket weiterleiten und anschließend eine Kopie der Nachricht von einem Sender erhalten, der das Paket auf einem anderen Weg erhalten hat. Zum anderen falls ein Nachbarknoten am Rande des Netzwerks das Paket nicht weiterleitet. In diesem Fall nimmt der Weiterleiter fälschlicherweise an, dass die Übertragung nicht erfolgreich war.

AAG - Fortgeschrittenes adaptives “Gossiping”

Bako et al. stellen in [BKSWo8] ein probabilistisches Protokoll vor, das 2-hop Wissen verwendet um die Empfangswahrscheinlichkeiten der direkten Nachbarn zu berechnen. Dazu wird zur Ankunft eines Broadcast Pakets (M) auf einem Knoten (X) von einem Sender (S) bestimmt, welche Netzwerkteilnehmer (N_S) sich im Senderradius von S befinden und damit das Paket ebenfalls empfangen haben. Anschließend wird die Menge an Nachbarn (N_X) von X bestimmt, die die Übertragung nicht empfangen haben (C_{Xnr}):

$$C_{Xnr} = N_X \setminus \{N_S, S\} \quad (3.4)$$

Für jeden dieser Knoten (c_i mit den Nachbarn N_{c_i}) werden nun die möglichen Weiterleiter (P_{c_i}) aus N_S für die Nachricht M bestimmt:

$$P_{c_i} = N_{c_i} \cap N_S \quad (3.5)$$

Basierend auf der Anzahl an Knoten in dieser Menge ($K = \#P_{c_i}$) kann nun für jeden Knoten c_i eine zugehörige Weiterleitungswahrscheinlichkeit (p_i) ermittelt werden:

$$p_i = \operatorname{prob}(K, \delta) \quad (3.6)$$

Für den einfachen Fall, dass $K = 1$ wird p_i auf 100 % festgelegt, da nur Knoten X den Nachbarn i erreichen kann. Eine Definition und Beschreibung der Funktion prob und des Parameters δ wurde in [KCGo6]

vorgestellt. Die Autoren postulieren darin, dass jede Applikation eine Empfangswahrscheinlichkeit (τ_A) spezifizieren kann. In Kombination mit der Anzahl an potentiellen Weiterleitern K und dem durchschnittlichen Radius des Broadcast Gebiets δ kann so eine minimale Weiterleitungswahrscheinlichkeit berechnet werden um τ_A einzuhalten.

Bako et al. stellten fest, dass dieser Wert für dichtere Netzwerke weiter reduziert werden kann ohne die Empfangswahrscheinlichkeit zu verschlechtern. Dieser von der Anzahl direkter Nachbarn abhängige Reduktionsfaktor (red) wurde mit Hilfe von Simulationen empirisch ermittelt.

Schlussendlich ergibt sich die finale Wahrscheinlichkeit zur Weiterleitung aus dem Maximum der Einzelwerte $p = \max(p_1, \dots, p_n)$.

DibA - Adaptiver distanzbasierter Broadcast Algorithmus

Der in [LSK09] vorgestellte Algorithmus basiert auf dem bereits vorgestellten **DBS**. Bei diesem wird ein fester Entfernungsgrenzwert verwendet unter dem Pakete nicht mehr weitergeleitet werden. Wie sich unter anderem in [WiCao2] gezeigt hat, sind solche nicht adaptiven Protokolle nicht für alle Netzwerkkonfigurationen gleichermaßen geeignet.

Die Autoren von [LSK09] schlagen daher statt einem fixen Grenzwert eine adaptive Variante vor. Dazu werden in der Wartephase des DBS Algorithmus Paketduplikate (N_p) gezählt und die minimale Entfernung ($d_{\min}$) zu einem der Sender gespeichert. Anschließend wird in Abhängigkeit von N_p der Distanzgrenzwert geändert. Die Anzahl der Paketduplikate lässt eine Abschätzung über die lokale Netzwerkdichte zu. Daher wird mit zunehmender Dichte ein höherer Grenzwert gewählt. Bei DibA geschieht dies über eine mehrfache Fallunterscheidung.

AID - Adaptiver Ansatz zur Informationsverteilung

Bakhouya et al. stellen in [BGL11] ebenfalls einen adaptiven Algorithmus vor. Dieser versucht aus dem Zeitintervall zwischen der Ankunft von Paketduplikaten auf Netzwerkdichte und Netzwerklast zu schließen. Dazu wird jede ankommende Broadcast Nachricht (M_0) um eine gleichverteilte zufällige Zeitdauer $\tau = U(0, RAD)$ verzögert. Für jedes in diesem Intervall ankommende Paketduplikat (M_i) wird die Zeitdifferenz $\Delta t_i = t_i - t_{i-1}$ zum vorhergehenden Duplikat gespeichert. Außerdem wird die Anzahl an Duplikaten (c) mitgezählt.

Sobald τ verstrichen ist, wird entschieden ob M_0 weitergeleitet werden soll. Dazu werden die Zeitintervalle Δt_i mit dem theoretischen

durchschnittlichen Ankunftsintervall $t_a = \tau/c$ verglichen (# steht hier wieder für die Anzahl):

$$s = \#(\Delta t_i > t_a) - \#(\Delta t_i \leq t_a) \quad (3.7)$$

Ist s nun kleiner als 0 wird angenommen, dass gerade sehr viele Broadcast-Pakete ankommen und das Paket wird verworfen. Andernfalls wird es weitergeleitet.

ECA - Evolutionärer Rechenansatz

Als weiterer Ansatz, die adaptive Broadcast-Übertragung zu verbessern, versuchen mehrere Veröffentlichungen die Anpassung von Protokollparametern mit Hilfe von evolutionären Algorithmen zu optimieren. Reina et al. stellen in [RTL+13] ein adaptives Protokoll zur Übertragung von Broadcast-Daten für Katastrophenschutz-Anwendungen vor. Das probabilistische Verfahren nutzt einen so genannten Ähnlichkeitskoeffizienten C_{ij} um die Weiterleitungswahrscheinlichkeit zu bestimmen.

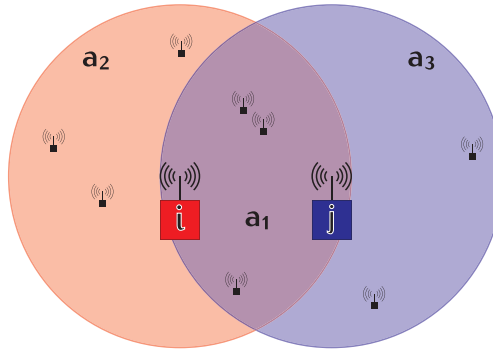


Abbildung 3.4: Schematische Darstellung der Abdeckung zweier Knoten in einem MANet.

Abbildung 3.4 zeigt eine schematische Darstellung zweier Netzwerkteilnehmer in einem MANet. Die Menge aller Knoten, die sowohl von i als j erreicht werden können wird mit a_1 bezeichnet. a_2 enthält alle Nachbarn von i , die nicht Nachbar von j sind. In a_3 sind hingegen alle Nachbarn von j enthalten, die nicht von i erreicht werden können. Mit Hilfe dieser Parameter definieren die Autoren von [RTL+13] den Ähnlichkeitskoeffizienten:

$$C_{ij} = \frac{a_1}{a_1 + \lambda(a_2 + a_3)} \quad (3.8)$$

Aus dieser Definition leiten die Autoren eine Entfernung im "Ähnlichkeitsraum" ab, die als nichtlinearer Term in die Berechnung der

Weiterleitungswahrscheinlichkeit eingeht (der letzte Summand in Gleichung 3.9):

$$p = P_{\min} + \left(1 - \frac{a_1}{a_1 + \lambda(a_2 + a_3)}\right)^\sigma \quad (3.9)$$

Ankommende Pakete werden wie in den vorangegangenen Protokollen um RAD verzögert bevor über die Weiterleitung entschieden wird. Aus diesen Überlegungen ergeben sich insgesamt vier Parameter, die mit Hilfe eines genetischen Algorithmus optimiert wurden:

λ Ähnlichkeitsfaktor

RAD Zufällige Wartezeit

$P_{\min}$ Minimale Weiterleitungswahrscheinlichkeit

σ Exponent der Wahrscheinlichkeitsfunktion

DFCN - Delayed Flooding with Cumulative Neighborhood

Ähnlich wie bei ECS verwenden Alba et al. in [ABF+06] einen genetischen Algorithmus um "offline" die besten Parameter für eine Weiterleitungsentscheidung zu treffen. Die Verwendung eines zellulären MoGa⁸ Ansatzes ermöglicht es je nach Zielsetzung verschiedene Konfigurationen auszuwählen.

Mit Hilfe eines Netzwerksimulators wurde eine Grenzwertfunktion nach den Zielen PDR, Ende-zu-Ende Verzögerung und minimaler Anzahl an Übertragungen, optimiert. Die Grenzwertfunktion betrachtet im Wesentlichen die lokale Netzwerkdichte sowie die Anzahl an Knoten, die durch das Weiterleiten eines Paketes neu erreicht werden können. Neben einer unteren und oberen Grenze für die RAD, gehen noch drei weitere Variablen in den Algorithmus ein: **minGain** beschreibt den minimalen Anteil an neu erreichbaren Knoten. **proD** ist die maximale Dichte, unterhalb derer bei neuen Nachbarn sofort alle zur Zeit auf Weiterleitung wartende Pakete versendet werden. Zuletzt definiert **safeDensity** eine Netzwerkdichte bis zu der alle Pakete weitergeleitet werden.

Das Verfahren wurde mit Hilfe von drei verschiedenen MANet Szenarien evaluiert: einem Einkaufszentrum, einem stadtweiten Netzwerk und einem automobilen Szenario. In der Auswertung werden die Ergebnisse für die unterschiedlichen Ziele präsentiert, jedoch nicht die zugehörigen Parameter, die zu den Ergebnissen geführt haben. Damit können die Resultate nicht für andere Netzwerke getestet werden.

8 Multi-Objective Genetic Algorithm (Multi-objektiver genetischer Algorithmus)

SRL - Netzwerklastbewusste Broadcast-heuristik für "Dominant Pruning"

Das in [LiKio0] vorgestellte "Dominant Pruning" (Dominantes Stutzen) beschreibt einen Algorithmus, bei dem ein Broadcast-Sender für jedes Paket mitteilt, welche seiner Nachbarn das Paket weiterleiten sollen um alle 2-hop Nachbarn zu erreichen. Dieser Algorithmus wurde in der Veröffentlichung von Al Amin et al. [ABVR09] dahingehend abgewandelt, dass eine gleichmäßige Lastverteilung auf allen Knoten des Netzwerks erreicht wird. Damit sollte laut den Autoren die Anzahl an Kollisionen und die Paketlaufzeit verringert und der Energiebedarf gleichmäßiger auf alle Knoten verteilt werden.

Die Auswahl der Weiterleiter erfolgt bei **SRL** mittels einer Heuristik, die nach der 2-hop Nachbar Erreichbarkeit und der aktuellen Last auf dem Weiterleiter gewichten kann. Dazu hängt jeder Knoten an die Broadcast- und "Hello"-Pakete seine aktuelle Netzwerklast an. In der Auswertung der Simulationsläufe wurden Latenzzeiten und die Verteilung der Paketweiterleitungen ausgewertet. Eine Betrachtung der PDR fand nicht statt.

SPB - Smartes probabilistisches Broadcasten

Yassein et al. beschreiben in [BBA10] mit **SPB** einen alternativen Algorithmus für das Broadcasten in **AODV** verbundenen Netzen. Dazu wird auf jedem Knoten die durchschnittliche Anzahl von Nachbarn im Netzwerk berechnet. Mit diesem Durchschnittswert werden Teilmengen des Netzwerks berechnet und ebenfalls wieder die durchschnittliche Anzahl an Nachbarn berechnet.

Um die Weiterleitungswahrscheinlichkeit zu bestimmen vergleichen die einzelnen Knoten ihre Zahl der 1-hop Nachbarn mit den Durchschnittswerten. Über eine Fallunterscheidung wird zwischen vier verschiedenen Wahrscheinlichkeiten ausgewählt. Diese für die Simulation genutzten Parameter werden in der Veröffentlichung nicht genannt. Ebenso scheint ein netzwerkweites Wissen über die Anzahl an Nachbarn, zumindest in größeren Netzen, nicht realistisch.

EDPB - Effizientes und dynamisches probabilistisches Broadcasten

Dembla und Chaba stellen [DeCh10] einen weiteren probabilistischen Ansatz vor, der die Anzahl direkter Nachbarn (n) mit dem Durchschnittswert der Nachbarn ($\bar{n}$) vergleicht. Ist $n > \bar{n}$, wird eine hohe lokale Dichte angenommen und die Weiterleitungswahrscheinlichkeit entsprechend niedrig gewählt (p_1). Ansonsten leiten die Knoten mit einer höheren

Wahrscheinlichkeit (p_2) weiter. Die Autoren geben keine Werte für diese beiden Parameter an.

SLAB - Statistisches standortunterstütztes Broadcasten

Ähnlich wie im bereits vorgestellten **DBS** Algorithmus, versucht das von Slavik und Mahgoub in [SlMa11] vorgestellte Protokoll eine Weiterleitungsentscheidung anhand eines bereits abgedeckten Empfangsbereiches zu treffen. Dazu wird die Weiterleitung einer Broadcast-Übertragung (P) um $t = T_{\max}(1 - d/r)$ verzögert. Wobei d die Distanz zum Sender und r den erwarteten Übertragungsradius bezeichnet.

Nach Ablauf dieses Zeitintervalls wird der Abstand (M) zum geometrischen Mittelpunkt $(\bar{x}, \bar{y})$ aller Sender von Duplikaten von P berechnet und auf den Übertragungsradius normiert:

$$M = \frac{1}{r} \sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} \quad (3.10)$$

Ist M nun größer als ein bestimmter Grenzwert M_c wird das Paket weitergeleitet. Mit Hilfe von Simulationen konnten die Autoren zeigen, dass die Erreichbarkeit in hohem Maße von M_c abhängt. Wird der Wert zu niedrig gewählt steigt zwar die Zuverlässigkeit, allerdings auf Kosten der Effizienz. Ein zu hoher Wert führt zu einer geringen PDR.

Zur adaptiven Anpassung von M_c wurde eine Grenzwertfunktion auf Basis eines MLP⁹ definiert. Dabei wurden zwei versteckte Ebenen und drei Eingangsvariablen gewählt:

- N Die Anzahl der direkten Nachbarn.
- Q Die so genannte Quadrat Statistik erfasst wie regelmäßig Punkte in einer Ebene angeordnet sind.
- κ Der Rice-Dämpfungsfaktor, der den Anteil der direkten Sichtverbindung bei einer Mehrwege-Signalausbreitung quantifiziert.

Mit Hilfe einer Fitnessfunktion, einem genetischen Algorithmus und vielen Simulationsläufen wurde die Grenzwertfunktion optimiert. Die Fitnessfunktion gewichtete dabei hauptsächlich nach der Erreichbarkeit sowie als untergeordnetem Parameter nach der "Übertragungsdichte" (siehe [SlMa11]). Als Ergebnis der Optimierungsläufe werden die Werte für das MLP angegeben, jedoch findet sich die Zuordnung der Eingangsvariablen nicht in der Veröffentlichung. Da das Protokoll für den Einsatz in Fahrzeug-Fahrzeug-Netzen gedacht ist, wurde es auf das 5,9 GHz Band und Freiluftszenarien optimiert.

⁹ Multi Layer Perceptron (Mehrschicht Perzeptron)

Obm - Optimierte Broadcast-Methoden mit Hilfe evolutionärer Algorithmen

In der Veröffentlichung von Abdou et al. [AHB+11] werden verschiedene evolutionäre Algorithmen untersucht, um die Verteilung von Broadcast-Daten in einem MANet zu optimieren. Dazu wählten die Autoren einen probabilistischen Ansatz bei dem, zur Steigerung der Zuverlässigkeit, ein Paket mehrfach versendet werden kann. Damit ergeben sich insgesamt vier Parameter, die das Protokollverhalten definieren: Die Weiterleitungswahrscheinlichkeit, die Anzahl an Wiederholungen, die Verzögerung zwischen den Wiederholungen und schließlich mit der TTL¹⁰ die maximale zulässige Anzahl an Weiterleitungen.

Insgesamt werden in den verschiedenen Szenarien vier Metriken mit Hilfe von MoGa minimiert:

- Anzahl der Kollisionen
- “Propagation Time” Zeit, nach der alle Knoten eines bestimmten Gebiets die Nachricht erhalten haben
- Anzahl der wiederholten Übertragungen
- Anteil “Erfolgreicher” (keine Paketverluste) Simulationsläufe (zu maximieren)

Mit dem gewählten Ansatz ist nur möglich für eine gegebene Netzwerkkonfiguration eine optimale Konfiguration zu finden. Jedoch wird im Ausblick der Veröffentlichung vorgeschlagen die lokale Dichte heranzuziehen um die Kommunikationsstrategie zu wechseln.

EMB - Effiziente multihop Broadcast-Kommunikation mit verteilter Protokoll Evolution

Wiandt et al. stellen in [WSV12] ein System vor, bei dem nicht nur Broadcast-Pakete übertragen werden, sondern gleich das ganze Protokoll. Dazu entwickelten die Autoren eine genetische Programmiersprache zur Protokollbeschreibung. Mit Hilfe dieser Sprache kann eine Entscheidungslogik aufgebaut werden, die ein Broadcast-Protokoll implementiert. Diese Logik wird periodisch an benachbarte Knoten weitergegeben, wo mit Hilfe der Metriken “Anzahl Duplikate” und “Nützliche Pakete” dessen Fitness evaluiert wird.

Um das Verhalten des Protokolls iterativ zu verbessern konkurrieren verschiedene Implementierungen um die Weiterverbreitung. Dabei kommt auch ein genetischer Algorithmus zum Einsatz, der mit den

¹⁰ Time to Live (Gültigkeitsdauer)

üblichen Operatoren “Mutation” und “Rekombination” neue Logiken erzeugt. Mit Hilfe von Simulationen konnte gezeigt werden, dass im Laufe der Zeit dieser “online” Ansatz immer bessere Ergebnisse lieferte.

[DVA_v]BHG - Distanz-, Valenz- und Durchschnittsbasiertes “Handschlags-Gossiping”

In [KoSi12] werden gleich drei Ansätze für adaptive multi-hop Broadcast-Protokolle vorgestellt, die auf dem gleichen Prinzip beruhen. Dazu verwendeten Kokuti und Simon ein Signalisierungsverfahren, das dem RTS/CTS¹¹ Verfahren von CSMA/CA ähnelt. Der Sender signalisiert seine Sendebereitschaft und wartet auf die Empfangsbereitschaft seiner Nachbarn. Anhand der Entfernung zu den Nachbarknoten und deren Grad werden für die Nachbarn Weiterleitungswahrscheinlichkeiten berechnet und dem Datenpaket vorangestellt.

Die drei vorgestellten Protokolle unterscheiden sich nur in der Art wie diese Wahrscheinlichkeit berechnet wird. **DBHG** berücksichtigt dazu nur die Distanz zu den Nachbarn, **VBHG** zusätzlich den Grad der Nachbarknoten und **AvBHG** noch den lokalen durchschnittlichen Grad der Knoten.

In der abschließenden Evaluation findet keine Betrachtung des benötigten Overheads statt. Dieser dürfte nicht unerheblich sein, da neben den Signalisierungspaketen auch die einzelnen Pakete vergrößert werden.

ABSM - Bestätigungsbasiertes Broadcast-Protokoll für statische bis hochmobile Netze

Ein weiteres adaptives Protokoll für die Verteilung von Broadcast-Daten in Fahrzeugnetzen stellten Ros et al. in [RRS12] vor. Der gewählte Ansatz ähnelt dem bereits vorgestellten **SBA**. Im Gegensatz zu den festen Werten für die Obergrenze des Wartezeitintervalls (RAD) wurde bei ABSM ein graphentheoretischer Ansatz gewählt. Dazu bestimmen die Knoten, ob sie Teil eines MCDS¹² Rückgrats sind. Falls sie das nicht sind, wird ein hoher Wert für RAD verwendet, andernfalls ein niedriger.

Zur höheren Zuverlässigkeit verwendet der Algorithmus **ACK** Signalisierungen, die die erfolgreiche Zustellung eines Paketes quittieren. Diese ergänzen auch den **SBA** Ansatz, bei dem Nachbarn, die das Paket bereits empfangen haben aus einer Liste gestrichen werden. Liegt nach

¹¹ Ready to Send - Clear to send (Sendebereitschaft - Sendefreigabe)

¹² Minimum Connected Dominating Set (Kleinste Menge verbindender Knoten)

einem bestimmten Zeitintervall keine Bestätigung jedes direkten Nachbarn vor, wird die Zustellung erneut gestartet.

Die Autoren argumentieren, dass der Overhead durch die **ACK** Pakete nicht ins Gewicht fällt, da sie an periodische "Hello"-Pakete angehängt werden. Eine quantitative Auswertung dazu wurde nicht durchgeführt.

3.1.3 Zusammenfassung

Betrachtet man die in den voranstehenden Abschnitten vorgestellten Lösungen und vergleicht diese mit den in Unterabschnitt 2.1.1 definierten Anforderungen, so gibt es keinen Ansatz, der alle erfüllen kann. Insbesondere berücksichtigen die vorgestellten Protokolle weder den Energieverbrauch, noch können den Broadcast-Daten Prioritäten zugewiesen werden.

Nun ist es nicht so, dass der Energiebedarf in den Protokolluntersuchungen keine Rolle spielt. In vielen Veröffentlichungen wird argumentiert, dass eine generelle Reduktion des Paketaufkommens - insbesondere redundanter Übertragungen - zu einem geringeren Ressourcenverbrauch führe. Dabei wird jedoch nicht berücksichtigt, dass der Energieverbrauch in heterogenen Netzwerken unterschiedliche Auswirkungen auf die einzelnen Knoten haben kann. So ist für Netzwerkgeräte, die ihren Strom aus einem Stromnetz beziehen, der Energiebedarf ein geringerer Faktor als für mobile, batteriebetriebene Geräte.

Ansätze, die die Stromversorgung oder Kosten für die Energie berücksichtigen, stammen häufig aus dem Bereich der WSN. Jedoch unterscheiden sich die Kommunikationsmuster bei diesen Netzen stark von Broadcast-Übertragungen in MANet. Es wird bei den Sensornetzen angenommen, dass alle Teilnehmer periodisch Daten generieren und diese an ein zentrales Gateway oder Server übermittelt werden müssen¹³. In [WNE02; JSAC01; ZKLY10] werden Broadcast-Protokolle präsentiert, die anhand eines "Broadcast Baumes" Daten innerhalb eines MANet verteilen. Bei der Konstruktion dieses Baumes werden auch Energiedaten, z. B. der aktuelle Akkufüllstand, verwendet. Zur Erstellung des Baumes auf einem Knoten wird jedoch netzwerkweites Wissen benötigt. Damit werden diese Lösungen für größere Netze nicht gut skalieren. Ebenfalls sind netzwerkweite Informationen in mobilen Netzen schnell veraltet und erfordern ein zusätzliches Datenaufkommen, um die Informationen aktuell zu halten.

Zur Priorisierung der Daten wird meist auf die höheren Ebenen, bzw. auf zusätzliche Lösungen wie Zugriffs- oder Datenverkehrssteuerung (vgl. z. B. [Wieg12]) verwiesen. Auch wenn die vorgestellten Lösungen

¹³ Man spricht in diesem Kontext von "Sources" und "Sink".

Priorisierung nicht explizit erwähnen, so lässt sich doch bei manchen Protokollen eine Wichtigkeit auf einen der Protokollparameter abbilden. So erlaubt z. B. das **AAG** Protokoll die Definition einer gewünschten PDR.

Die meisten Protokolle verwenden einen oder wenige lokale Parameter als Entscheidungsgrundlage. Darauf basierend werden die individuellen Entscheidungen in Form von Heuristiken oder Schwellenwerten getroffen. Auf der anderen Seite stehen Protokolle, die die Struktur des Netzwerkes als einen Graph betrachten und versuchen mit graphentheoretischen Ansätzen, wie z.B. dem MCDS das Entscheidungsproblem zu lösen.

Die Ansätze aus dem Stand der Technik sind somit nicht vollständig für den Einsatz in AmI-Umgebungen geeignet, da sie entweder wichtige Eigenschaften nicht beachten oder viel Potential durch die Beschränkung auf wenige lokale Parameter verschenken. Durch das in den folgenden Absätzen entwickelte Protokoll soll ein optimal auf die Gegebenheiten von AmI zugeschnittenes Broadcast-Protokoll zur Verfügung stehen, das seine Entscheidungen auf eine Vielzahl von Parametern stützt.

3.2 EINE OPTIMALE WEITERLEITUNGSSTRATEGIE

Trotz neuerer Ansätze, die sich adaptiv an die Netzwerkeigenschaften anpassen oder per ML versuchen eine Weiterleitungsstrategie zu entwickeln, gibt es kein Protokoll, das für den Einsatz in AmI-Szenarien optimal geeignet wäre. Insbesondere die Optimierung auf Energieverbrauch und die Beachtung unterschiedlicher Wichtigkeiten ist nicht gegeben.

In diesem Abschnitt wird der Fragestellung nachgegangen, welche Weiterleitungsstrategie für ein gegebenes statisches Netzwerk optimal ist. Kritisch ist hierbei der Begriff "optimal" zu sehen. Hinsichtlich des Energieverbrauches ist die beste Strategie einfach nicht weiterzuleiten - entsprechend niedrig wird allerdings die Paketzustellrate sein.

Die Datenübertragung in einem Drahtlosnetzwerk hängt an vielen Stellen vom Zufall ab. So sinkt z. B. die Wahrscheinlichkeit, dass ein Paket erfolgreich empfangen werden kann mit der Entfernung. Ebenfalls verwendet z. B. CSMA/CA Zufallswerte für den Kanalzugriff. Bei komplexeren Netzwerken kommen dazu noch Probleme wie z. B. "Hidden Station"¹⁴. Eine analytische Vorhersage, wie sich ein reales Netzwerk verhalten wird, ist daher schwer.

Zur Evaluation der verschiedenen Parametersätze kommen daher Simulationen mit dem bereits vorgestellten Simulator OMNeT++ zum Ein-

¹⁴ Dieses Phänomen tritt auf, wenn zwei Stationen sich in der Reichweite eines Empfängers befinden, sich gegenseitig aber nicht detektieren können.

satz. Dabei können zur Laufzeit eine Vielzahl von Parametern erfasst und im Anschluss ausgewertet werden.

3.2.1 Evolutionäre Optimierung

Die wahrscheinlich einfachste Form eine Weiterleitungsstrategie zu entwickeln besteht darin, für jeden Teilnehmer eines Netzwerkes festzulegen, ob Pakete weitergeleitet werden sollen oder nicht¹⁵. Diese binäre Entscheidung (e) lässt sich zu Beginn einer Simulation für jeden Netzknoten ($i \in \{\text{Netzwerk mit } n \text{ Knoten}\}$) einstellen. Im Anschluss wird die Simulation gestartet und über eine bestimmte Zeitdauer die vorgegebene Konfiguration ($K = (e_1, e_2, \dots, e_n)$) evaluiert.

Am Ende eines Simulationslauf werden verschiedene Metriken erfasst um die "Fitness" oder Leistungsfähigkeit der Weiterleitungsstrategie zu bestimmen.

PDR Die PDR gibt das Verhältnis zwischen empfangenen und gesendeten Paketen an. Dazu wird die Anzahl der empfangenen Pakete im Messzeitraum ($\Delta r = r_{\text{Ende}} - r_{\text{Start}}$) und die Anzahl der gesendeten Pakete ($\Delta s = s_{\text{Ende}} - s_{\text{Start}}$) für jeden Knoten erfasst. Damit ergibt sich die PDR zu:

$$\text{PDR} = \frac{\sum_i^n \Delta r_i}{(n-1) \sum_i^n \Delta s_i} \quad (3.11)$$

Zu beachten ist, dass von einem Knoten versandte Pakete auf diesem nicht zu den Empfangenen gezählt werden.

LEISTUNG Der durchschnittliche Leistungsbedarf ($\bar{P}$) wird ermittelt indem die Restkapazität (c) der Batterie für jeden Knoten abgefragt wird. Über die Versuchsdauer (t) ergibt sich die durchschnittliche Leistung pro Knoten zu:

$$\bar{P}_i = \frac{\sum_i^n \Delta c_i}{n} \quad (3.12)$$

ENDE ZU ENDE VERZÖGERUNG Das so genannte "Delay" (d) wird für jedes empfangene Paket (p_r) als Differenz zwischen Empfangszeitpunkt (t_r) und Zeitpunkt der Paketerstellung (t_c) gespeichert. Die durchschnittliche Verzögerung ergibt sich damit zu:

$$\bar{d} = \frac{\sum_{p_r} (t_r - t_c)}{\Delta r} \quad (3.13)$$

¹⁵ Ein Paket meint in diesem Zusammenhang immer ein einzigartiges Paket. Duplikate desselben Pakets werden verworfen.

Möchte man nun die optimale Konfiguration z. B. bezüglich der PDR ermitteln, so kann man alle Kombinationen von K simulieren, um die maximale Zustellrate zu bestimmen. Für ein statisches Netzwerk mit 20 Knoten gibt es 2^{20} verschiedene Möglichkeiten für K . Bei einer Paketrage von 20 Paketen/s benötigt ein normaler Büro PC¹⁶ in der Größenordnung von 5 s um K für 5 Simulationssekunden auszuwerten (entspricht 2000 generierten Paketen). Um alle Kombinationen zu testen, müsste der Computer folglich ca. 60 Tage rechnen¹⁷.

Beachtet man, dass im o. g. Beispiel nur ein Parameter mit zwei Einstellmöglichkeiten verwendet wurde, so wird schnell deutlich, dass diese Art der Optimierung für komplexere Entscheidungsalgorithmen nicht mehr praktikabel ist. Die Mathematik hält zur Lösung solcher nichtlinearer Optimierungsprobleme die Klasse der evolutionären Algorithmen bereit. Einen guten Überblick über die verschiedenen Ansätze dazu liefert [TLZ15]. Viele davon wurden bereits im Umfeld der Netzwerkkommunikation eingesetzt (siehe u. a. [RMR07] und [Förso7]).

Im Rahmen dieser Dissertation wurden verschiedene Ansätze implementiert und getestet. Unter anderem kam "Simulierte Abkühlung", der "Bergsteigeralgorithmus" und genetische Algorithmen mit verschiedenen Selektions- und Rekombinationsverfahren zum Einsatz. Dabei wurde eine Fitnessfunktion verwendet, die die drei o. g. Metriken gewichtet und zu einem Wert zusammenfasst.

Die Verwendung von Zufallswerten im Kanalmodell, dem Kanalzugriff, etc. führt zu einem verhältnismäßig hohen Rauschen der Metriken. Zur Ermittlung der Werte wurde daher eine längere Versuchsdauer bzw. die Bildung von Mittelwerten mehrerer Durchläufe notwendig. Bei allen im vorherigen Ansatz genannten Ansätzen brauchte es dadurch sehr lange bis der Algorithmus zu einem Optimum konvergierte. Insbesondere bei den später vorgestellten höherdimensionalen Weiterleitungsstrategien wurde teilweise kein Optimum gefunden. In Anbetracht der zusätzlichen Kombinationsmöglichkeiten durch die Wahl der Gewichte der Fitnessfunktion wurden diese Ansätze verworfen.

3.2.2 Mehrzieloptimierung

Eine Möglichkeit zur Optimierung nichtlinearer Systeme hinsichtlich mehrerer Ziele bietet die Klasse der "Multi Objective Genetic Algorithms" (MoGa Mehrziel genetische Algorithmen). Ein weit verbreiteter

¹⁶ Intel i5-3470 CPU mit 32 GB Arbeitsspeicher

¹⁷ Da die Versuchsläufe unabhängig voneinander sind, lassen sie sich gut parallelisieren. Entsprechend sinkt die Dauer linear mit der Anzahl der verwendeten Recheneinheiten: Für den genannten Prozessor mit 4 Kernen auf 15 Tage

Algorithmus, der für die Optimierung der Weiterleitungsstrategie sehr gute Ergebnisse liefert, ist der in [DPAMo2] vorgestellte **NSGA-II**.

NSGA-II wurde so in OMNeT++ implementiert, dass das ganze Optimierungsverfahren in einer fortlaufenden Simulation geschehen konnte. Dazu wird zuerst eine initiale Population erstellt. Jedes Genom daraus wird anschließend für eine bestimmte Dauer simuliert und die Resultate für jedes Individuum gespeichert. Nach dem Durchlauf einer Population werden die Ergebnisse gemäß dem "nicht dominaten Sortieren" und der "Crowding" (Engstand)-Distanz von **NSGA-II** sortiert. Aus den besten Ergebnissen wird anschließend eine Elternpopulation gebildet, die durch Rekombination wieder eine neue Population erzeugt. Als Rekombinationsoperator kam dabei "Simulated Binary Crossover" (Simulierte binäre Rekombination) nach [DeAg94] zum Einsatz.

Abweichend von der originalen Implementierung von **NSGA-II** wird für die Lösungen auf der Pareto Front keine gleichmäßige Verteilung angestrebt. Vielmehr sollen "gute" Ergebnisse (gemäß einer Fitnessfunktion) zu einem größeren Anteil in die nachfolgenden Generationen eingehen. Dazu wird die Auswahl zweier Elternteile in der Rekombinationsphase mit Hilfe des Rouletterad Algorithmus nach [LiLi12] getroffen. Die zugrunde liegende Fitnessfunktion (siehe Gleichung 3.15) gewichtet die PDR zu 85 %, den Energieverbrauch mit 10 % und die Ende-zu-Ende Verzögerung mit 5 %. Eine stabile und schnelle Implementierung der Operatoren von **NSGA-II** wurde in jMetal (siehe [DuNe11]) umgesetzt. Daher wurde der "Crowding"-Operator aus diesem Framework übernommen und für C++ angepasst.

Die Näherung an eine optimale Weiterleitungsstrategie erfolgte mit Hilfe von drei verschiedenen Typen von Chromosomen, die in den folgenden Absätzen beschrieben werden.

3.2.2.1 Cluster basiertes Chromosom

Bereits in [DaBh97] verwendeten die Autoren ein graphentheoretisches Verfahren, das MCDS, um ein Kommunikationsrückgrat zur Datenverteilung in einem Ad-hoc-Netzwerk zu berechnen. Damit sollte es auch möglich sein Broadcast-Daten entlang dieser Knoten weiterzuleiten um alle Netzwerkteilnehmer bei minimalen Energieverbrauch zu erreichen.

Ähnlich dieser Theorie, bei der die Weiterleitungsstrategie eines Knotens daraus besteht Pakete entweder weiterzuleiten oder zu verwerfen, wurde das "Cluster Chromosom" entworfen. Dabei wird die Strategie als boolesche Variable für jeden Knoten im Netzwerk gespeichert.

Abbildung 3.5 zeigt ein Resultat des MoGa für ein Netzwerk mit 20 Knoten. Weiterleitende Geräte sind in grün markiert, nicht weiterleitende in rot. Exemplarisch wurde für einen Knoten der mittlere Senderradius (als schwarzer Kreis) angegeben. Ein Blitz-Symbol kennzeichnet

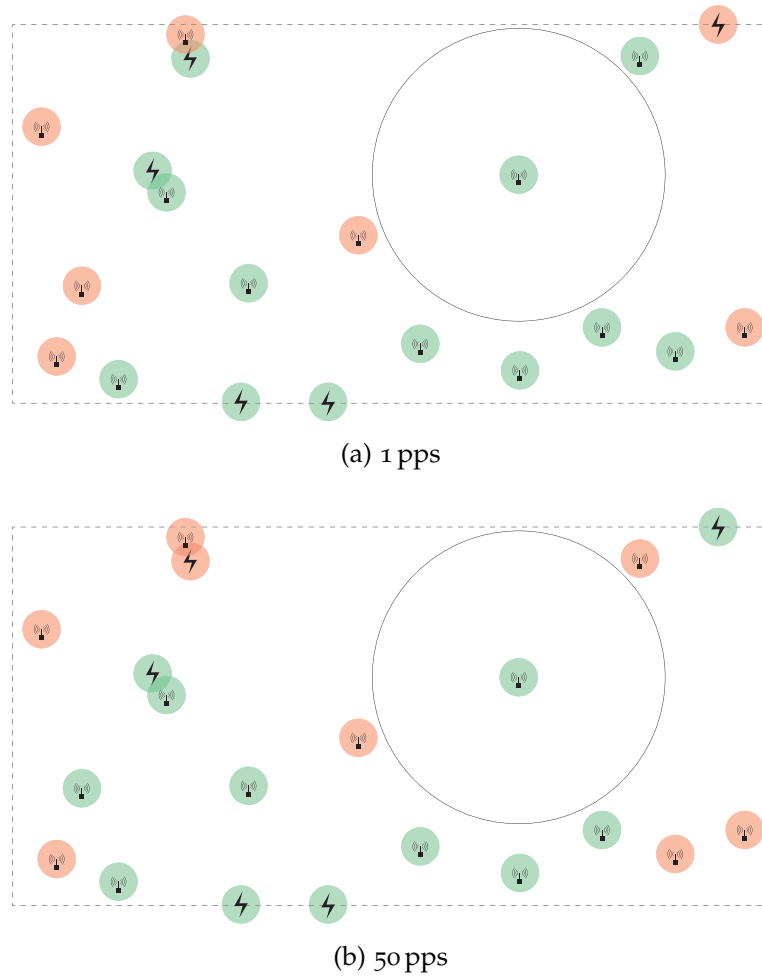


Abbildung 3.5: Ergebnis des evolutionären Algorithmus des Cluster Chromosoms nach 50 Generationen für zwei verschiedene Paketraten. Die PDR wurde mit 90 %, der Energiebedarf mit 10 % gewichtet.

die Netzwerkteilnehmer mit fester Stromversorgung. Die gewählte Gewichtung für die Resultate waren 90 % PDR und 10 % Energieverbrauch. Wie man sieht sind die Ergebnisse für ein Paket pro Sekunde (Abbildung 3.5a) nicht identisch mit dem für eine Emissionsrate von 50 Paketen pro Sekunde (Abbildung 3.5b wie man es bei einem MCDS erwarten würde).

Dieses Verhalten kommt daher, dass in der Realität Paketverluste aufgrund von Paketkollisionen und stochastischem Kanalverhalten auftreten. Dies wird vom einfachen graphentheoretischen Modell nicht erfasst. Eine Erklärung für das Resultat bei niedriger Paketrade ist, dass zusätzliche Knoten bei der Weiterleitung helfen und Paketverluste durch mangelnde Signalstärke ausgleichen können. Da die Zustellrate sehr hoch gewichtet wurde, fällt dieser Effekt mehr ins Gewicht als der zusätzliche Energieaufwand.

Bei der höheren Paketrade kommt die hohe Zahl an Kollisionen zum Tragen. Diese erhöht sich mit jedem weiterleitenden Knoten. Daher erkennt man am Ergebnis (zumindest oben links und unten rechts), dass weniger Knoten weiterleiten als bei der niedrigeren Paketrade. Bei beiden Lösungen leiten Knoten, die sich am Rand des Netzwerks befinden, Pakete nicht weiter.

Tabelle 3.2: Resultate für ein Netzwerk mit 20 Knoten nach der Optimierung mittels ClusterChromosom

	PDR	Leistung ([mW])	Delay [ms]
1 pps			
Verwerfen	13,4	0 (144,6)	0,3
Weiterleiten	67,8	1 (146,9)	3,9
Resultat	70,2	0,58 (145,9)	2,5
50 pps			
Verwerfen	13,3	0 (156,2)	0,3
Weiterleiten	29,7	1 (204,2)	3,9
Resultat	32,5	0,57 (183,6)	3,0

Tabelle 3.2 zeigt die Ergebnisse zu den Simulationsläufen von Abbildung 3.5. Um eine Einordnung der Resultate zu ermöglichen sind ebenfalls die Werte für eine "Flooding"-Strategie (Weiterleiten) und die Grundlinie für keine Weiterleitungen (Verwerfen) gegeben. Die durchschnittliche Leistung bei diesen Versuchsläufen wurde genutzt, um eine Normierung der Resultate vorzunehmen. Damit lässt sich der Energieverbrauch sehr gut mit **Flooding** vergleichen. Wie man sieht benötigt die optimierte Weiterleitungsstrategie weniger Energie, um eine höhere Zustellrate zu erreichen.

3.2.2.2 Nachbarschafts-basiertes Chromosom

Um eine weitere Verbesserung in der Zustellrate bei sinkendem Energieverbrauch und Delay zu erreichen, wurde das im vorherigen Abschnitt beschriebene Weiterleitungsschema erweitert. Für jeden Knoten kann nun festgelegt werden, für welchen seiner Nachbarn er Pakete weiterleitet.

Das Chromosom eines Individuums besteht nunmehr aus einem Vektor, der für jeden Netzwerkteilnehmer wieder einen Vektor von booleschen Variablen speichert. Im ungünstigsten Fall ergeben sich für ein

Netz mit n Teilnehmern $2^{n(n-1)}$ verschiedene Kombinationsmöglichkeiten.

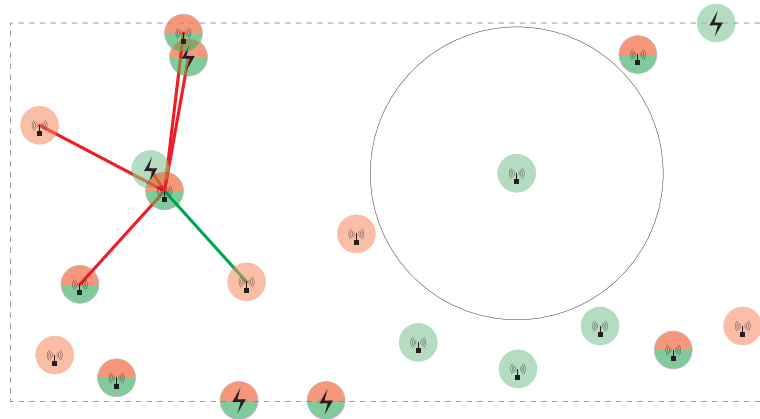


Abbildung 3.6: Ergebnis des evolutionären Algorithmus für das nachbaschaftsbasierte Chromosom nach 50 Generationen für 50 Pakete pro Sekunde. Exemplarisch ist für einen Knoten die Weiterleitungsstrategie dargestellt.

In Abbildung 3.6 ist das Ergebnis der Optimierung einer nachbaschaftsbasierten Weiterleitungsstrategie dargestellt. Knoten die alle Pakete weiterleiten sind grün markiert, komplett verwerfende rot. Findet eine Unterscheidung nach dem Sender des Pakets statt, ist der Netzwerkteilnehmer zweifarbig gekennzeichnet.

Für einen Knoten wurde exemplarisch die Weiterleitungsstrategie grafisch dargestellt. Die Farbe der Verbindungen zu seinen direkten Nachbarn gibt an, ob Pakete von diesen Knoten weitergeleitet werden. Wie man erkennen kann, leitet der Netzwerkteilnehmer, für den die Weiterleitungsstrategie dargestellt ist, kaum Pakete weiter. Diese Aufgabe übernimmt der direkt benachbarte Knoten mit Stromversorgung. Lediglich für den südöstlich gelegenen Nachbarn werden Broadcast-Daten weitergeleitet, da dieser sich gerade außerhalb der Reichweite des nordwestlich gelegenen Knotens befindet.

Die zugehörigen Resultate der Optimierungsläufe sind in Tabelle 3.3 dargestellt. Im Vergleich zum Cluster Chromosom konnte für die niedrige Datenrate eine höhere Zustellrate auf Kosten des Energieverbrauchs erzielt werden. Bei der hohen Datenrate konnte insbesondere der Energiebedarf deutlich gesenkt werden.

3.2.2.3 Verzögerndes Chromosom

In einem letzten Ansatz wird die Dimensionalität des Chromosoms noch einmal gesteigert. Um Weiterleitungsstrategien ähnlich dem **SBA**, **DBS** oder **ABS** Protokoll abzubilden, werden Pakete um ein bestimmtes RAD

Tabelle 3.3: Resultate für ein Netzwerk mit 20 Knoten nach der Optimierung mittels nachbarschaftsbasiertem Chromosom

	PDR	Leistung ([mW])	Delay [ms]
1 pps			
Verwerfen	13,4	0 (144,6)	0,3
Weiterleiten	67,8	1 (146,9)	3,9
Resultat	70,7	0,69 (146,2)	2,5
50 pps			
Verwerfen	13,3	0 (156,2)	0,3
Weiterleiten	29,7	1 (204,2)	3,9
Resultat	32,7	0,42 (176,3)	2,2

verzögert. Dieser Wert wird für jeden Nachbarn und jeden Knoten im Chromosom hinterlegt.

Außerdem wird eine weitere Bedienung für das Weiterleiten eines Pakets hinzugefügt, die auf der Auswertung der innerhalb des RAD empfangenen Paketduplikate beruht:

Leite ein Paket von Nachbarn x weiter, falls es noch nicht von den Nachbarn $y_1, \dots, y_n$ empfangen wurde.

In einem vollvermaschten Netzwerk mit n Teilnehmer ergibt sich alleine aus dieser Bedingung ein Suchraum mit $2^{n(n-1)(n-2)}$ Möglichkeiten. Bei dieser Größe braucht auch ein genetischer Algorithmus eine Vielzahl an Generationen um gute Ergebnisse zu finden. Die Optimierungsstrategie wurde daher als inkrementeller Prozess umgesetzt.

Beide in den vorangegangenen Abschnitten genannten Weiterleitungsstrategien (Cluster (C) und Nachbarschafts (N) basiertes) sind Teilmengen des verzögernden Chromosoms (V):

$$C \subseteq N \subseteq V \quad (3.14)$$

Somit kann ein Chromosom $c \in C$ auch in N dargestellt werden. Zur Optimierung in V wurde daher zuerst über eine bestimmte Anzahl von Generationen in C gesucht und anschließend alle Lösungen in N abgebildet. Daraufhin wurde bis zu einer bestimmten Generation in N optimiert. Danach wurden die Resultate nach V konvertiert und über die verbliebenen Generationen nach optimalen Lösungen gesucht.

Abbildung 3.7 zeigt das Ergebnis nach 200 Generationen. Nach 12 bzw. 60 Generationen wurde der Chromosomtyp gewechselt. Die Weiterleitungsentscheidung von z. B. Knoten 16 in Bezug auf Pakete von 14 lässt sich folgendermaßen interpretieren:

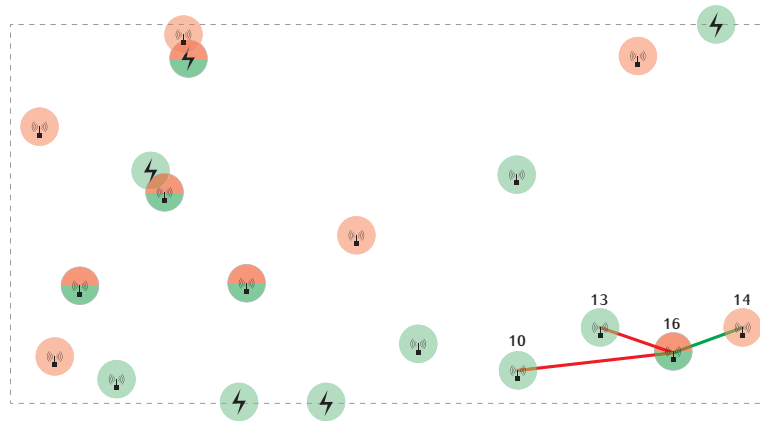


Abbildung 3.7: Ergebnis für das verzögernde Chromosom nach 200 Generationen für 50 Pakete pro Sekunde. Exemplarisch ist für einen Knoten die Weiterleitungsstrategie dargestellt.

Knoten 16 leitet ein Paket von Nachbarn 14 nach 4,9 ms nur weiter, falls es das Paket noch nicht von den Nachbarn 10 und 13 empfangen hat.

Tabelle 3.4: Resultate nach der Optimierung mittels verzögerndem Chromosom

	PDR	Leistung ([mW])	Delay [ms]
1 pps			
Verwerfen	13,4	0 (144,6)	0,3
Weiterleiten	67,8	1 (146,9)	3,9
Resultat	70,3	0,58 (145,9)	9,3
50 pps			
Verwerfen	13,3	0 (156,2)	0,3
Weiterleiten	29,7	1 (204,2)	3,9
Resultat	33,1	0,44 (177,6)	5,3

In Tabelle 3.4 sind die Ergebnisse des Versuchslauf eingetragen. Man erkennt gegenüber den vorherigen Chromosomtypen eine Verbesserung des Energieverbrauchs bei einem Paket pro Sekunde und der PDR bei 50 pps. Allerdings steigt erwartungsgemäß die durchschnittliche Verzögerung.

Tabelle 3.5: Parameter zur Generierung der unterschiedlichen Szenarien für OMNeT++

Parameter	Anzahl	Einstellungen
Statische, mobile Knoten	11	(0,5), (0,10), (0,20), (0,50), (5,0), (5,5), (5,15), (10,0), (10,10), (20,0), (20,30)
Datenrate	5	1, 5, 10, 20, 50
Wiederholungen	5	
Gewichtungen	21	$w_{\text{PDR}} + w_{\text{Energie}} + w_{\text{Delay}} = 100\%$
w_{PDR}		50%-100% in 10% Inkrement
w_{Energie}		0%-50% in 10% Inkrement
w_{Delay}		0%-50% in 10% Inkrement

3.2.3 Ergebnisse des evolutionären Protokolls

Um eine möglichst breite Datenbasis als Trainingsdaten für das spätere maschinelle Lernverfahren zu haben, wurde eine Vielzahl verschiedener Netzwerke und Konfigurationen simuliert. Die Einstellungen für OMNeT++ entsprachen dabei den in Unterabschnitt 2.2.1 spezifizierten.

Tabelle 3.5 fasst die verwendeten Parameter zur Generierung der unterschiedlichen Szenarien noch einmal zusammen. In der Summe wurden somit 275 Optimierungsläufe für alle Kombination aus Datenrate und Knotenanzahl in fünf unterschiedlichen Anordnungen durchgeführt. Die Rechenzeit wurde dabei auf einen Großrechner der Abteilung Nachrichtentechnik der Universität Ulm, sowie auf mehrere Workstations verteilt. Nach ca. 30 Tagen waren alle Simulationen abgeschlossen.

Abbildung 3.8 zeigt exemplarisch für einen Versuchslauf die kontinuierliche Verbesserung über 100 Generationen. Da das Hautaugenmerk auf dem Energieverbrauch und der PDR lag, ist die dreidimensionale Darstellung nur leicht gekippt um den Bezug zu diesen beiden Größen nicht zu verlieren. Es lässt sich die dreidimensionale Paretofront erahnen, gegen die die Ergebnisse der letzten Generation streben.

Am Ende jedes Versuchslaufs steht als Ergebnis des Optimierungsverfahrens eine komplette Population der letzten, sowie deren Eltern aus der vorangegangenen Generation fest. Aus dieser Menge an Chromoso-

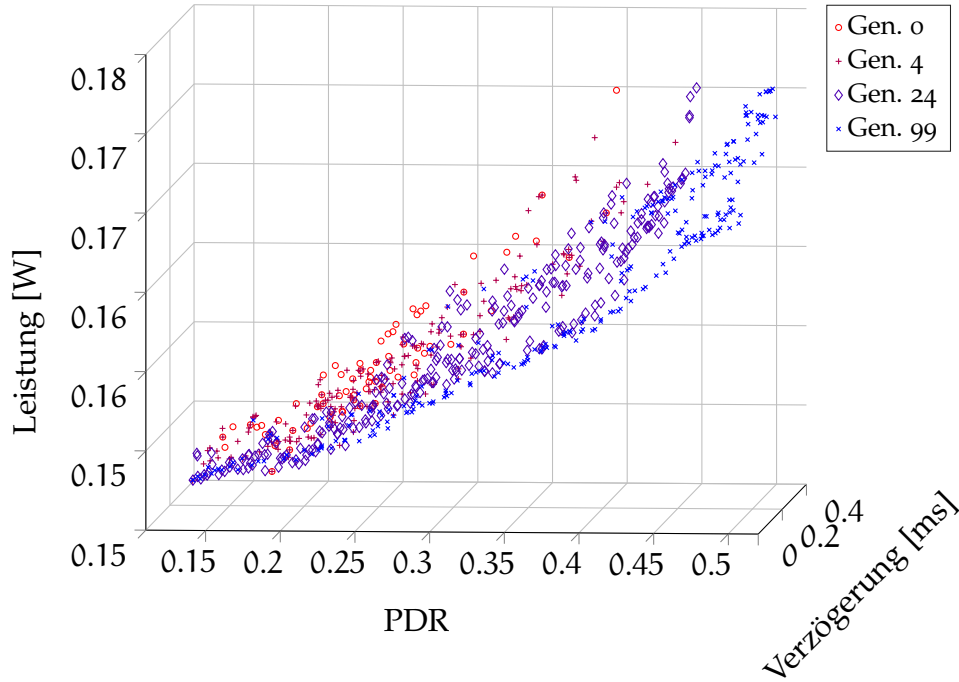


Abbildung 3.8: Verlauf des Optimierungsverfahrens über 100 Generationen.

men werden 21 herausgesucht¹⁸, die eine Fitnessfunktion f maximieren:

$$f = w_{\text{PDR}} \cdot \text{PDR} + w_{\text{Energie}} \frac{P_{\text{flood}} - P}{P_{\text{flood}} - P_{\text{drop}}} + w_{\text{Delay}} \left(\frac{d_{\text{max}} - d}{d_{\text{max}} - d_{\text{drop}}} \right) \quad (3.15)$$

Die Parameter w sind dabei die Gewichte gemäß Tabelle 3.5. PDR, P und d die gemessene Paketverlustrate, Leistung bzw. durchschnittliche Ende zu Ende Verzögerung. Die mit flood gekennzeichneten Werte stammen aus einem vorgelagerten Versuchslauf mit dem **Flooding** Protokoll. Bei drop wurden keine Pakete weitergeleitet. Als maximale Verzögerung wurde d_{max} zu 100 ms angenommen.

Anschließend wurde für diese Ergebnisse erneut eine Simulation durchgeführt. Dabei wurden alle Weiterleitungsentscheidungen, die die Knoten trafen, mitprotokolliert. Sobald ein Paket bei einem Netzwerkgerät eintraf, wurden folgende Daten gespeichert:

- Eine eindeutige Paket ID

¹⁸ Dass nicht alle Ergebnisse ausgewertet wurden, lag im Wesentlichen am benötigten Speicherbedarf. So belegten die Ergebnisse der 21 Auswertungen pro Simulation bereits ca. 37 GB. Da im anschließenden ML Prozess ein bestimmter Punkt auf der Pareto Front gewählt werden musste, wurde diese Auswahl an dieser Stelle bereits vorweggenommen, bzw. eingeschränkt.

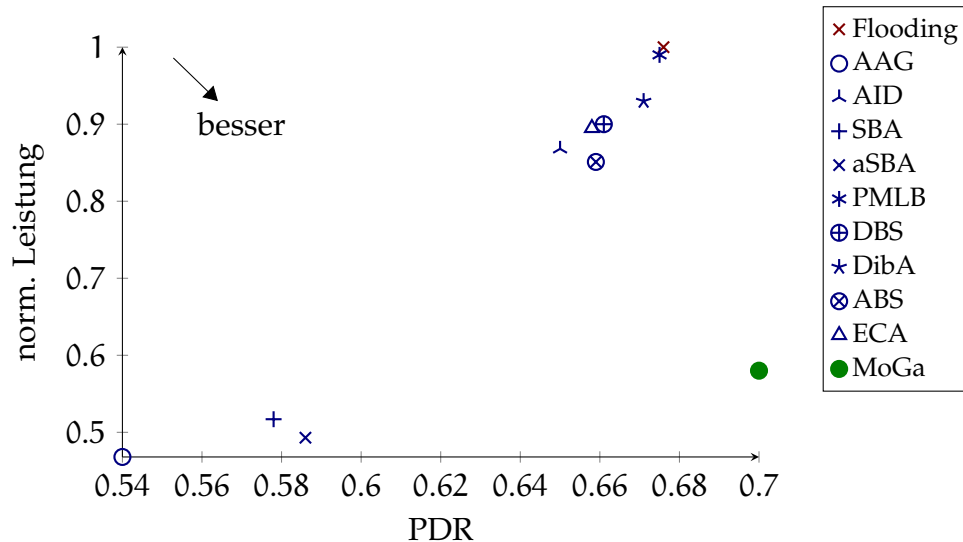


Abbildung 3.9: Exemplarisches Ergebnis für den Vergleich des Stand der Technik mit dem Resultat der Optimierung. Im Netzwerk befanden sich 5 Knoten mit fester Energieversorgung und 15 mit Batterie.

- Empfangender Knoten
- Sender
- Letzter Knoten vor Sender
- Quelle des Pakets
- Paketduplikat
- Signalstärke
- Aktuelle ankommende Paketrade am Empfänger
- Verzögerungszeit
- Weiterleitungsentscheidung

Zusätzlich wurden die Nachbarschaftsbeziehungen und Positionen der Netzwerkteilnehmer für den Versuchslauf gespeichert.

Um zu zeigen, dass diese Weiterleitungsstrategien tatsächlich optimale oder zumindest sehr gute Ergebnisse liefern, wurden die in Unterabschnitt 3.1.2 genannten Protokolle in OMNeT++ implementiert. Für eine Auflistung der für die Protokolle verwendeten Einstellungen sei auf Anhang A.3, bzw. den entsprechenden Abschnitten unter Abschnitt 3.1, verwiesen.

Abbildung 3.9 zeigt einen Vergleich des Stand der Technik mit dem Ergebnis der evolutionären Optimierung. Das zugrunde liegende Netzwerk entspricht dabei dem aus Unterabschnitt 3.2.2. In der Grafik ist die normierte Leistung gegen die PDR, bei einer Paketrade von einem Paket pro Sekunde, aufgetragen. Die Resultate für 50 pps und dem Delay können Tabelle A.2 entnommen werden.

Im Vergleich mit **Flooding** leisten die Resultate des Optimierungsverfahrens bei entsprechender Gewichtung eine höhere Zustellrate bei niedrigerem Energieverbrauch. Ebenso wird im Vergleich zu den Stand der Technik Protokollen eine höhere PDR erreicht. Aufgrund eines Implementierungsfehlers¹⁹ können die Resultate aus den Optimierungsläufen nicht direkt mit denen der existierenden Protokolle verglichen werden. Es wurde jedoch jeweils ein Durchlauf mit **Flooding** vor Beginn des MoGa durchgeführt. Alle Ergebnisse, bei denen die PDR hoch gewichtet wurde, erreichen eine höhere Zustellrate bei gleichzeitig niedrigerem Energieverbrauch als **Flooding**. Bei den Stand der Technik Protokollen wird nur bei hohen Datenraten und Dichten eine höhere PDR erreicht. Dies ist ein starker Hinweis darauf, dass die Ergebnisse der Optimierungsläufe tatsächlich "optimal" sind.

3.3 MASCHINELLES LERNEN

In den vorangegangenen Abschnitten wurde erläutert, wie mit Hilfe eines Netzwerksimulators und einem MoGa eine optimale Weiterleitungsstrategie hinsichtlich mehrerer Gewichtungsparameter entwickelt wurde. Als Ausgabe des Optimierungsprozesses wurden die Weiterleitungsentscheidungen 275 Netzwerkkonfigurationen mit je 21 verschiedenen Kombinationen von Gewichtungsfaktoren mitgeschrieben.

Basierend auf dieser Datenbasis wird in diesem Unterkapitel mit einem maschinellen Lernverfahren ein Entscheidungsalgorithmus entwickelt, der in der Lage ist die Weiterleitungsstrategien aus dem Optimierungsprozess auf unbekannte und mobile Netzwerke zu übertragen. Dazu wird zuerst die Extraktion von "Feature" (Merkmale) aus den einzelnen Entscheidungen beschrieben, mit denen im Anschluss ein maschinelles Lernverfahren trainiert wurde.

Mit Hilfe der unterschiedlichen Gewichtungsfaktoren können, je nach Zielsetzung der Datenübertragung, angepasste Ergebnisse erzielt werden.

¹⁹ Gegen Ende der Promotion wurde festgestellt, dass bei der Implementierung des "Wurde das Paket bereits empfangen"-Kriteriums der KE ein Fehler nicht abgefangen wurde. Damit wurden auch Pakete, die fehlerhaft auf der Medienzugriffsschicht ankamen, als bereits empfangen eingetragen. Bevor diese Daten auf der Transportschicht ankamen, wurden sie von den unteren Schichten bereits verworfen. Korrekt empfangene Duplikate wurden aber ebenfalls nicht behandelt, da fälschlicherweise angenommen wurde, dass bereits ein Duplikat empfangen wurde. In der Konsequenz ist damit die PDR niedriger. Daher wurden für den Vergleich eine Auswahl von Optimierungsläufen noch einmal durchgeführt.

3.3.1 *Extraktion der Feature*

Zur Datenvorbehandlung, dem maschinellen Lernprozess und der Analyse der Ergebnisse wurde die Software "Rapidminer Studio - Community Edition"²⁰ in der Version 6.5.002 verwendet. Rapidminer beschreibt sich selbst als die Nummer 1 unter den quelloffenen Analyse Plattformen. Sie bietet eine Vielzahl an vordefinierten Funktionen (s. g. Operatoren) zur Datentransformation, ML und Datenanalyse. Falls benötigte Funktionen nicht durch vordefinierte Operatoren realisierbar sind, können über eine Skriptsprache eigene Erweiterungen in Rapidminer geschrieben werden. Aus Performancegründen kann die Implementierung eigener Operatoren auch über externe Java-"Plugins" und einer API²¹ erfolgen.

Die Realisierung des Lernverfahrens wurde als zweistufiger Prozess in Rapidminer umgesetzt. Dazu werden die Rohdaten in der Feature Extraktion zuerst analysiert und paketweise in einer externen Datenbank gespeichert. In einem zweiten Schritt werden zufällig Pakete aus der Datenbank entnommen und mit diesen Daten der ML-Algorithmus trainiert und analysiert.

Der Prozessablauf wird bei Rapidminer als grafisches Ablaufdiagramm dargestellt. Abbildung 3.10 zeigt beispielhaft das Diagramm für die Feature Extraktion. Im linken und unteren Teil werden die Rohdaten für einen Versuchslauf und Gewichtungsvektor aus den mitgeschriebenen Protokolldateien geladen. Das sind zum einen die Informationen zu den Netzwerkknoten und deren Nachbarschaftsbeziehungen, zum anderen die Weiterleitungsentscheidungen pro Paket.

Diese Datensätze werden nun an mehrere Unterprozesse weitergegeben, die daraus die in den nächsten Abschnitten erläuterten Features extrahieren (im Prozessablauf ist exemplarisch die Erfassung von Winkelinformationen hervorgehoben). Nachdem diese Vorgänge abgeschlossen sind, werden die Informationen aus den Unterprozessen so zusammengefügt, dass für jedes einzigartige Paket auf jedem Knoten, ein vollständiger Datensatz in einer Datenbank gespeichert werden kann.

3.3.2 *Featurebeschreibung*

Bei der Auswahl der zu implementierenden Features wurde dem maschinellen Lernverfahren eine möglichst große Entscheidungsgrundlage geboten. Daher wurde eine Vielzahl von Merkmalen, die den Algorithmen der existierenden Ansätze zur Verteilung von Broadcast-Daten als Grundlage dienen, in Rapidminer implementiert.

²⁰ Url: <https://rapidminer.com/products/studio/> (abgerufen am 8.3.2016)

²¹ Application Programming Interface (Programmierschnittstelle)

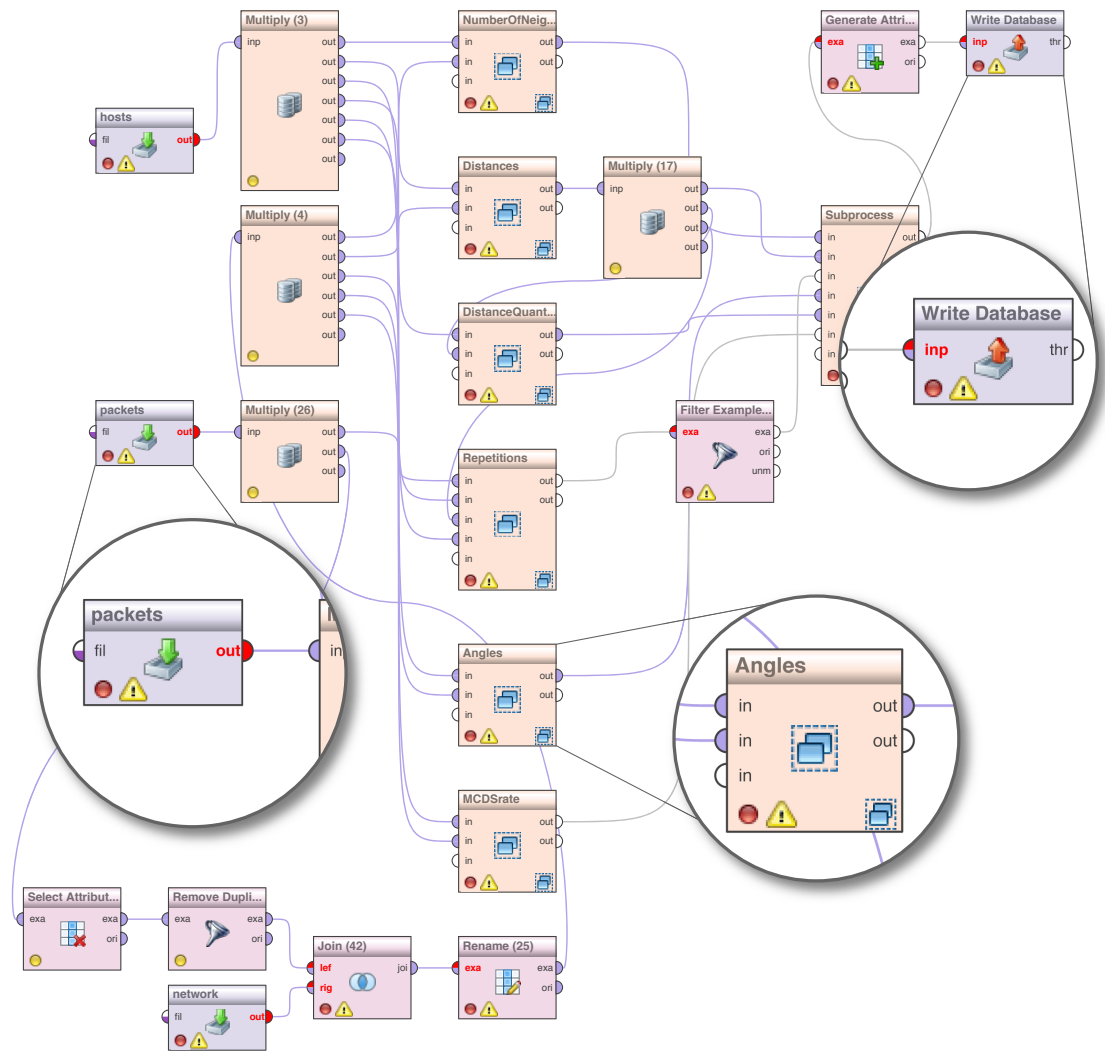


Abbildung 3.10: Übersicht über die Rapidminer Datei zur Extraktion der Attribute.

3.3.2.1 Lokale Informationen

In diesem Abschnitt werden die Features vorgestellt, die lokal auf den Knoten zur Verfügung stehen:

STROMVERSORGUNG Ein boolescher Wert der angibt, ob eine kabelgebundene Stromversorgung vorhanden ist.

DATENVERKEHR Die aktuelle Anzahl an Paketen pro Sekunde.

SIGNALSTÄRKE Gibt die Signalstärke empfangener Pakete an.

ANZAHL DUPLIKATE Anzahl der empfangenen Paketduplikate innerhalb eines RAD.

3.3.2.2 *Informationen direkter Nachbarn*

Durch das Versenden s. g. "Hello"-Pakete können Knoten Informationen über sich an ihre Nachbarn verteilen. Damit lassen sich verschiedene Features ableiten:

ZUSÄTZLICHE ABDECKUNG Wie bei dem Ansatz des **ABS** Protokolls wird mittels eines "Grid Filling" (Gitter/Raster Füllung) Ansatzes der zusätzliche Flächenanteil berechnet, der bei einer Weiterleitung des Paketes abgedeckt wird. Werden innerhalb eines RAD Paketduplikate empfangen, wird der dadurch erreichte Bereich von der Fläche abgezogen.

ABSTAND Der Abstand zum Sender des Paketes.

ABSTANDSPERZENTILE Gibt Auskunft über die Abstandsverteilung zu den Nachbarn. Bei welchem Radius liegen [25,50,75,90]% der Nachbarn innerhalb eines Kreises.

MINIMALER ABSTAND Die minimale Distanz zu den Sendern innerhalb eines RAD empfangener Paketduplikate.

WINKELBEREICHE Um auch geometrische Informationen zu berücksichtigen geben diese Attribute an, welcher Prozentsatz der Nachbarn sich in einem $\pm 45^\circ$, $\pm 90^\circ$ oder $\pm 135^\circ$ großen Kreissegment befinden. Die Bereiche werden dabei in der Verlängerung des einfallenden Signals aufgespannt.

3.3.2.3 *2-hop Informationen*

Versenden die Knoten in ihren "Hello"-Paketen auch Informationen über ihre Nachbarn, können auf den Empfängern Informationen über eine Netzwerkdistanz von 2 Knoten gewonnen werden:

ANTEIL AN DEN MCDS Aus den Nachbarschaftsinformationen kann ein lokaler Netzwerkgraph abgeleitet werden. Berechnet man darin die Menge aller MCDS gibt das Feature an, zu welchem Anteil der Knoten Teil dieser Mengen ist.

MINIMALER GRAD DER NACHBARN

VERBLEIBENDE NACHBARN NACH VERSAND Gibt an wie viele Nachbarn durch den Sender des Pakets noch nicht erreicht wurden.

VERBLEIBENDE NACHBARN Ähnlich wie beim **SBA** Protokoll wird bei jedem Empfang eines Paketduplikats die bereits erreichten Nachbarn abgezogen. Nach Ablauf eines RAD wird die verbleibende Anzahl in diesem Feature hinterlegt.

ANTEIL VERBLEIBENDER NACHBARN Diese zwei Attribute entsprechen den beiden zuvor genannten, geteilt durch die Anzahl der direkten Nachbarn.

ANTEIL 2-HOP NACHBARN Gibt das Verhältnis von 2-hop Nachbarn geteilt durch die direkten Nachbarn an.

3.3.3 Ergebnisse des Maschinellen Lernens

Als Resultat aus der im vorherigen Abschnitt vorgestellten Feature Extraktion stand eine Datenbank mit ca. 57 Millionen Weiterleitungsentscheidungen für drei verschiedene Gewichtungsvektoren für ein ML Verfahren zur Verfügung. Die Gewichtungen ($\vec{w} = [w_{\text{PDR}}; w_{\text{Energy}}; w_{\text{Delay}}]$) wurden dabei so gewählt, dass ein Broadcast-Protokoll mit drei Prioritätsklassen unterstützt werden kann:

WICHTIG Für sehr wichtige Daten wird versucht die höchstmögliche PDR zu erreichen mit $\vec{w} = (1; 0; 0)$.

NORMAL Eine ausgeglichene Gewichtung mit $\vec{w} = (0, 7; 0, 3; 0)$.

HINTERGRUND Für Hintergrunddaten wird der Energieverbrauch höher bewertet mit $\vec{w} = (0, 5; 0, 5; 0)$.

Für alle drei Prioritäten wurde die Ende-zu-Ende Verzögerung mit 0 gewichtet, da die Lösungen aufgrund der Pareto Optimalität bereits sehr gute Ergebnisse hinsichtlich des Delays liefern.

Für jede dieser Gewichtungen wurde ein maschinelles Lernverfahren gesucht, das die Weiterleitungsentscheidungen möglichst genau nachvollzieht. Zu diesem Zweck wurden in einem Kreuzvalidierungsverfahren mehrere Algorithmen getestet. Aus dem Gesamtdatensatz wurden zufällig 100.000 Datensätze gezogen, die für die Kreuzvalidierung genutzt wurden. Der entsprechende Operator in Rapidminer teilt die Daten in k Teilmengen auf (in unserem Fall 10) und verwendet $k - 1$ Mengen zum Trainieren des ML Algorithmus und eine Menge zur Evaluierung. Dieser Vorgang wird für alle k Mengen wiederholt.

Zur Klassifikation der Entscheidungen wurden sechs verschiedene Algorithmen betrachtet: Naive Bayes ([FHT09, S. 210f]), Entscheidungsbäume ([FHT09, S. 308]), "Random Forrest" ([FHT09, S. 587ff]), "Support Vector Machines" ([FHT09, S. 417ff]), automatisches Mehrschicht Perzeptron (AutoMLP [BrSh10]) und künstliche neuronale Netze (ANN [FHT09, 392ff])²².

²² Zur besseren Vergleichbarkeit wurden die Standardwerte von Rapidminer für die Algorithmen übernommen. Für das neuronale Netz wurden zwei Ebenen mit je 10 Knoten definiert. Im Zuge der Evaluation der verschiedenen Verfahren wurden un-

Tabelle 3.6: Ergebnisse der maschinellen Lernalgorithmen

	Genauigkeit	AUC
Entscheidungsbäume	68,8	0,69
Naive Bayes	74,4	0,80
Random Forrest	67,0	0,75
Support Vector Machines	67,0	0,73
AutoMLP	88,1	0,94
ANN	84,1	0,91

Tabelle 3.6 zeigt die Ergebnisse der verschiedenen ML Verfahren. Um diese miteinander vergleichen zu können wurde neben den Werten für die Genauigkeit auch die Fläche unter der ROC²³ Kurve (AUC²⁴ vgl. [Fawco4]) angegeben.

Die besten Ergebnisse liefern die auf neuronalen Netzen basierenden **AutoMLP** und **ANN** Methoden. Jedoch fällt bei diesen Verfahren eine Interpretation des Ergebnisses schwer, da die komplexe Interaktion der Neuronen nicht nachvollziehbar ist. Ebenso können diese Netzwerke leicht übertrainiert werden und erkennen dann einzelne Konfigurationen und nicht den Gesamtzusammenhang.

Im Gegensatz dazu lässt der Naive Bayes Klassifizierer eine Interpretation der Ergebnisse zu (siehe die folgenden Abschnitte). Ebenfalls besteht durch die von Rapidminer gewählte Repräsentation der Wahrscheinlichkeiten als Normalverteilungen keine Gefahr des Übertrainierens. Vielmehr werden die Ergebnisse dadurch geglättet und es können auch bis dato unbekannte Featurekombinationen durch den Entscheider behandelt werden.

Der gewählte Naive Bayes Klassifizierer setzt in der Theorie die Unabhängigkeit der Attribute voraus. Dies ist bei einigen der Features offensichtlich nicht gegeben. Trotzdem können diese Klassifizierer gute Ergebnisse liefern (siehe auch [DoPa96]).

Um das Lernverfahren weiter zu verbessern wurde in einem evolutionären Verfahren die Auswahl der Attribute optimiert. Damit konnte am Ende für die drei gewählten Gewichtungen ein optimaler Satz an Features für den Naive Bayes Klassifizierer bestimmt werden. Die final verwendeten Features sind in Tabelle A.3 aufgelistet.

Nachdem die Optimierung und der Lernprozess für eine Gewichtung abgeschlossen wurde, gibt Rapidminer den gelernten Bayes Klassifizie-

terschiedliche Einstellungen der Operatoren getestet. Diese führten jedoch zu keinen signifikanten Verbesserungen.

²³ Receiver Operating Characteristic Kurve (auch Grenzwertoptimierungskurve)

²⁴ Area Under Curve (Fläche unter der [ROC] Kurve)

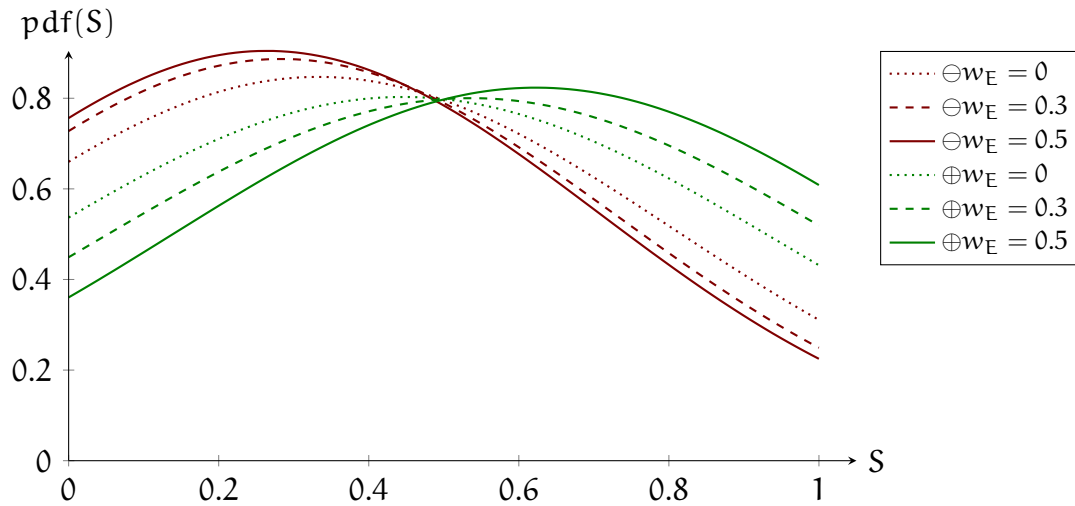


Abbildung 3.11: Ergebnis des Naive Bayes Trainings als Wahrscheinlichkeitsdichte (engl. probability density function - pdf) für das Attribut "Stromversorgung" (S) bei unterschiedlichen Gewichtungen der Energie (w_E).

rer in Form von Normalverteilungen aus. Dabei wird für jedes Feature der Mittelwert und die Standardabweichung für eine positive und negative Weiterleitungsentscheidungen angegeben. Abbildung 3.11 zeigt das Resultat für das Feature "Stromversorgung", das angibt, ob eine externe Stromversorgung verfügbar ist. Man erkennt, dass eine Weiterleitung von Paketen bei vorhandener Stromversorgung umso wahrscheinlicher ist, je höher die Energie gewichtet wurde. Umgekehrt sinkt die Wahrscheinlichkeit wenn diese nicht vorhanden ist.

3.4 ERGEBNISSE

Zum Testen des trainierten Entscheidungsalgorithmus wurde die Software OMNeT++ erweitert. Als Grundlage dienen wie in den vorangegangenen Simulationen UDP Daten, die per Broadcast an das ganze Netzwerk verteilt werden sollen. Um die unterschiedliche Wichtigkeit der Daten abzubilden, werden die Datenpakete mit einer von drei Prioritäten annotiert. Die Entscheidung welche Priorität gewählt wird erfolgt dabei zufällig. Für jede der Klassen kann eine eigene Paketlänge eingestellt werden. Zusätzlich zu dem Protokoll wurden die "Knowledge Engine" (siehe Unterabschnitt 2.2.2) erweitert. Damit stehen alle notwendigen Attribute für die Weiterleitungsentscheidung durch die Bayes Klassifizierer zur Verfügung.

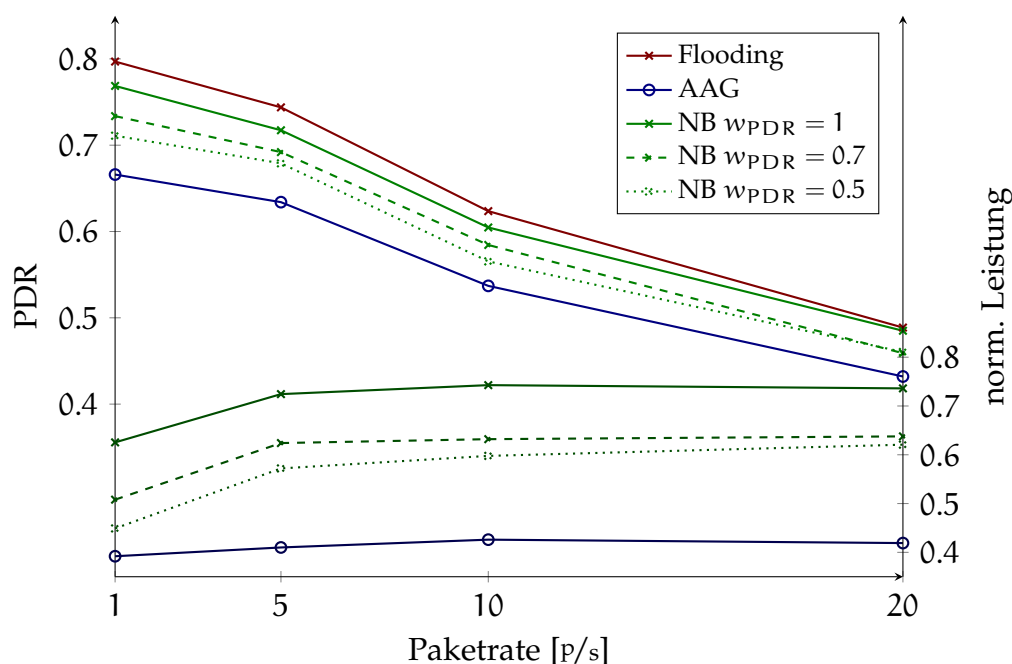


Abbildung 3.12: Mittelwerte von je 5 Versuchsläufen verschiedener Protokolle. Im Netzwerk befanden sich 5 statische und 15 mobile Knoten.

Um der zu Beginn des Kapitels aufgeworfenen Frage nachzugehen, ob die Nachbildung der optimalen Weiterleitungsentscheidungen allgemeingültig ist und auch auf veränderliche Netzwerke angewandt werden kann, wurden die Simulationen für mobile Netzwerke durchgeführt. Dabei wurde wieder eine Unterteilung in statische Knoten mit fester Stromversorgung (die sich entsprechend nicht bewegen) und mobile Knoten mit einer Batterie getroffen. Diese mobilen Teilnehmer bewegten sich im Bereich der Versuchsfläche mit einer Geschwindigkeit zwischen 1,1 und 1,5 m/s.

3.4.1 Vergleich mit dem Stand der Technik

Zunächst wurde auf eine Einteilung in Prioritäten verzichtet, um die Leistungsfähigkeit der gelernten Naive Bayes Protokolle zu zeigen. Abbildung 3.12 zeigt die Ergebnisse für fünf Szenarien mit je fünf statischen und 15 mobilen Knoten. Zur besseren Übersicht werden die Fehlerbalken nicht dargestellt. Das komplette Resultat mit allen getesteten Protokollen kann in Tabelle A.4 auf Seite 128 eingesehen werden. Wie man sieht liefern die Naive Bayes Protokolle sehr gute Ergebnisse, die für die PDR nahe am **Flooding** liegen, jedoch deutlich weniger Energie benötigen. Je nach gewählter Gewichtung verhalten sich die Protokolle erwartungsgemäß. Zum Vergleich mit dem Stand der Technik wurde

das **AAG**-Protokoll mit aufgenommen, das durch seine graphentheoretische Betrachtung einen geringeren Stromverbrauch benötigt, jedoch für alle Paketraten eine geringere PDR erreicht²⁵.

Vergleicht man die auf Naive Bayes basierten mit den in Unterabschnitt 3.1.2 vorgestellten Protokolle (siehe Tabellen A.4, A.5 und A.6) so erreichen sie eine hohe PDR bei gleichzeitig niedrigem Energieverbrauch. Bessere Ergebnisse werden bei höheren Datenraten und Netzwerkdichten durch die adaptive Variante von **SBA**, **aSBA**, erzielt. Dies ist der sehr effektiven Vermeidung von Paketduplikaten durch den graphenbasierten Ansatz von **SBA** geschuldet. Dabei muss jedoch eine sehr hohe Ende-zu-Ende Verzögerung, auch bei niedrigen Datenraten, in Kauf genommen werden. Werden wenige Pakete pro Sekunde gesendet, geht der Algorithmus zu häufig von einem erfolgreichen Empfang auf den Nachbarn aus. Damit sinkt die erreichte Paketzustellrate.

3.4.2 Kombination zu einem prioritätsbasierten Protokoll

Die drei Bayes Protokolle wurden anschließend zu einem prioritätsbasierten Broadcast-Protokoll kombiniert. Dabei entspricht der naive Bays Klassifizierer mit der höchsten Gewichtung für die PDR den wichtigsten Daten. Abweichend zu den vorherigen Versuchsläufen wurde die Paketlänge in Abhängigkeit der Priorität gewählt²⁶. Für Daten hoher (mittlerer, niedriger) Wichtigkeit wurden 64 (192, 1024) byte eingestellt.

Um die Ergebnisse mit den Protokollen des Stand der Technik vergleichen zu können, wurden zwei Protokolle, die in den vorangegangenen Testläufen gute Ergebnisse zeigten, ähnlich den naive Bayes Protokollen, zu zwei Protokollen mit Prioritätsunterstützung kombiniert. Entsprechend wurde für das **AAG** Schema unterschiedliche Erreichbarkeiten (hoch $\hat{=}$ 99 % mittel $\hat{=}$ 90 % und niedrig $\hat{=}$ 75 %) gewählt. Für das **SBA** Protokoll wurden unterschiedliche Zeiten für das RAD eingestellt (hoch $\hat{=}$ 1 ms mittel $\hat{=}$ 3 ms und niedrig $\hat{=}$ 10 ms).

In Abbildung 3.13 sind exemplarisch die Mittelwerte von je 5 Versuchsläufe unterschiedlicher Netzwerke mit je 5 stationären und 15 mobilen Endgeräten dargestellt. Man sieht, dass bei niedrigen Paketraten das Naive Bayes basierte Protokoll sehr gute Zustellraten bei niedrigem Energieverbrauch erzielt. Bei höheren Paketraten sorgen die dynamischen Weiterleitungsalgorithmen von **AAG** und **SBA** für einen geringeren Overhead und somit auch Energieverbrauch. Betrachtet man weitere Szenarien (siehe Tabelle A.7) so bestätigt sich diese Einschätzung.

²⁵ Die für das **AAG** einstellbare Zuverlässigkeit wurde zu 99 % gewählt.

²⁶ Entsprechend dem Aml-Szenario wurde angenommen, dass z. B. Steuerdaten eine hohe Priorität bei gleichzeitig kurzer Paketlänge haben. Daten niedriger Priorität könnten z. B. Delay-tolerante Firmware Updates mit hoher Paketlänge sein.

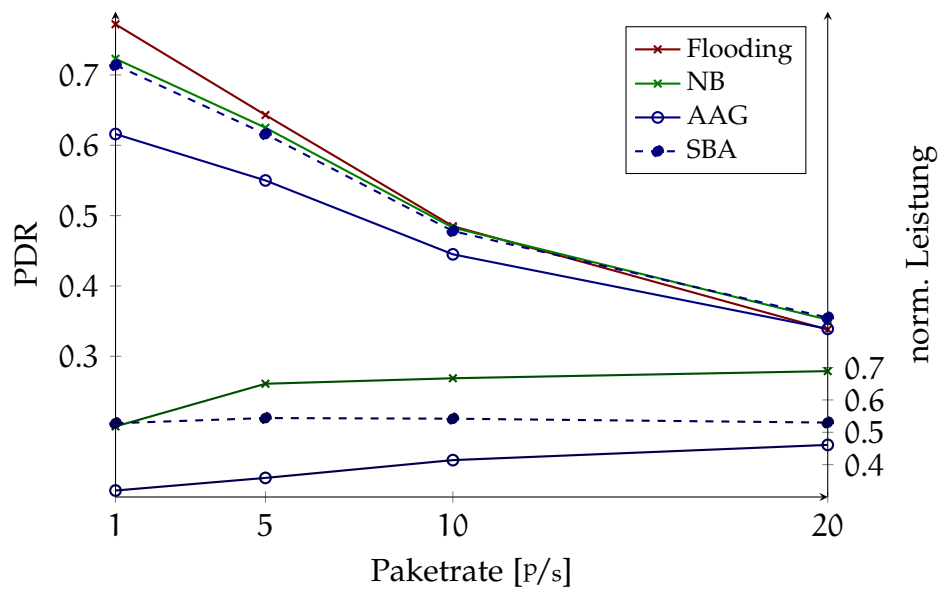


Abbildung 3.13: Mittelwerte von je 5 Versuchsläufen verschiedener prioritäts-basierter Protokolle. Im Netzwerk befanden sich 5 statische und 15 mobile Knoten.

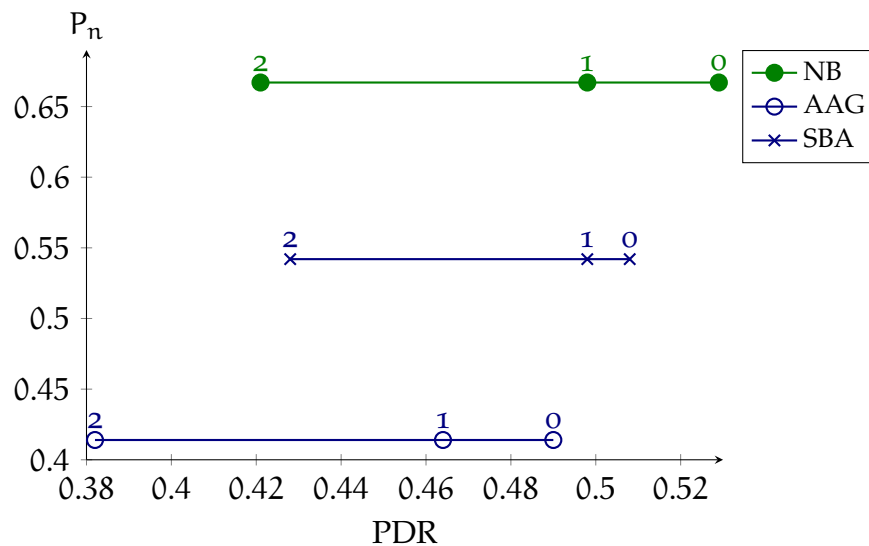


Abbildung 3.14: Mittelwerte der einzelnen Prioritäten von je 5 Versuchsläufen bei 10 Paketen pro Sekunde. Aufgetragen ist die normierte Leistung (P_n) über die PDR. Im Netzwerk befanden sich 5 statische und 15 mobile Knoten.

Zur genaueren Analyse dieses Verhaltens ist das Ergebnis für 10 Pakete pro Sekunde in Abbildung 3.14 für die unterschiedlichen Prioritäten aufgeschlüsselt. Da die Leistungsermittlung durch eine Energiedifferenz über die gesamte Versuchsdauer erfolgt, kann die normierte Leistung nicht nach Prioritäten aufgeschlüsselt werden. Man erkennt, dass für hohe Wichtigkeiten (o) eine größere Zustellrate als bei den anderen Protokollen erreicht werden kann. Für niedrigere Prioritäten hingegen (1,2) wird für die erzielte PDR mehr Energie als bei **SBA** benötigt. Dies ist unter Umständen den längeren Paketen (siehe auch das folgende Unterkapitel) geschuldet und bietet Raum für weitere Verbesserung.

3.4.3 Sensitivitätsanalyse

Bei der Generierung der Trainingsdaten mit Hilfe des evolutionären Algorithmus wurden die Umgebungsparameter über alle Versuchsläufe konstant gelassen. In diesem Abschnitt soll daher untersucht werden, wie sich die Protokolle unter anderen Bedingungen verhalten.

Abbildung 3.15 zeigt die Ergebnisse für wiederum fünf Wiederholungen eines Szenarios mit fünf stationären und 15 mobilen Knoten. Die Datenrate wurde mit fünf Paketen pro Sekunde und die Paketlänge für alle Prioritäten auf 128 byte eingestellt. In Abbildung 3.15a ist zum Vergleich die Basislinie dargestellt.

Wird mit 1280 byte die Paketlänge erhöht, so sinkt erwartungsgemäß die Zustellrate durch die höhere Kanallast und gestiegene Wahrscheinlichkeit für Paketkollisionen. Wie man in Abbildung 3.15b sieht, scheint der Naive Bayes Ansatz darunter stärker zu leiden als die anderen Protokolle. Das gelernte Protokoll verwendet in allen Gewichtungen die ankommende Paketrage als Entscheidungskriterium. Da diese nicht als Maß genügt, um die Kanallast zu schätzen, trifft der Algorithmus nicht ideale Entscheidungen.

Abbildung 3.15c zeigt das Verhalten für höhere Geschwindigkeiten. Dabei bewegen sich die mobilen Knoten mit einer Geschwindigkeit zwischen 11 und 15 m/s. Man erkennt, dass die Leistungsfähigkeit des Naive Bayes Ansatzes erhalten bleibt, jedoch **AAG** und **SSB** eine schlechtere Zustellrate erzielen. Beide Protokolle benötigen 2-hop Informationen, die periodisch per "Hello"-Paketen übertragen werden. Steigt die Geschwindigkeit, veraltet diese Information schnell auf den Geräten und die Algorithmen müssen ihre Entscheidungen anhand nicht mehr aktueller Informationen treffen. Um diese aktuell zu halten, müsste die "Hello"-rate erhöht werden, was wiederum zusätzlichen Overhead bedeuten würde.

Um den Effekt einer höheren Dämpfung, zum Beispiel zur Kommunikation zwischen verschiedenen Stockwerken, zu testen, wurde das

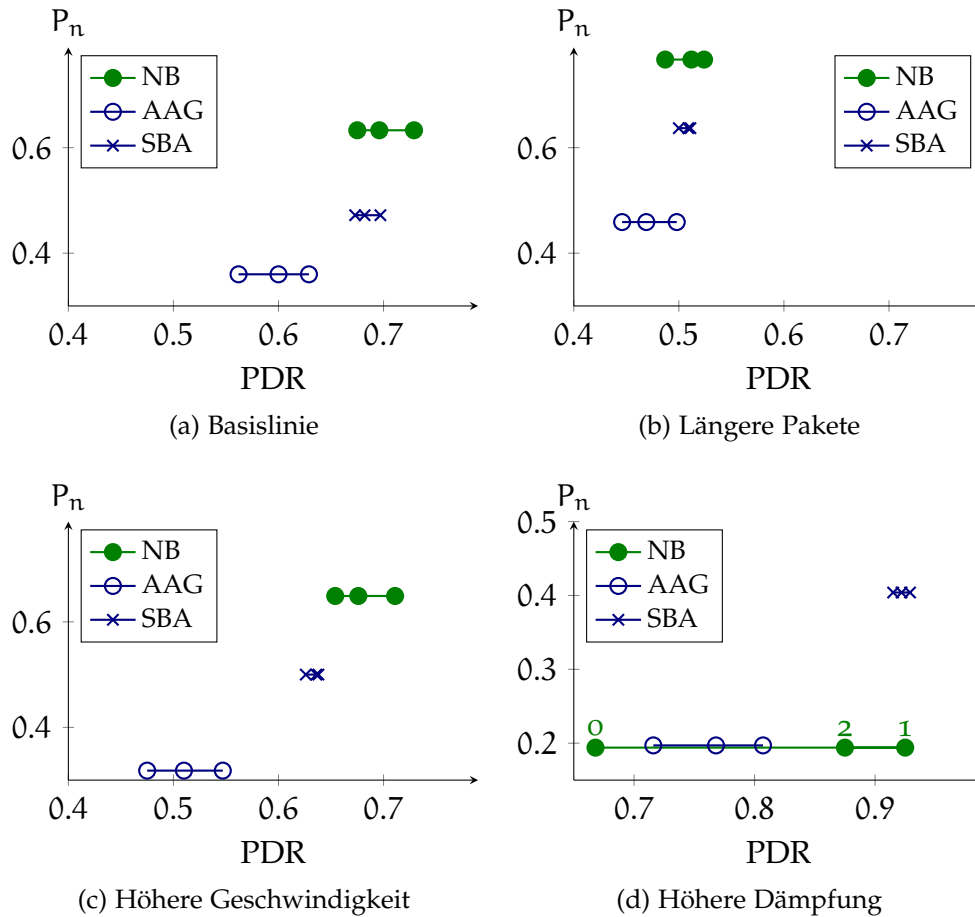


Abbildung 3.15: Ergebnis der Sensitivitätsanalyse für drei von den Trainings-szenarien abweichenden Einstellungen. Aufgetragen ist die normierte Leistung (P_n) über die PDR. Zur besseren Übersicht wurde im letzten Graph eine abweichende Skalierung der Achsen gewählt.

Szenario abgeändert. Die Dämpfungskonstante des Kanalmodells wurde auf 5 und die Fläche auf $60 \times 30 \text{ m}^2$ verkleinert um die verminderte Reichweite auszugleichen. Um die Ergebnisse besser nachvollziehen zu können, wurde die Priorität der korrespondierenden Daten an den Datenpunkten des Naive Bayes Protokolls in Abbildung 3.15d notiert. Wie man sieht erzielt das Protokoll ein sehr gutes Ergebnis für die mittlere (1) und niedrige (2) Priorität. Jedoch wird für die hohe (0) Priorität eine sehr schlechte Zustellrate erreicht. Dieses Verhalten kann durch die verwendeten Attribute des zugrunde liegenden Bayes Klassifizierer erklärt werden. Wie man in Tabelle A.3 sieht, werden bei dieser Priorität sowohl alle Distanzquantile als auch die zusätzliche Abdeckung zur Entscheidung verwendet. Diese Eigenschaften hängen jedoch in hohem Maße

von der angenommenen Dämpfung ab. Daher geht der Entscheider von falschen Voraussetzungen aus und trifft suboptimale Entscheidungen.

3.5 ZUSAMMENFASSUNG

In diesem Kapitel wurde ein Naive-Bayes-basiertes Protokoll entwickelt, das Daten je nach Wichtigkeit zuverlässiger oder energiesparender übermitteln kann. Dazu wurde zu Beginn des Kapitels der aktuelle Stand der Technik mit seinen Ansätzen zur Übermittlung von Broadcast-Daten beleuchtet.

Mit Hilfe eines Mehrzieloptimierungsverfahrens wurde in einem gestaffelten Prozess eine optimale Weiterleitungsstrategie hinsichtlich dreier Kriterien, Zustellrate, Energieverbrauch und Ende-zu -Ende Verzögerung, hergeleitet. In einer Vielzahl von Simulationsläufen wurden für viele verschiedene Szenarien optimiert und die Resultate an 21 Stellen der Paretofront ausgewertet. Dabei wurde jede Weiterleitungsentscheidung inklusive dem aktuellen Kontext mitprotokolliert.

Aus den Protokolldateien und Netzwerkkonfigurationen wurde eine Menge an "Features" extrahiert, die in einer großen Datenbank zusammengeführt wurden. Mit Hilfe dieser Trainingsdaten wurden drei Naive Bayes Klassifizierer trainiert, die unterschiedlichen Gewichtungen der drei Optimierungszielen entsprechen. Ein genetischer Algorithmus zur Auswahl der relevanten Attribute konnte die Klassifizierungsgenauigkeit noch einmal steigern.

Die drei Klassifizierer wurden zu einem Protokoll zusammengefasst, das je nach Priorität der Daten eine hohe Zuverlässigkeit oder eine hohe Energieeffizienz zum Ziel hat. Im Vergleich mit angepassten Protokollen aus dem Stand der Technik zeigt sich, dass das Naive Bayes basierte Protokoll sehr gute Zuverlässigkeit bietet. Jedoch bleibt es bei höheren Datenraten und Knotendichten in der Energieeffizienz für mittlere und niedrige Prioritäten hinter den besten existierenden Ansätzen zurück.

Um die Anfälligkeit des entwickelten Protokolls auf (im Hinblick auf zu den Trainingsdaten) abweichenden Rahmenbedingungen zu testen, wurde eine Sensitivitätsanalyse durchgeführt. Dabei ergab sich gegenüber den existierenden Protokollen eine schlechtere Performance bei längeren Paketen. Eine Änderung der Dämpfungskonstante des Kanalmodells hingegen beeinflusste die mittleren und niedrigen Prioritäten kaum, während die hoch priorisierten Daten massiv schlechtere Zustellraten erzielten. Dies konnte mit der Auswahl der Attributwerte beim Training des Klassifizierers begründet werden. Im Gegensatz zu den Vergleichsprotokollen ist der Naive Bayes Ansatz kaum anfällig für höhere Geschwindigkeiten und konnte seine hohe Zustellrate beibehalten.

Das in diesem Kapitel entwickelte Protokoll leistet insbesondere für hoch priorisierte Daten eine hohe Zuverlässigkeit bei vergleichsweise geringem Energieverbrauch. Für die anderen Prioritäten kann ebenfalls eine hohe Zustellrate erzielt werden, jedoch benötigt das Protokoll in Szenarien mit hohem Datenaufkommen und hohen Knotendichten mehr Energie als das angepasste **SBA** Protokoll, das durch seinen dynamischen Ansatz Paketduplikate effektiver vermeiden kann.

INTELLIGENTE VORWÄRTS FEHLERKORREKTUR

Im vorangegangenen Kapitel wurde ein Broadcast-Protokoll vorgestellt, das, abhängig von der Priorität der Daten, diese mit hoher Zuverlässigkeit bei gleichzeitig niedrigem Energieverbrauch im gesamten Netzwerk verteilen kann. Betrachtet man die Ergebnisse, so fällt auf, dass die Zustellrate auch bei dichteren Netzwerken nicht bei 100 % liegt. Insbesondere bei höheren Paketraten sinkt die erreichte PDR sehr schnell.

Daraus ergeben sich mehrere Fragestellungen, denen in diesem Kapitel nachgegangen werden soll:

Kann die Zustellrate durch ein adaptives Vorwärtsfehlerkorrektur-Verfahren erhöht werden?

Ist ein selbst-lernendes Verfahren in der Lage, basierend auf den Paketprioritäten eine "optimale" Strategie zu entwickeln?

Paketverluste in mobilen Ad-hocNetzwerken können aus mehreren Gründen auftreten:

FEHLER BEI DER DEKODIERUNG Je größer die Distanz zwischen Sender und Empfänger ist, desto schwächer wird das empfangene Signal aufgrund von Dämpfung. Mehrwegeausbreitung und die damit verbundenen Interferenzen können ebenfalls zu geringerer Signalstärke führen. Bei zu geringem Signal-Rausch-Verhältnis können Pakete nicht mehr, oder nur noch fehlerhaft dekodiert werden.

PAKETKOLLISIONEN Versuchen zwei Knoten gleichzeitig über ein geteiltes Medium Daten zu übertragen, so kommt es zu einer Überlagerung der Signale und die Daten können, zumindest innerhalb einer Kollisionsdomäne, nicht korrekt empfangen werden. Bei dem in dieser Arbeit verwendeten Übertragungsprotokoll nach IEEE 802.11 wird versucht, solche Kollisionen durch die Verwendung von CSMA/CA für Broadcast-Übertragungen zu vermeiden. Dennoch versagt dieser Algorithmus bei dem "Hidden Station" (Versteckte Stationen)-Problem und Pakete gehen verloren.

ZU HOHE KANALLAST Wird versucht über das gemeinsame Medium zu viele Daten zu übertragen, nimmt die Wartezeit bis der Kanal nicht mehr belegt ist, stetig zu. Kann ein Broadcast-Paket innerhalb einer gewissen Zeit nicht übertragen werden, wird es aus der Sendewarteschlange entfernt und geht verloren.

Um trotzdem eine hohe Zustellrate erreichen zu können, wird in diesem Kapitel eine Erweiterung zum Naive Bayes basierten Protokoll des vorangegangenen Kapitels, vorgestellt. Diese soll bei gleichzeitig geringem Energiemehrverbrauch die PDR erhöhen. Dazu wird zuerst der Stand der Technik nach existierenden Ansätzen untersucht. Anschließend wird der eigene Lösungsansatz präsentiert. Dieser wird zur Unterstützung von Prioritäten erweitert und die Adaptivität und Energieeffizienz mit Hilfe eines maschinellen Lernverfahrens verbessert.

4.1 STAND DER TECHNIK

Eine der einfachsten Methoden die Zustellrate zu erhöhen ist, Pakete mehrfach zu senden. Dies kann zum einen proaktiv erfolgen. Neben simplen Ansätzen mit festen Wiederholungen und statischen Intervallen gibt es auch adaptive Varianten. In [ABC+12] schlagen Abdou et al. ein Protokoll vor, das unter anderem die Anzahl an Paketwiederholungen und den Abstand zwischen den Wiederholungen mittels eines evolutionären Algorithmus "offline" optimiert. In Abhängigkeit von der lokalen Netzwerkdichte wird zur Laufzeit ein bestimmter Parametervektor aus den Resultaten gewählt.

Zum anderen kann die Übertragung von Paketwiederholungen davon abhängig gemacht werden ob das Paket korrekt auf allen Nachbarknoten empfangen wurden. Neben der Bestätigung des Empfangs durch die Empfänger mittels "ACK" Paketen existieren auch Ansätze, die versuchen, durch passives "Mithören" auf den erfolgreichen Empfang zu schließen. In [SBDK10] stellen die Autoren einen "Message loss avoidance" (Vermeidung von Paketverlust) genannten Algorithmus vor. Dieser zählt nach einer Übertragung die Anzahl von weiterhin empfangenen Paketduplikaten. Unterschreitet diese einen definierbaren Wert C so wird angenommen, dass die Daten nicht erfolgreich übertragen wurden. Nach einer Wartezeit d wird das Paket erneut versendet. Dieser Vorgang wird bis zu k mal wiederholt.

Einen auf Bestätigungspaketen basierenden Ansatz, das **ABSM** Protokoll, stellen Ros et al. in [RRS12] vor. Das zugrunde liegende Broadcast-Protokoll basiert auf einem graphentheoretischen Ansatz mittels MCDS. Die Zuverlässigkeit wird bei **ABSM** erreicht, indem "ACK" Nachrichten in den periodischen "Hello" Paketen mitgeschickt werden. Jeder Knoten hält für jedes Paket eine Liste vor, in der alle seine Nachbarn eingetragen sind. Bestätigt ein Nachbar den Empfang des Paketes wird er aus der Liste entfernt. Ist nach Ablauf eines Zeitintervalls diese Liste nicht leer, wird das Paket erneut versandt. Aufgrund der hohen Verzögerung eignet sich das Protokoll nur für delay-tolerante Daten.

Viele Ansätze zur zuverlässigen Datenübertragung finden sich im Bereich der Multicast-Übertragungen. Im Gegensatz zu Broadcast-Protokollen basieren solche für Multicast auf der Konstruktion von Bäumen oder Pfaden zwischen der Quelle und den Zielen. Dennoch können manche Ansätze zur Vermeidung von Paketverlust auch auf Broadcast angewandt werden.

In [XWZ10] stellen die Autoren mit **R-ODMRP** eine Erweiterung des **ODMRP** (vgl. [LSGo2]) vor. Bei diesem Mesh-basierten Ansatz werden Cluster gebildet, um die Daten effizient zu übertragen. Mit Hilfe eines applikationsspezifischen Grenzwerts η_0 kann die gewünschte Zuverlässigkeit der Übertragung eingestellt werden. Dieser Grenzwert wird verwendet um zu überprüfen, ob eine ausreichende Anzahl an Nachbarn das Paket quittiert hat. Andernfalls wird es erneut versendet. Liegt das Paket nicht mehr auf dem Knoten vor, kann es erneut von einer der näher an der Quelle liegenden Stationen angefordert werden.

Im Gegensatz zu diesen reaktiven Ansätzen erlauben proaktive Ansätze auch die Übertragung von Echtzeitdaten. In [Rube98] stellen die Autoren einen der ersten Ansätze zur zuverlässigen Multicast-Übertragung mit Hilfe von Vorwärtsfehlerkorrektur (FEC¹) vor. Dabei werden die Pakete blockweise zusammengefasst und eine einstellbare Zahl von Redundanzpaketen erzeugt. Mit Hilfe eines "NACK" (Not ACKnowledge - nicht bestätigt) Mechanismus können die Empfänger zur Dekodierung benötigte Pakete erneut anfordern. Die Wiederherstellung von Paketen und die Aufforderung zum erneuten Senden finden nur auf den letztendlichen Multicast-Empfängern statt. Ebenso generiert nur der Sender Redundanzpakete.

Einen ausgefeilteren Ansatz für drahtlose Niedrigenergie-Netze stellen Li et al. mit **ASM** in [LXHS12] vor. Das Routing der Pakete wird dabei von einem Baum-basierten, opportunistischen Multicast-Protokoll übernommen. Innerhalb des Baumes werden Informationen über die Paketverlustrate zurück Richtung Quelle übermittelt. Jeder Knoten des Baumes leitet nur die höchste PLR² weiter. Mit diesen Informationen kann die Paketquelle die minimale Anzahl an Redundanzpaketen errechnen, um einen bestimmten Datenverlust im Mittel nicht zu überschreiten. Auf dem Weg von Quelle zu den Empfängern können die Weiterleiter ebenfalls für den nachfolgenden Teilast die benötigte Zahl an Redundanzpaketen ermitteln. Zur Reduktion des Datenverkehrs und damit der benötigten Energie werden somit überflüssige Pakete nicht weitergeleitet.

Das **ASM** Protokoll sollte durch diese astweise Behandlung der Redundanzpakete gut mit einem Multi- und Broadcast-Problem der zuverlässigen

¹ Forward Error Correction (Vorwärtsfehlerkorrektur)

² Packet Loss Rate (Paketverlustrate)

lässigen Datenübertragung zurechtkommen - den “Crying Babies” (**CB** - Schreiende Babys) (vgl. [HSC95], [KHW09]). Dieses Problem tritt auf, wenn zu einem Empfänger eine besonders schlechte Verbindung besteht. Der Versuch dessen Zuverlässigkeitsanforderungen zu erfüllen kann in einer Verschlechterung des Empfangs der restlichen Empfänger führen (z.B. durch zusätzliche Paketkollisionen eines ausgelasteten Kanals).

Betrachtet man den Stand der Technik, so gibt es nur wenige Ansätze, die die Wichtigkeit der Daten, bzw. applikationsspezifische Anforderungen berücksichtigen. Davon finden sich viele im Forschungsfeld der Multicast-Übertragungen. Die dafür genutzten Baumstrukturen, bzw. das netzwerkweite Wissen über die Struktur des Netzwerkes liegt bei Broadcast-Protokollen selten vor.

Energieeffizienz steht bei den meisten Ansätzen im Hintergrund. Dabei können adaptive Protokolle wie **ASM** gleichzeitig dem **CB**-Problem entgegenwirken und die zusätzlich benötigte Energie minimieren.

Im Kontext der energieeffizienten Broadcast-Übertragungen für Aml wäre hingegen ein Algorithmus erstrebenswert, der möglichst energieeffizient agiert. Insbesondere soll er kontextabhängig entscheiden, welcher Grad an Redundanz nötig ist, um Pakete mit hoher Zuverlässigkeit an seine Nachbarn zuzustellen. Dabei sollen auch die Applikationsanforderungen, bzw. die Priorität der Pakete berücksichtigt werden. Wie bereits diskutiert, stehen netzwerkweite Informationen über die Struktur, PLR, etc., nicht zur Verfügung. Der Stand der Technik scheint für diese Anforderungen keine Lösung parat zu halten.

4.2 ENTWURF UND REALISIERUNG

Zur Steigerung der Zuverlässigkeit bei der Übertragung von Broadcast-Daten, wurde ein effizientes Protokoll entwickelt, das die Priorität der Pakete berücksichtigt. Als Entscheidungsgrundlage stehen ihm dabei nur lokale, bzw. den per “Hello-Paketen” übertragenen Informationen der Nachbarn zur Verfügung.

Um eine realistische Einschätzung des Laufzeitverhaltens und der zu erwartenden Ende-zu-Ende Verzögerung zu erhalten (insbesondere in Hinblick auf die geplante prototypische Implementierung in Kapitel 5), wurde die Entwicklung und Evaluation auf einem realen Testbed durchgeführt. Dies bestand aus mehreren Smartphones auf Androidbasis des Typs “Samsung, Galaxy Nexus”. Als Entwicklungsframework kam die Software **Adhoc3** (vgl. Abschnitt 2.3) zum Einsatz.

4.2.1 Adaptive Vorwärtsfehlerkorrektur mittels SPC-2D

Zum Test der Plattform und als erster Schritt hin zu einer möglichst adaptiven Fehlerkorrektur, wurde ein Protokoll auf Basis des zweidimensionalen SPC³ Korrekturverfahrens (vgl. [PBS14]) implementiert. Der vollständige Ansatz, Simulationsergebnisse und die Ergebnisse der realen Versuchsläufe wurden in [PBS14] veröffentlicht.

Der wichtigste Parameter des **SPC-2D** Korrekturverfahrens ist die Dimension der für das De- und Encodieren verwendeten Matrix $R(l, m)$. Um die erreichbare Zustellrate besser abschätzen zu können, wurde in Simulink der Firma Mathworks ein kleiner Simulator umgesetzt. Dieser generiert einen Datenstrom mit konstanter Bitrate, der in Pakete konstanter Größe aufgeteilt wird. Aus dem Paketstrom werden mit dem **SPC-2D** Verfahren für je $l \cdot m$ Datenpakete $l + m$ zusätzliche Redundanzpakete erzeugt. Diese werden über einen simulierten Kanal verschickt, der in einem Bernoulli-Prozess Pakete mit einer einstellbaren Verlustwahrscheinlichkeit an einen Empfänger zustellt. Dort werden fehlende Pakete, sofern möglich, wiederhergestellt.

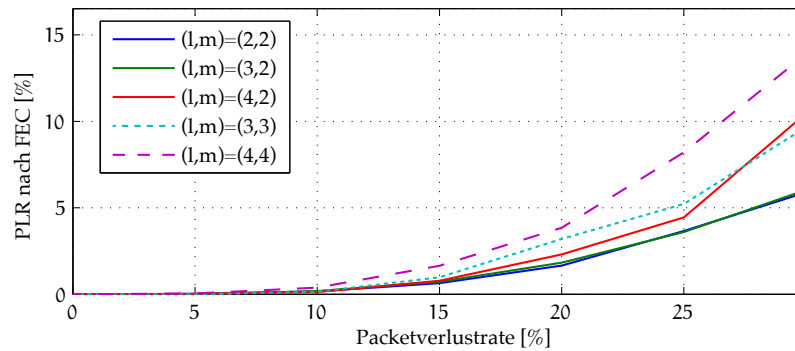


Abbildung 4.1: Ergebnis der Simulationsläufe für das **SPC-2D** Vorwärtsfehlerkorrekturverfahren. Aufgetragen ist die eingestellte Verlustrate des simulierten Kanals und die resultierende nach der Wiederherstellung bei verschiedenen Matrixdimensionen.

In Abbildung 4.1 ist das Ergebnis dieser Versuchsläufe, die Verlustrate nach der Wiederherstellung der Pakete über der eingestellte Verlustrate für unterschiedliche Matrixdimensionen, dargestellt. Man sieht, dass erwartungsgemäß die Einstellungen mit dem höchsten Redundanzgrad

$$r = \frac{\text{\#Redundanzpakete}}{\text{\#Datenpakete}} \quad (4.1)$$

mit $l, m = (2, 2)$ die niedrigste PLR nach der Wiederherstellung erreicht.

³ Single Parity Check (Einfache Paritätskontrolle)

Betrachtet man eine festgelegte Verlustrate, die im Mittel nicht überschritten werden soll, so lässt sich für jede Matrixdimension ein Wert finden, bis zu dem diese verwendet werden kann⁴. Aus diesem Zusammenhang wurde ein einfacher Algorithmus hergeleitet, der rein auf der Verlustrate auf Applikationsebene (**aPLR**) basiert. Diese wird von den Empfängern periodisch mit den "Hello"-Paketen an den Sender zurückgemeldet. Entsprechend dem Algorithmus in Abbildung 4.2 und

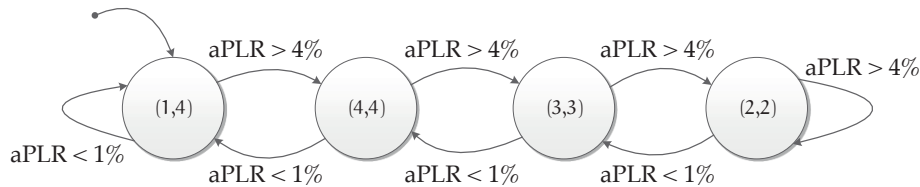


Abbildung 4.2: Zustandsautomat für den Algorithmus zur Wahl der Matrixgröße.

dem Maximum der Verlustrate wählt der Sender eine neue Matrixgröße. Steigt die aPLR über 4 % wird der nächsthöhere Redundanzgrad ausgewählt. Verbessert sich die Verlustrate auf Applikationsebene wieder und fällt unter 1 % wird die vorherige Matrixgröße gewählt. Als kleinstmögliche Dimension wurde der einfache SPC-Algorithmus mit 4 Daten- und einem Paritätspaket verwendet. Damit können kleinere Paketausfälle durch Kollisionen ausgeglichen werden, ohne dass qualitative Abstriche gemacht werden müssen.

Um das Verfahren zu testen wurde eine 80 m lange Strecke entlang eines Verbindungsgangs der Universität Ulm abgesteckt. Obwohl der Versuch im Freien stattfand, wurde die Übertragung durch das Universitäts-eigene Wlan Netz gestört. Ebenso sind Interferenzen durch die baulichen Strukturen aufgetreten. An das eine Ende der Strecke wurde der Sender platziert. Das Empfangsgerät wurde in der Hand gehalten während die Strecke abgeschritten wurde. Am anderen Ende angekommen, wurde der gleiche Weg wieder zurück gegangen.

⁴ Nimmt man z. B. 5 % erlaubte Verlustrate, so ergeben sich folgende Zuordnungen (mit zunehmenden Redundanzgrad):

$$R = \begin{cases} (1,1) & \text{für } (0 \leq \text{plr} < 5) - \text{keine Redundanz} \\ (1,4) & \text{für } (5 \leq \text{plr} < 12) - \text{SPC} \\ (4,4) & \text{für } (12 \leq \text{plr} < 22) \\ (3,3) & \text{für } (0 \leq \text{plr} < 25) \\ (2,2) & \text{für } (0 \leq \text{plr} < 27) \end{cases}$$

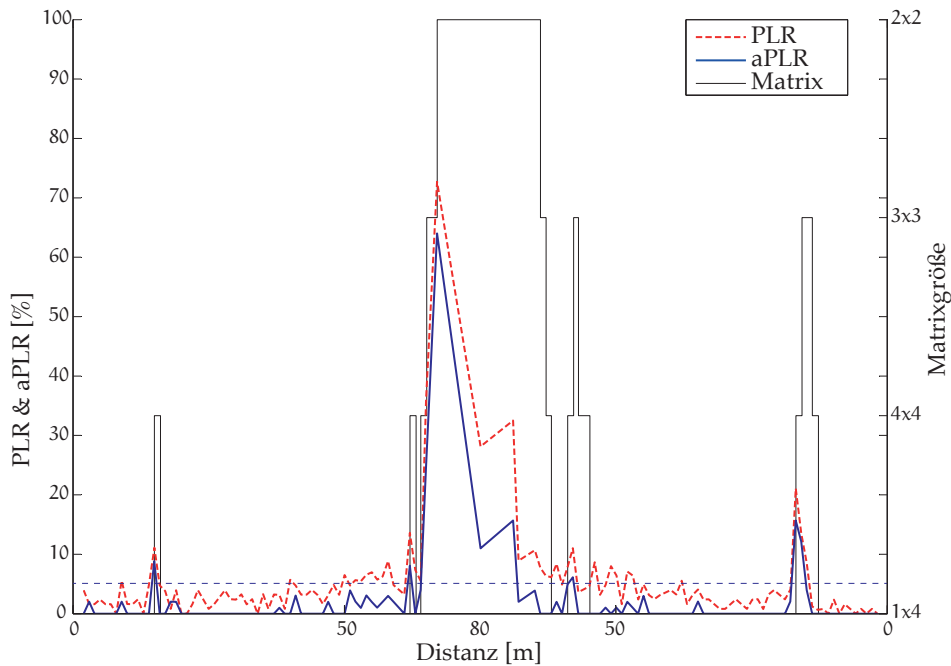


Abbildung 4.3: Versuchsergebnis einer adaptiven Vorwärtsfehlerkorrektur mit dem **SPC-2D** Verfahren. Dabei wurden die Verlustraten auf Applikations- (aPLR) und Netzwerkebene (PLR) gemessen und die Größe der Korrekturmatrix (Matrix) mitgeschrieben. Eine Distanz von 80 m wurde hin und zurück abgeschritten.

Abbildung 4.3 zeigt das Ergebnis dieses Versuchslaufs. Das adaptive Korrekturverfahren schafft es die meiste Zeit, die aPLR unter 5 % zu halten. Steigt die Verlustrate, wählt der Algorithmus eine Matrixgröße mit mehr Redundanz. Zugleich wird weniger Redundanz gewählt, wenn sich die Empfangsbedingungen verbessern. Man sieht jedoch auch, dass der Algorithmus oft zu langsam reagiert. So steigt die Verlustrate bei ca. 15 m (sowohl beim Hin- als auch beim Rückweg) so plötzlich, dass der nötige Redundanzgrad nicht erreicht wird bevor der Bereich mit schlechten Empfangseigenschaften schon wieder verlassen wurde. Dies ist im wesentlichen der Frequenz, mit der "Hello"-Pakete gesendet wurden (1 Hz), geschuldet. Eine Erhöhung würde zwar die Reaktionsgeschwindigkeit des Algorithmus verbessern, zugleich steigt aber die Kanallast und die Anzahl der Pakete, über die die Verlustraten bestimmt werden, nimmt ab. Damit steigt das Rauschen der aPLR und der Algorithmus trifft häufiger nicht ideale Entscheidungen.

Ebenso sieht man auch die Grenzen des Korrekturverfahrens. Bei ca. 65 m verschlechtern sich die Empfangsbedingungen rapide. Damit können zum Einen "Hello"-Pakete des Senders nicht empfangen werden

und dem Algorithmus fehlt die Entscheidungsgrundlage. Zum Anderen ist es oberhalb einer PLR von ca. 27 % nicht mehr möglich, mit dem **SPC-2D**-Verfahren eine $aPlr < 5 \%$ zu erreichen.

4.2.2 *Selbstlernende Vorwärtsfehlerkorrektur mit Reed-Solomon*

Im vorherigen Abschnitt konnte gezeigt werden, dass ein adaptives Fehlerkorrekturverfahren auf der Versuchsplattform verwendet werden kann und eine einstellbare maximale PLR auf Applikationsebene mit dem gewählten Ansatz die meiste Zeit unterschritten wird. Eines der größten Probleme des gewählten Verfahrens ist seine Trägheit. So ist der Algorithmus nicht in der Lage, proaktiv auf die Netzwerkbedingungen zu reagieren, sondern muss zuerst auf die Rückmeldung über die erzielte Zustellrate warten. Die fixen Grenzwerte der Hysterese und dass nur auf die jeweils angrenzende Matrixgrößen gewechselt werden kann, verlangsamt die Reaktion bei größeren Schwankungen der Empfangsbedingungen weiter.

In den vorangegangenen Versuchen wurden alle Pakete gleich behandelt. Um eine unterschiedliche Priorität der Daten zu unterstützen, wurde der Ansatz erweitert. Wie bei den Untersuchungen in Kapitel 3 werden drei Prioritäten unterstützt: **Niedrig**, **Mittel** und **Hoch**. Für jede dieser Kategorien kann ein eigener Redundanzgrad eingestellt werden.

Um das System dynamischer und flexibler zu machen wurden die starren Übergänge zwischen den Zuständen aufgegeben. Stattdessen kommt ein ähnlicher Ansatz wie in Unterabschnitt 3.3.3 zum Einsatz. Dabei versucht ein selbst lernendes System aus dem aktuellen Geräte- und Netzwerkkontext die optimale Konfiguration zu finden.

Eine weitere Verbesserung betrifft das Vorwärtsfehlerkorrekturverfahren, das durch das effizientere Reed-Solomon-Verfahren (siehe [ReSo60]) ersetzt wurde. Bei diesem werden für k Datenpakete insgesamt $n > k$ Pakete übertragen⁵. Werden davon k Pakete empfangen, können die Originaldaten vollständig wiederhergestellt werden.

4.2.2.1 *Versuchsaufbau und Aufnahme der Grundlinie*

Ähnlich wie in Unterabschnitt 4.2.1 wurde das Framework **Adhoc3** zur Implementierung und Durchführung der Experimente genutzt. Als Testbed kamen vier Android basierte Smartphones "Galaxy Nexus" von Samsung zum Einsatz. Um mehr Kontrolle über die Verlustraten zwischen

⁵ Der Reed-Solomon-Code wurde so implementiert, dass für je k Pakete $n - k$ Redundanzpakete generiert werden. Diese werden gebildet, indem RS^6 byteweise auf die Datenpakete angewandt wird. Sollten diese kürzer als das längste Paket sein, werden sie mit Nullen aufgefüllt.

den Geräten zu haben, wurde auf Freifeld-Versuche verzichtet und der Paketverlust in den Empfangseinheiten simuliert. Dazu kann auf jedem Gerät eine Wahrscheinlichkeit eingestellt werden, mit der ankommende Pakete verworfen werden.

Um weitere Paketverluste zu minimieren wurden die Geräte nahe beieinander angeordnet. Damit blieben als Störeinflüsse nur das Universitäts-Wlan und die unkontrollierten Paketeinschleusungen bei den Versuchen des Bürokollegen. Diese wurden durch die Wahl einer möglichst freien Übertragungsfrequenz minimiert.

Die Software **Adhoc3** wurde um ein Sende- und Empfangsprotokoll erweitert, das den Reed-Solomon Code implementiert. Dabei wurden drei Prioritäten (Niedrig, Mittel und Hoch) vorgesehen, denen unterschiedliche n und k zugewiesen werden können. Um möglichst wenig Latenz bei der En- und Dekodierung zu verursachen, bzw. eine Verarbeitung in Echtzeit überhaupt erst zu ermöglichen, wurde der Reed-Solomon Code als "natives" Programm in C geschrieben. Dabei wurde die Bibliothek "zfec"⁷ von Zooko O'Whielacronx als Basis benutzt und um eine Anbindung per "jni" an Android erweitert.

Als Datenquelle wurden ein oder zwei Datenströme konstanter Bitrate generiert. In Anlehnung an ein Audioübertragungssystem (siehe Kapitel 2) wurde mit 4 kbps ein hochpriorer Datenstrom und ein mittelpriorer mit 128 kbps erzeugt. Den restlichen Daten, wie z. B. "Hello"-Paketen, wurde die niedrigste Priorität zugeordnet. Die Datenströme wurden mit 50 pps paketisiert, mit Steuer- und Kopfdaten versehen und über das Netzwerk übertragen. Als guter Kompromiss aus zusätzlicher Latenz und Flexibilität bei der Wahl der Redundanz erwies sich in den Versuchen ein k von 6. Dies wurde bei allen nachfolgenden Versuchen verwendet.

Um die Funktionsfähigkeit des Systems zu testen und einen ersten Eindruck von der Leistungsfähigkeit von RS zu bekommen, wurden Grundlinien für den Datenstrom mittlerer Priorität aufgenommen.

Abbildung 4.4a zeigt diese für ein einfaches Netzwerk bestehend aus einem Sender und einem Empfänger. Für unterschiedliche Verlustwahrscheinlichkeiten ($p = 0, 5, 10, 25$ und 50%) und n ($6 \leq n \leq 12$) wurden die resultierende PDR nach dem Fehlerkorrekturverfahren bestimmt. Die theoretischen Werte wurden mit der Wahrscheinlichkeit, dass ein Paket ankommt ($1 - p$) und - falls es nicht empfangen wurde - wiederhergestellt werden kann ($P_{\oplus}|\text{nicht empfangen}$), berechnet:

$$\alpha\text{PDR} = (1 - p) + p(P_{\oplus}|\text{nicht empfangen}) = (1 - p) + p \sum_{i=k}^{n-1} \binom{n-1}{i} (1 - p)^i p^{(n-1-i)} \quad (4.2)$$

⁷ Url: <https://tahoe-lafs.org/trac/zfec/> (abgerufen am 14.4.2016)

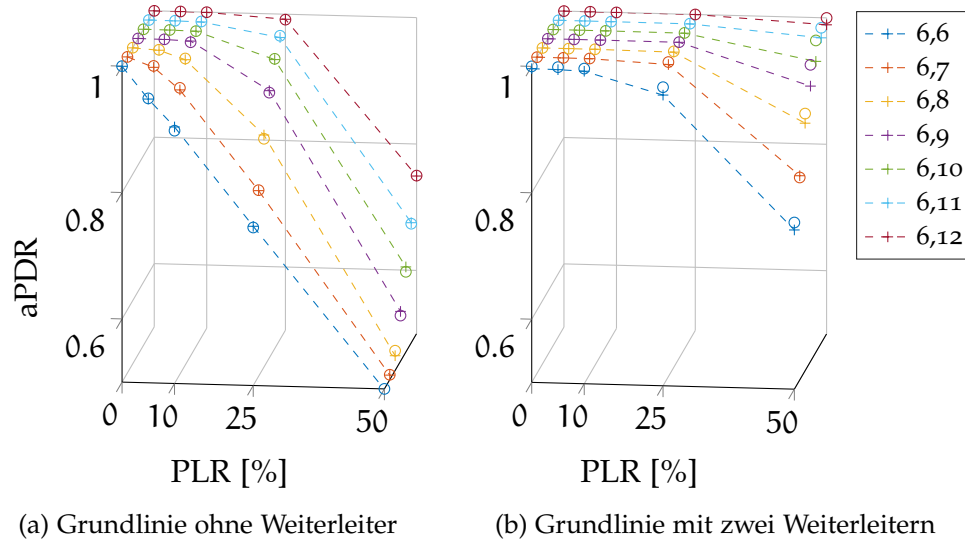


Abbildung 4.4: Grundlinie der PDR auf Applikationsebene (aPDR⁸) für simulierte Paketverluste (mit Wahrscheinlichkeit PLR) nach Anwendung des Reed-Solomon Verfahrens mit verschiedenen (k, n) . Für die Messungen kam das Testbed auf **Adhoc3**-Basis zum Einsatz. Die theoretischen Werte sind mit $\circ$ gekennzeichnet.

In Abbildung 4.4b ist die Empfangsrate auf einem Empfänger in einem Netzwerk mit zwei Weiterleitern dargestellt. Diese implementieren ebenfalls das RS-Verfahren und versenden wiederhergestellte Pakete an ihre Nachbarn. Sowohl für den Sender als auch für die zwei zusätzlichen Knoten wurde ein identisches n ($6 \leq n \leq 12$) gewählt. Die theoretischen Werte wurden durch Betrachtung aller möglichen Pfade zwischen Sender und Empfänger berechnet⁹.

Wie man in den Graphen von Abbildung 4.4 sehen kann, folgen die experimentellen Daten sehr gut den theoretischen Werten. Lediglich bei dem Netzwerk mit zwei zusätzlichen Weiterleitern liegen die realen Werte bei hohen Verlustraten unter den theoretischen. Dies kann darauf zurückzuführen sein, dass zusätzlich zu dem synthetischen ein realer Paketverlust auftreten kann. Durch zwei zusätzlichen Weiterleiter wird der Datenverkehr annähernd verdreifacht, was die Wahrscheinlichkeit für Paketkollisionen erhöht.

4.2.2.2 Voraussetzungen für das maschinelle Lernen

In den vorangegangenen Abschnitten konnte gezeigt werden, dass die Versuchsumgebung und Reed-Solomon Implementierung funktionieren und sehr hohe Zustellraten, insbesondere bei der Verwendung mit

⁹ Eine mehrfache Ausbreitung zwischen den zwei Weiterleitern wurde nicht berücksichtigt

Weiterleiten, erzielt werden können. Um der zu Beginn des Kapitels aufgeworfenen Fragestellung

Ist ein selbstlernendes Verfahren in der Lage, basierend auf den Paketprioritäten eine “optimale” Strategie zu entwickeln?

nachzugehen, müssen noch die Voraussetzungen für die selbstlernenden maschinellen Lernverfahren geschaffen werden.

Wie bereits in Kapitel 3 stellt sich die Frage was eigentlich “optimal” in diesem Kontext bedeutet. Betrachtet man den Zusammenhang zwischen der erreichten Zustellrate auf Applikationsebene und dem gewählten Grad an Redundanz in Abbildung 4.4, so würde man intuitiv eine “So wenig Redundanz wie möglich — so viel wie nötig” Strategie wählen. Da auch die Prioritäten der Pakete berücksichtigt werden sollen, kann den wichtigeren Paketen mehr Redundanz als den niederprioren eingeräumt werden.

Ähnlich wie in Abschnitt 3.2 wird dieser Zusammenhang mathematisch mit einer Fitnessfunktion beschrieben. Den Paketen niedriger Priorität wird dabei keine Redundanz hinzugefügt. Als Variablen zur Berechnung dienen die erreichte Zustellrate auf Applikationsebene (αPDR_H für hohe, bzw. αPDR_M mittlere Priorität) und dem “Overhead” (Mehraufwand) ($O_H = (n-k/k)$). Zur Gewichtung der Prioritäten und zwischen Zustellrate und Overhead dienen die Parameter w_H (Gewicht für hohe Priorität), $w_{\alpha\text{PDR}_H}$ (Zustellrate hohe Priorität) und $w_{\alpha\text{PDR}_M}$ (Zustellrate mittlere Priorität). Als Bewertung für die eingestellten n und den erreichten Zustellraten ergibt sich die Fitness f zu:

$$f = w_H[w_{\alpha\text{PDR}_H}\alpha\text{PDR}_H + (1 - w_{\alpha\text{PDR}_H})(1 - O_H)] \\ + w_M[w_{\alpha\text{PDR}_M}\alpha\text{PDR}_M + (1 - w_{\alpha\text{PDR}_M})(1 - O_M)] \quad (4.3)$$

Für die nachfolgenden Versuche wurden die Gewichte $w_H = 0.7$, $w_{\alpha\text{PDR}_H} = 0.95$ und $w_{\alpha\text{PDR}_M} = 0.9$ gewählt.

Um die Fitnessfunktion aus Gleichung 4.3 zu maximieren und a priori die optimalen Werte für n zu bestimmen, benötigen die getesteten maschinellen Lernverfahren mehrere Attribute als Entscheidungsbasis. Wie bereits in Unterabschnitt 3.3.2 wurden mehrere Features ausgewählt:

DATENRATE Die aktuelle eingehende Datenrate in byte/s. Darin werden alle Pakete erfasst, die von der Applikation **Adhoc3** empfangen werden.

PDR MITTELWERT Der Mittelwert der Zustellrate auf Netzwerkebene zu allen direkten Nachbarn.

PDR STANDARDABWEICHUNG Standardabweichung der Zustellraten.

NACHBARN Anzahl an 1-hop Nachbarn.

Der am schwierigsten korrekt zu ermittelnde Wert ist dabei die Zustellrate zu den direkten Nachbarn. Da dieser nicht auf dem Gerät gemessen werden kann, muss er auf den Nachbarknoten bestimmt und an den Sender zurück übermittelt werden. Für die Versuchsumgebung wurde dazu auf den "Hello"-Mechanismus von **Adhoc3** zurückgegriffen. Die Geräte messen kontinuierlich die PDR für jeden Nachbarn und fügen diese Informationen in die "Hello"-Pakete ein.

Zur Bestimmung der PDR wurden verschiedene Verfahren getestet:

ANZAHL EMPFANGENER PAKETE Bei diesem Verfahren wird die Anzahl an empfangenen Paketen seit dem letzten "Hello"-Paket für jeden Nachbarn mitgezählt. Diese wird im nächsten Paket mitgeschickt. Der Sender führt ebenfalls für jeden Nachbarn eine Statistik über die versandten Pakete seit Ankunft des letzten HP¹⁰. Das Verhältnis der beiden Werte ergibt dann die PDR. Da die Auswertung dieser Werte zu unterschiedlichen Zeitpunkten stattfindet, schwankt die Zustellrate stark. Es kann sogar zu einer PDR größer als 100 % kommen.

VERHÄLTNIS DER PAKETRATEN Um das zuvor genannte Verfahren zu verbessern, wurde statt der absoluten Zahlen die Paketrate in Pakete/s berechnet. Ansonsten blieb der Ansatz gleich. Damit konnte zwar das Problem der unterschiedlichen Auswertezeitpunkte verbessert werden, jedoch bleibt ein hohes Rauschen bei niedrigen Paketraten.

SEQUENZNUMMERN Das zuverlässigste und genaueste Verfahren ist die Verwendung aufeinanderfolgender Paketnummern für jedes Paket. Damit lässt sich die PDR auf den Empfängern berechnen und über die "Hello"-Pakete zurück übermitteln. Auf der einen Seite erfordert die Sequenznummer zusätzlichen Overhead in den Kopffdaten jedes Paketes, auf der anderen werden die Nummern ebenfalls zur eindeutigen Identifikation der Pakete für die Vorwärtsfehlerkorrektur benötigt.

Da der Paketverlust stochastisch auftritt, rauscht der Wert für die PDR mehr oder weniger stark. Um dennoch ein stabiles Attribut für die Lernverfahren bilden zu können, wird der Wert für jeden Nachbarn als gleitender Mittelwert gebildet. Die zugehörige Fensterbreite wird dabei so gewählt, dass die tatsächliche Zustellrate in einem 95 % Konfidenzintervall liegt.

¹⁰ "Hallo"-Paket

4.3 MASCHINELLE LERNVERFAHREN

In den vorherigen Abschnitten wurde zum einen die verwendeten Attributwerte erläutert, zum anderen eine Fitnessfunktion vorgestellt, die aus der erreichten Zustellrate und dem dafür aufgewendeten Overhead, eine quantitative Bewertung des eingestellten Redundanzgrades (n_H, n_M) erlaubt. Diese Funktion soll nun durch ein maschinelles Lernverfahren zur Laufzeit optimiert werden. Um möglichst schnell auf wechselnde Umgebungsbedingungen reagieren zu können, soll dies durch eine Klassifikation der Attribute zu einer optimalen n_H, n_M Kombination geschehen.

Für die Optimierung wurden mehrere ML-Verfahren evaluiert. Dabei wurde besonderes Augenmerk auf die Lerngeschwindigkeit und auf die Stabilität der Lösung gelegt. Um die Verfahren miteinander vergleichen zu können wurde das selbe Testsetup wie zur Aufnahme der Grundlinien genutzt. Die Verlustrate wurde ebenfalls auf den Empfängern eingestellt, diesmal jedoch alle 65 s in 5 % Schritten von 0 bis 50 % erhöht. Um keine Artefakte der vergangenen Messungen zu erfassen, wurde die Aufzeichnung der Pakete erst 5 s nach der Umschaltung begonnen.

Nach dem Abschluss eines solchen Versuchslaufs wurde wieder bei 0 % begonnen und ein neuer Durchlauf gestartet. Die Datenspeicher der ML-Algorithmen wurden dabei nicht zurückgesetzt. Damit sollten die Verfahren durchgängig lernen und mit jedem Beginn der Versuchsläufe ihr vorhandenes Wissen nutzen können.

Ein Aspekt, der allen "Online"-Lernverfahren gemein ist, ist die Exploration. Da die Algorithmen zu Beginn der Versuche keine Kenntnisse über die Abhängigkeit der zu klassifizierenden Eigenschaft von den Attributen haben, muss die Reaktion des Systems auf die gewählten Parameter zuerst erkundet werden. Mit zunehmenden Wissen kann dies mehr und mehr für die Entscheidungen genutzt werden und weniger Exploration ist notwendig¹¹. Sofern das Lernverfahren keine Explorationsstrategie vorsieht, wurde mit einer festen Wahrscheinlichkeit von 10 % exploriert. Dabei wurde mit Hilfe des "Roulette Rad"-Verfahrens eine Kombination n_H, n_M ausgewählt.

STATISTIK Das zuerst getestete Lernverfahren ist streng genommen kein Lernverfahren, sondern eine Statistik über die Attribute, den gewählten Parametern und daraus resultierenden Werten. Dies entspricht einer Zuordnung von:

$$\vec{v} = [a_1, a_2, \dots, a_n, n_H, n_M] \Rightarrow \vec{p} = [aPDR_H, aPDR_M] \quad (4.4)$$

¹¹ Diesen Konflikt nennt man auch "exploitation vs exploration" (Ausnutzen oder Erkunden).

Während der Exploration werden bevorzugt unbekannte n_H, n_M getestet. Diese Kombination wird zusammen mit den aktuellen Attributwerten in einer Liste gespeichert. Sobald ein Nachbar eine Rückmeldung per HP über die erreichte $aPDR_H$ und $aPDR_M$ gibt, werden diese Werte als Datensatz in einer großen Speicherstruktur abgelegt. Liegt bereits ein Datum für die gleichen Werte vor ($\vec{p}_{t+1}(\vec{v})$), so wird dieses mit einem Gewicht α aktualisiert:

$$\vec{p}_{t+1}(\vec{v}) = \alpha \vec{p}(\vec{v}) + (1 - \alpha) \vec{p}_t(\vec{v}) \quad (4.5)$$

Je nach Zeitdauer zwischen dem Setzen von n_H, n_M und der Ankunft eines "Hello"-Pakets wird α verringert. Dabei liegt das Maximum bei der Periodendauer der HP. Die Speicherstruktur kann für reellwertige Attribute sehr groß werden. Daher werden die Werte diskretisiert. Jedes Attribut kann somit 20 verschiedene Werte annehmen. Die maximale Größe der Speicherstruktur ergibt sich damit zu $20^4 \cdot 7^2 \cdot 2 \cdot 8 \text{ byte} \approx 120 \text{ MiB}$.

Um die Lerngeschwindigkeit dieses Verfahrens zu analysieren, wurde das zuvor geschilderte Testverfahren für unterschiedliche Werte von α durchgeführt. Ebenfalls wurde ein weiterer Koeffizient β eingeführt, der das Updategewicht in Abhängigkeit des Rauschverhaltens der PDR für starkes Rauschen senkt. Auch für diesen Parameter wurden mehrere Versuche durchgeführt.

Betrachtet man den Lernprozess, so braucht das Verfahren bei kleinen α sehr lange um gute Ergebnisse zu erzielen. Wird der Koeffizient hingegen zu hoch gewählt, verlernt der ML-Algorithmus gute Lösungen, falls die PDR nicht korrekt bestimmt werden konnte. Selbst bei den besten Werten für α und β braucht das System lange um gute Werte zu erreichen und diese in den nachfolgenden Durchgängen nicht zu verlernen.

K-NÄCHSTE NACHBARN Eine Möglichkeit, die Lerngeschwindigkeit des statistischen Verfahrens zu erhöhen, bietet das k-nächste Nachbarn Verfahren (knn). Dabei wird versucht $\vec{p} = [aPDR_H, aPDR_M]$ eines unbekannten Attributvektors $\vec{v} = [a_1, a_2, \dots, a_n, n_H, n_M]$ anhand k bekannter $\vec{p}(\vec{v})$ zu schätzen. Dabei werden die Datenpunkte verwendet, die den geringsten Abstand zum unbekannten $\vec{v}$ haben.

Die Distanz wird über einen gewichteten "Manhattan-Abstand" folgendermaßen berechnet¹²:

$$d = \left[\sum_{i=1}^n w_i |a_i - v_i| \right] + w_{n+1} |n_H - p_H| + w_{n+2} |n_M - p_M| \quad (4.6)$$

¹² Wie im vorangegangenen Verfahren wurden die Attributwerte diskretisiert. Damit ist $0 \leq a_i, v_i < 20$

Für die in Unterunterabschnitt 4.2.2.2 vorgestellten Attribute wurde der Gewichtungsvektor

$$\vec{w} = [w_{Dr.}, w_{Avg.PDR}, w_{Std.PDR}, w_{\#N}, w_{n_H}, w_{n_M}] = [1, 2, 3, 5, 3, 3] \quad (4.7)$$

verwendet. Falls bereits Messungen zum Datenpunkt $\vec{v}$ vorliegen, werden die Informationen umliegender Datenpunkte zusätzlich berücksichtigt. Der korrekte Punkt wird um den Faktor P_w höher gewichtet.

Für kleine k und große P_w geht das “knn”-Verfahren in das statistische Lernverfahren über. Dies wird durch die Messungen bestätigt. Das Verfahren braucht lange um gute Kombinationen zu lernen, behält diese dann aber stabil bei. Wird k zu groß gewählt, bzw. P_w zu klein, rauscht das Ergebnis sehr stark und konvergiert nicht zu einem stabilen Verhalten.

Ebenso wie das zuvor beschriebene Verfahren benötigt “K-nächste Nachbarn” eine große Datenbank für die Speicherung der Werte. Das Berechnen der Distanz und damit das Auffinden benachbarter Werte benötigt viel Rechenzeit. Selbst bei der besten Parameterkombination ist die Lerngeschwindigkeit moderat und das Resultat nicht besonders stabil.

Q-LERNEN Eine Familie von ML-Verfahren, die häufig für Problemstellungen im Umfeld von MANet eingesetzt wurde, ist das “bestärkende Lernen” (vgl. [ANY13]). Dessen prominentester Vertreter ist das s. g. “Q-Lernen”. Bei diesem Verfahren wird in jedem Zustand S eine Aktion A gewählt, so dass eine “Belohnung” R auf längere Sicht maximiert wird.

Angewendet auf das Problem der optimalen n_H, n_M ergibt sich damit folgende Zuordnung:

- s Die Zustände ergeben sich aus den aktuellen Attributwerten. Um den Zustandsraum nicht beliebig groß werden zu lassen, wurden die Attribute wie bereits bei den vorangegangenen Ansätzen diskretisiert.
- A Als Aktion kann in jedem Zustand eine Kombination von n_H, n_M gewählt werden.
- R Die Belohnung besteht aus dem Ergebnis der in Gleichung 4.3 definierten Fitnessfunktion.

Das Lernverfahren definiert zwei Parameter: Mit α – vergleichbar mit dem zuvor definierten Updategewicht – kann die Lerngeschwindigkeit eingestellt werden. γ bestimmt die “Weitsicht” des Algorithmus. Mit niedrigen Werten wird der kurzfristige Nutzen höher bewertet, bei hohen berücksichtigt der Algorithmus möglichen zukünftigen Nutzen.

In den Versuchsläufen zeigte sich der Algorithmus sehr anfällig gegenüber Rauschen. Da dies sowohl in der Bestimmung des aktuellen Zustands, als auch bei der Berechnung der "Belohnung" auftritt, ist die Berechnung der "Q"-Werte zu unzuverlässig um stabile Resultate zu erhalten. Betrachtet man das Verhalten im Detail, so oszilliert der Algorithmus häufig zwischen zwei Zuständen und setzt in den Aktionen unterschiedliche Kombinationen von n_H, n_M . Dadurch wird auch bei geringem α kein stabiles Ergebnis erreicht.

Ein weiterer Nachteil des Verfahrens ist der sehr große Zustands- und Aktionsraum, der im Speicher hinterlegt sein muss. Wird jeder "Q"-Wert mit 8 byte gespeichert, ergibt sich eine Datenbankgröße von ≈ 60 MiB.

CACLA Ein weiterer Vertreter aus dem Bereich des "bestärkenden Lernens" ist der "Continuous Actor Critic Learning Automaton" (Kontinuierliche-Aktions kritischer Lernautomat (CACLA)) Algorithmus. Dabei werden die Aktionen nicht als diskret, sondern als kontinuierlich angenommen. Es konnte gezeigt werden, dass mit dieser Herangehensweise auch diskrete Probleme effektiver als z. B. mit "Q-Lernen" gelöst werden können (vgl. [VaWio9]).

Im Unterschied zu "Q-Lernen" werden die Aktionen nicht mehr als Linearkombinationen von n_H, n_M in den Zuständen gespeichert, sondern als zwei reell-wertige Zahlen (r_H, r_M). Erzielt während der Exploration eine n_H, n_M Kombination ein "optimaleres" Ergebnis, werden die r angepasst:

$$r_{t+1} = (1 - \alpha)r_t + \alpha n \quad (4.8)$$

Da keine Informationen über die Ergebnisse der einzelnen n_H, n_M vorliegen, kann das "Roulette Rad"-Verfahren nicht zur Exploration verwendet werden. Daher wurden zwei normalverteilte Zufallszahlen mit einer Standardabweichung von 1,5 um die zum aktuellen r gerundet und als n verwendet.

Ebenso wie für "Q-Lernen" wurden verschiedene Kombinationen von α und γ getestet. Für keinen der Versuchsläufe konnte der Algorithmus gute Ergebnisse erzielen. Eine Konvergenz zu bestimmten n für verschiedene Verlustraten ist nicht erkennbar. Das Verfahren scheint mehr als die bisherigen vom Rauschen der Attribute und der aPDR betroffen zu sein. Dadurch werden bei stochastisch besseren Verlustraten die Aktionen zu suboptimalen Werten hin verschoben.

4.3.1 Naiver Bayes-Klassifizierer

In Unterabschnitt 3.3.3 wurde das Naive Bayes Verfahren bereits eingesetzt um zu entscheiden, ob ein Paket weitergeleitet werden soll oder

nicht. Es kann allerdings auch eingesetzt werden, um komplexere Entscheidungen zu treffen, wie Z. B. die Wahl zwischen verschiedenen Strategien für zuverlässigen Unicast-Datenverkehr ([LQZ13]).

Um eine optimale Kombination von n_H, n_M für die Vorwärtsfehlerkorrektur zu wählen, wird in einem ersten Schritt NB¹³ zur Schätzung der zu erwartenden aPDR bei entsprechenden n verwendet. Damit kann für jede Kombination von n_H, n_M Gleichung 4.3 berechnet und die optimalen Parameter gewählt werden.

Die Schätzung der Verlustrate auf Applikationsebene erfolgt nach dem Prinzip der maximalen Wahrscheinlichkeit. Dazu wird die Wahrscheinlichkeit (P) für eine Priorität (p) und n für die aktuellen Attribute ($\vec{a}$) für mehrere mögliche aPDR ($aPDR_i$) bestimmt:

$$P(aPDR_i | a_1, \dots, a_n, n, p) = \frac{P(aPDR_i | n, p) P(a_1, \dots, a_n | aPDR_i, n, p)}{P(a_1, \dots, a_n)} \quad (4.9)$$

Unter der Annahme, dass die Attribute unabhängig voneinander sind (siehe Unterabschnitt 3.3.3), lässt sich der Satz von Bayes anwenden.

Da $P(a_1, \dots, a_n)$ für alle aPDR gleich ist, lässt sich das Ergebnis normalisieren und die geschätzte Verlustrate ergibt sich zu:

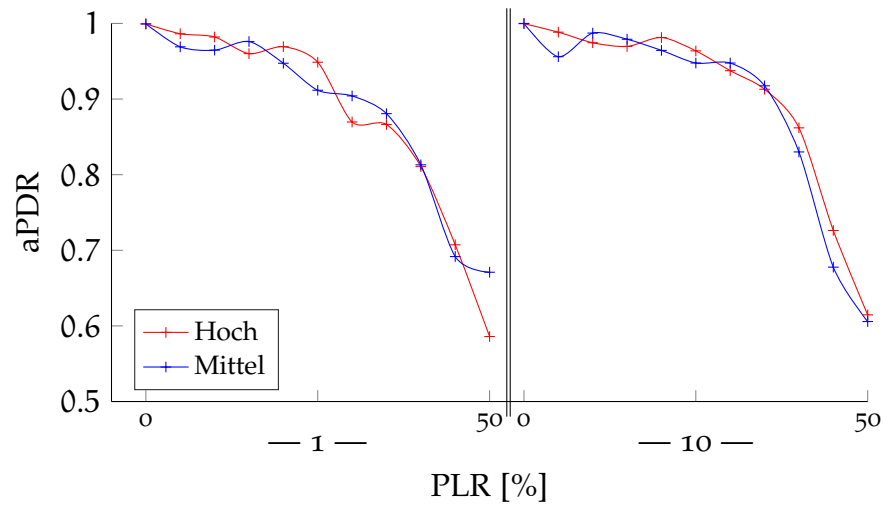
$$aPDR(n, p) \approx \underset{aPDR_i}{\operatorname{argmax}} \quad P(aPDR_i | n, p) \prod_k P(a_k | aPDR_i, n, p) \quad (4.10)$$

Um die bedingten Wahrscheinlichkeiten in der Software abzuspeichern wurde angenommen, dass die Zustellrate für eine Priorität nicht von n der anderen abhängt. Somit wurden für “Hoch” und “Mittel” zwei Speicherstrukturen angelegt, die wiederum 7 Speicher für die unterschiedlichen n enthalten. Darin ist zum einen die Verteilung der Verlustraten ($P(aPDR_i)$), zum anderen für jede mögliche $aPDR_i$ die Wahrscheinlichkeit für ein Attribut ($P(a_k)$) als Normalverteilung gespeichert.

Die Speicherung als Verteilungsfunktion birgt zwei wesentliche Vorteile: zum einen ist weniger Speicherplatz erforderlich als z. B. ein Histogramm benötigen würde, zum anderen können damit bis dahin nicht gemessene Werte geschätzt werden. Nachteilig ist hingegen, dass die Daten als normal-verteilt angenommen werden, was in der Realität nicht immer zutrifft.

Zur Aktualisierung des Mittelwerts und der Standardabweichung bei Ankunft von “Hello”-Paketen wurde das Verfahren nach [West79] verwendet. Wie beim statistischen Lernverfahren kamen zwei Parameter zum Einsatz, über die sich die Lernrate (α) und die Berücksichtigung des Rauschverhaltens (β) der Verlustraten einstellen lässt.

13 Naive-Bayes



(a) Zustellrate

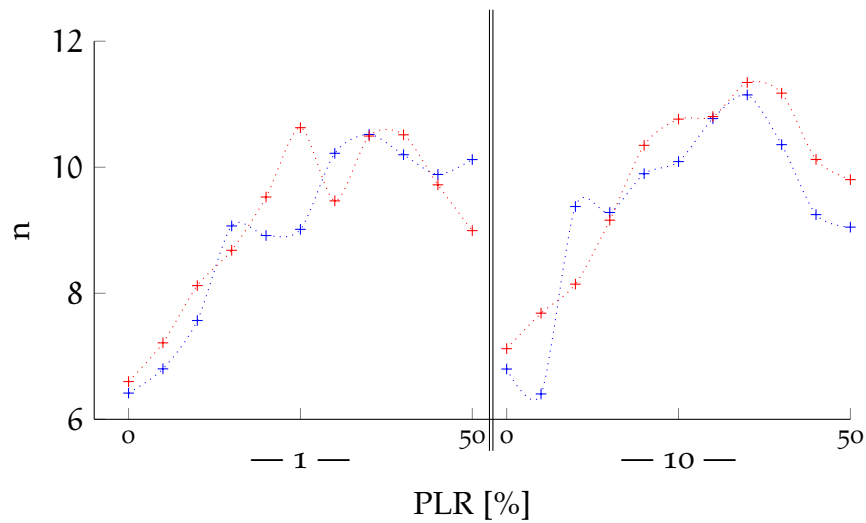
(b) gewählter Redundanzgrad n

Abbildung 4.5: Konvergenzverhalten des Naive Bayes Lernverfahrens über 10 Versuchsläufe. Innerhalb eines Versuchslaufs wurde die Verlustrate schrittweise von 0 auf 50 % erhöht. Dargestellt sind die Ergebnisse des ersten und letzten Durchgangs.

Abbildung 4.5 zeigt das Ergebnis eines Versuchslaufs, bei dem α und β einen guten Kompromiss zwischen Lerngeschwindigkeit und Stabilität bilden. Dargestellt ist der erste und letzte Durchgang. Dazwischen liegen 8 weitere Versuche, die zur besseren Übersicht ausgeblendet wurden. In Abbildung 4.5a ist die aPDR über die jeweils eingestellte PLR aufgetragen. Abbildung 4.5b zeigt die vom Algorithmus gewählten n .

Wie man sieht erreicht das naive Bayes-Verfahren bereits im ersten Durchgang gute Ergebnisse. Für die hohe Priorität werden bis 30 % hohe Zustellraten erreicht. Ebenso wird dieser meistens eine höhere Redundanz eingeräumt, was den Vorgaben der Bewertungsfunktion entspricht. Betrachtet man das Ergebnis für den zehnten Versuchslauf, so wird die hohe Zustellrate noch länger beibehalten. Erst ab 35 % scheint ein weiterer Einsatz von Redundanz zu keiner weiteren Verbesserung zu führen. Betrachtet man die Bewertungsfunktion in Gleichung 4.3, so sollte n_H immer über n_M liegen. Dies ist bei 10 % Verlustrate nicht der Fall. Der Algorithmus wählte ein deutlich größeres n für die mittlere Priorität. Daran sieht man gut die stochastische Natur des Versuches und des Lernprozesses. Wurde eine inkorrekte PLR gemessen, wird von falschen a priori Daten ausgegangen und eine falsche aPDR geschätzt. Ebenso wird das Resultat für den falschen Datensatz aktualisiert, was wiederum die nächste Schätzung verschlechtert. Somit dauert es eine längere Zeit, bis falsch gelernte Daten wieder verlernt werden.

Trotz der Probleme, die beim Lernvorgang auftreten können, bietet das naive Bayes-basierte Verfahren eine gute Lerngeschwindigkeit und Stabilität. Durch die Näherung der Ergebnisse als Normalverteilungen können zum einen bisher unbekannte Werte von Attributen geschätzt werden. Zum anderen ist der benötigte Speicher der Datenbasis deutlich geringer als bei den anderen Ansätzen. So benötigt der Algorithmus bei den vorgenommenen Diskretisierungen der Werte ca. 18 KiB Speicher.

4.3.2 Evaluation

Zur Evaluation der intelligenten Vorwärtsfehlerkorrektur mit dem naiven Bayes-Klassifizierer (genannt **iFec**) wurde das gleiche Testszenario wie zur Aufnahme der Grundlinien in Unterunterabschnitt 4.2.2.1 verwendet. Zuerst wurde das Verhalten in einem einfachen Netzwerk, bestehend aus einem Sender und einem Empfänger, untersucht.

Abbildung 4.6 zeigt die Durchschnittswerte für fünf Wiederholungen. Dabei wurde dem Lernverfahren 60 s Zeit für die Exploration gegeben. Anschließend wurde das gelernte Wissen angewandt und der Datenverkehr für 60 s aufgezeichnet. Dieses Vorgehen wurde für 0,5, 10, 25 und 50 % simulierte Verlustrate auf dem Kanal wiederholt. Die Datenbanken wurden für jeden Versuch zurückgesetzt.

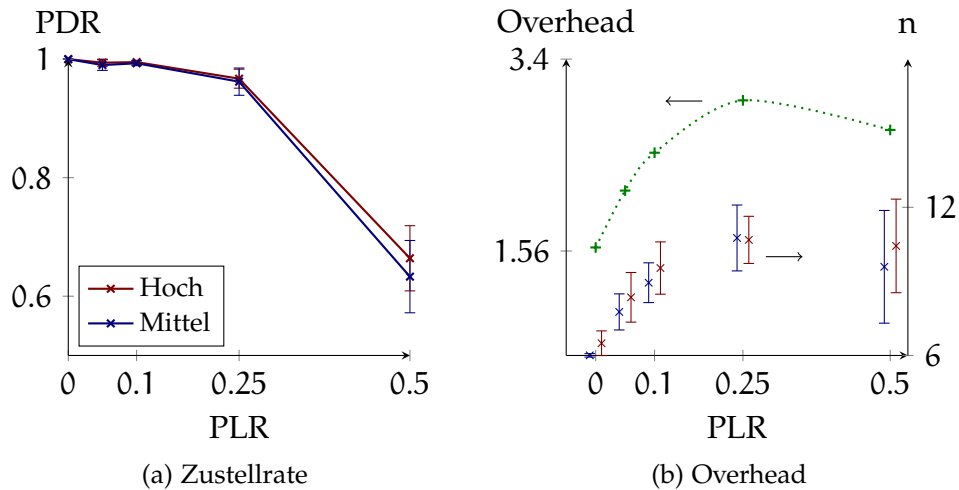
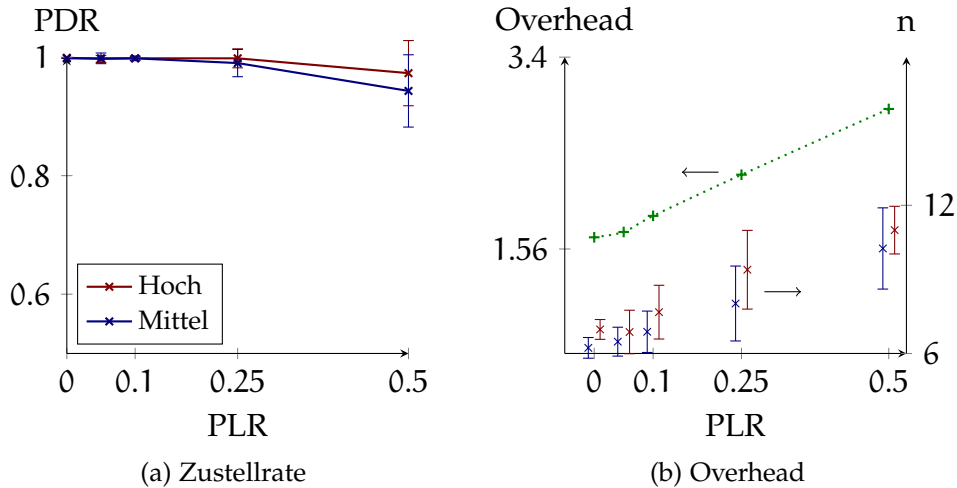


Abbildung 4.6: Evaluation von **iFec** in einem Netzwerk mit einem Sender und Empfänger. Die Resultate sind Durchschnittswerte von 5 Versuchsläufen mit zugehöriger Standardabweichung.

In Abbildung 4.6a ist die erzielte Zustellrate auf Applikationsebene für die hohen- und mittelpriorisierten Pakete über die Verlustrate aufgetragen. Wie man sieht wird bis zu 25 % PLR eine sehr hohe Zustellrate erreicht. Im Detail betrachtet liegt die aPDR der Pakete hoher Priorität über der der mittleren.

Abbildung 4.6b zeigt zum einen im unteren Bereich die durchschnittlichen n für die Prioritäten. Zur besseren Darstellung sind die Datenpunkte leicht auf der Abszisse von der korrekten Position weggerückt. Die Werte beziehen sich jedoch auf nächstgelegene PLR. Zum anderen ist in grün der Overhead aufgetragen. Dieser Wert gibt den Faktor an, der aufgewendet wurde um die Nutzdaten von Sende- zu Empfangsapplikation zu übertragen. Darin sind sowohl die Kopfdaten aller Netzwerkschichten, als auch die Redundanzpakete enthalten. Man erkennt, dass **iFec** für niedrige Verlustraten kaum Datenmehraufwand gegenüber der Übertragung ohne Vorwärtsfehlerkorrektur (≈ 1.56) verursacht. Betrachtet man die durchschnittlichen n , so zeigt sich, dass das Verfahren für die höhere Priorität größere n wählt. Bei 25 % wird für beide ein n von ca. 10 gewählt. Dieses Ergebnis lässt sich wie zuvor auf stochastische Effekte zurückführen.

Für einfache Netzwerke mit einem Sender und Empfänger liefert **iFec** gute Ergebnisse. Um zu testen, wie sich der Algorithmus für komplexere Netzwerke verhält, wurde das Netzwerk um zwei Weiterleiter erweitert. Die Versuchsanordnung entspricht damit der zweiten in Unterunterabschnitt 4.2.2.1.



Abbildungung 4.7: Evaluation von **iFec** in einem Netzwerk mit einem Sender, Empfänger und zwei Weiterleitern. Die Resultate sind Durchschnittswerte von 5 Versuchsläufen mit zugehöriger Standardabweichung.

In Abbildung 4.7 sind die Resultate der Versuchsläufe dargestellt. Das Vorgehen entsprach dabei dem für zwei Teilnehmer. Die Verlustraten wurden auf allen Knoten identisch eingestellt. Wie man in Abbildung 4.7a erkennen kann, wird für alle Verlustraten eine hohe aPDR erreicht. Insbesondere liegt die Zustellrate für die hohe Priorität über der mittleren. Dies spiegelt sich auch in den durchschnittlichen n in Abbildung 4.7b wieder. **iFec** wählt durchgehend größere n für die hohe Priorität. Vergleicht man den Overhead mit den Werten für das einfachere Netzwerk (vgl. Abbildung 4.6b), so müssen weniger Daten bei gleicher Verlustrate verschickt werden. Der Algorithmus lernt die gestiegene aPDR durch die Weiterleiter und wählt niedrigere n .

4.4 EVALUATION DES GESAMTSYSTEMS

In den vorangegangenen Abschnitten wurde ein Protokoll vorgestellt, das, abhängig von der Priorität der Daten durch intelligente Vorwärtsfehlerkorrektur, die Zustellrate erhöhen kann.

Kapitel 3 befasste sich mit der energieeffizienten, zuverlässigen Weiterleitung priorisierter Daten. Durch die Kombination der beiden Protokolle sollte sich die PDR weiter steigern lassen. Insbesondere sollte die Bewertungsfunktion von **iFec** dafür sorgen, dass der Energiemehraufwand eher für die zuverlässige Übertragung höher priorisierter Daten verwendet wird.

Zur Evaluation der Protokollkombination wurde **iFec** in OMNeT++ implementiert. Dadurch kann eine Vielzahl von Szenarien und Protokollkombinationen getestet werden. Zum Vergleich mit dem Stand der Technik werden die Protokolle **AAG** und **SBA** in der in Unterabschnitt 3.4.2 vorgestellten prioritätsbasierten Variante verwendet. Zur Vermeidung von Paketverlusten wurde ebenfalls das in [SBDK10] und Abschnitt 4.1 vorgestellte MLA¹⁴-Schema in OMNeT++ implementiert.

Da in MLA Prioritäten nicht berücksichtigt werden, wurden je nach Priorität unterschiedliche Parametervektoren $\vec{v} = [C; d; k]$ gewählt. Für die hohe Priorität $\vec{v}_H = [2; 0.01; 3]$ mittlere $\vec{v}_M = [1; 0.1; 1]$ und niedrige $\vec{v}_L = [1; 0; 1; 0]$. Die weiteren Konfigurationseinstellungen des Simulators entsprechen denen aus Unterabschnitt 3.4.2.

Wie bereits bei den vorangegangenen Versuchsläufen wurde eine Vielzahl an Szenarien unterschiedlicher Anzahl statischer und mobiler Knoten bei verschiedenen Paketraten untersucht. Um die Signifikanz der Daten zu erhöhen wurde jedes Szenario fünfmal für unterschiedliche Startwerte der Zufallsgeneratoren für die Positionierung und des Kanalzugriffs evaluiert. Die nachfolgend aufgeführten Werte sind die Mittelwerte aus diesen Versuchsläufen.

Die Messungen wurden nach der Explorationsphase durchgeführt. Dazu wurde den Algorithmen 500 s Simulationszeit gegeben, um ihre Entscheidungsbasis durch Exploration aufzubauen. Danach wurde das gelernte Wissen nur noch angewendet, die Resultate der aPDR jedoch weiterhin in die Datenbanken eingetragen.

In Tabelle 4.1 sind die Resultate für ein Netzwerk, bestehend aus 5 stationären und 15 mobilen Knoten, eingetragen. Die Paketrate wurde auf 5 pps eingestellt. In der ersten Spalte sind in kursiv die Weiterleitungsprotokolle eingetragen. Darunter stehen jeweils die Protokolle zur Vermeidung von Paketverlust. Wie bereits in Unterabschnitt 3.4.2 gezeigt werden konnte, erzielt das Naive Bayes-Weiterleitungsprotokoll sehr gute Zustellraten bei geringem Energieverbrauch.

Vergleicht man die Ergebnisse so sieht man, dass es keine "beste" Lösung gibt bei der die Zustellrate maximal und gleichzeitig der Energieverbrauch und die Ende-zu-Ende Verzögerung minimal ist. Jedoch erreicht das **iFec**-Protokoll insbesondere in Kombination mit dem Naive Bayes basierten Weiterleitungsprotokoll deutlich bessere Zustellraten bei geringem Energiemehraufwand.

Dieses Verhalten tritt bei höheren Paketraten noch stärker auf. In Tabelle 4.2 sind die Ergebnisse für 20 Paket pro Sekunde eingetragen.

¹⁴ Message Loss Avoidance (Vermeidung von Paketverlust)

Tabelle 4.1: Resultate für ein Netzwerk mit 15 mobilen und 5 statischen Knoten für 5 pps.

	αPDR_H	αPDR_M	P_{norm}	Delay [ms]
<i>Flooding</i>				
–	69,5	68,7	1	3,18
iFec	76,5	74,6	1,06	10,14
MLA	80,4	73,9	1,14	7,42
<i>AAG Prio.</i>				
–	58,0	54,1	0,28	1,82
iFec	64,0	58,3	0,31	38,39
MLA	69,9	57,3	0,41	8,82
<i>Naive Bayes</i>				
–	69,6	65,3	0,47	3,43
iFec	75,2	69,4	0,51	21,65
MLA	80,8	73,7	1,02	8,34
<i>SBA Prio.</i>				
–	65,9	65,1	0,49	3,55
iFec	72,1	69,5	0,52	32,91
MLA	78,2	70,3	0,63	8,90

Hier sinkt für die Kombination aus **iFec** und Naive Bayes sogar der Energieverbrauch¹⁵.

Ein Wert, der bei dem **iFec** Protokoll negativ auffällt, ist die Ende-zu-Ende Verzögerung. Diese liegt insbesondere bei 5 pps höher als bei den anderen Übertragungsarten. Bei dem **MLA**-Verfahren wird die Redundanz für jedes Paket gebildet. Bei **iFec** hingegen werden Pakete zur Wiederherstellung verlorengegangener Daten über eine Gruppe von k Paketen gebildet. Da dies prioritätsabhängig geschieht, muss der Algorithmus warten bis k Pakete gleicher Priorität vorliegen. Je geringer die Emissionsrate ist, desto länger dauert folglich die Redundanzbildung und somit auch die Wiederherstellung eines Pakets. In Folge steigt die durchschnittliche Ende-zu-Ende Verzögerung. In Tabelle 4.2 erkennt man, dass diese bei höheren Paketrate sinkt und typischerweise unter der von MLA liegt.

¹⁵ In der Tabelle nicht dargestellt sind die Werte für die Zustellrate Pakete niedriger Priorität. Durch die Verwendung von **iFec** mit dem NB-Protokoll verringert sich diese von 42,8 % auf 35,1 %, womit sich die Energiereduktion erklärt.

Tabelle 4.2: Resultate für ein Netzwerk mit 15 mobilen und 5 statischen Knoten mit 20 pps.

	αPDR_H	αPDR_M	P_{norm}	Delay [ms]
<i>Flooding</i>				
–	56,6	54,8	1	7,34
iFec	70,9	63,1	1,06	20,24
MLA	77,7	56,0	1,16	32,17
<i>AAG Prio.</i>				
–	54,9	49,4	0,35	2,21
iFec	66,3	56,4	0,39	16,08
MLA	68,8	49,4	0,49	15,86
<i>Naive Bayes</i>				
–	65,4	57,5	0,47	4,07
iFec	76,3	63,1	0,46	15,39
MLA	78,1	56,0	1,09	28,11
<i>SBA Prio.</i>				
–	59,0	57,5	0,63	5,32
iFec	71,6	68,4	0,69	18,52
MLA	79,7	60,5	0,80	21,14

4.5 ZUSAMMENFASSUNG

In diesem Kapitel wurde ein adaptives Verfahren entwickelt, das in der Lage ist, die Zustellrate prioritätsabhängig zu steigern. Dazu wurden zuerst die existierenden Lösungen des aktuellen Stands der Technik vorgestellt. Anschließend ein Verfahren, bei dem abhängig von der aktuellen Zustellrate auf Applikationsebene mehr oder weniger Redundanz für einen Datenstrom erzeugt wird.

Bei der Evaluation des Verfahrens offenbarten sich mehrere Schwächen. So ist es zum einen durch das Feedbacksystem über die "Hello"-Pakete und die Modellierung als endlicher Automat relativ träge. Zum anderen ermöglicht das gewählte Vorwärtsfehlerkorrekturverfahren keine feingranulare Abstufung des Redundanzgrades.

In der Folge wurde ein selbst-lernendes System entwickelt, das proaktiv die Parameter des Reed-Solomon Fehlerkorrekturverfahrens setzen kann. Über eine Gewichtungsfunktion kann prioritätsabhängig eingestellt werden, wie die Kosten-Nutzen-Abwägung zwischen dem Datenaufwand durch die Redundanz und der geschätzten Zustellrate zu bewerten ist.

Für das selbst-lernende System wurden verschiedene maschinelle Lernverfahren getestet. Dabei wurden Parametervariationen evaluiert, um eine möglichst hohe Lerngeschwindigkeit bei stabilen Endresultaten zu erhalten. Die Fähigkeit des Naive Bayes-Klassifizierers, unbekannte Parameterkombinationen aus den zugrundeliegenden Normalverteilungen zu schätzen, sowie die damit einhergehende Glättung der Werte, ließen die Wahl auf diesen Algorithmus fallen.

Die Kombination aus Gewichtungsfunktion, Attributen und Lernverfahren wurde zu einem prioritätsabhängigen Fehlerkorrekturprotokoll, genannt **iFec**, zusammengefasst. Dieses wurde zuerst auf Android-basierten Smartphones in einfachen Netzwerken evaluiert. Insbesondere in einem Netzwerk mit zwei Weiterleitern konnte **iFec** den Redundanzgrad sehr gut steuern und sorgte für hohe Zustellraten. Ebenso konnte die Abhängigkeit von den vorgegebenen Prioritäten gezeigt werden. Das Protokoll gewährte den hochprioren Daten mehr Redundanz und erreichte somit eine höhere aPDR.

In einem letzten Schritt wurde die Leistungsfähigkeit der Kombination von **iFec** mit dem in Kapitel 3 entwickelten Naive Bayes-basierten Weiterleitungsprotokoll evaluiert. Dazu wurde das Fehlerkorrekturprotokoll in OMNeT++ implementiert und in komplexen Netzwerken mit verschiedenen Paketraten getestet. Es konnte gezeigt werden, dass die Kombination aus dem NB-Protokoll mit **iFec** sehr gute Zustellraten bei geringem Energiebedarf erzielt. Insbesondere benötigt das Fehlerkorrekturprotokoll nur wenig zusätzliche Energie um die aPDR deutlich zu steigern.

PROTOTYPISCHE IMPLEMENTIERUNG

Ein gerne herangezogenes Szenario wenn es um den potentiellen Einsatz von MANet geht, ist das Katastrophenszenario. Nach einem Störfall, Naturkatastrophe oder ähnlichem ist die vorhandene Kommunikations-Infrastruktur zerstört worden. Einsatz- und Rettungskräfte sind vor Ort und sollen verletzte Personen suchen und retten, die Schäden sichten, etc. Dabei ist ein hohes Maß an Koordination und Kommunikation der Einsatzkräfte erforderlich. Hier bietet sich ein MANet an, das von den Geräten der Einsatzkräfte spontan gebildet werden kann und keine existierende Infrastruktur benötigt.

In diesem Umfeld wurde ein Demonstrator realisiert, der die Leistungsfähigkeit des in den vorangegangenen Kapiteln vorgestellten Protokolls zeigen soll. Die dabei implementierte Applikation stellt ein dezentrales Echtzeit-Audiokommunikationssystem zur Verfügung.

Die grundsätzliche Funktionalität wurde in [PrKl13] und [PBS14] vorgestellt. Echtzeit-Audioübertragungen stellen hohe Anforderungen an das darunterliegende Netzwerk und die Übertragungseigenschaften. Gemäß den Empfehlungen der ITU¹ (vgl. [ITU 03]) sollte die "Mund zu Ohr"-Verzögerung bei Sprachkommunikation idealerweise unter 150 ms, nicht jedoch über 400 ms liegen. Ein weiteres Qualitätsmerkmal ist die Paketverlustrate, die bei Sprachdaten nicht unter 5 % fallen sollte (vgl. [MaAn09, S. 501]).

Neben der Verwendung von Vorwärtsfehlerkorrektur gibt es weitere Verfahren um die Paketverlustrate zu senken und die Zuverlässigkeit auf Applikationsebene zu erhöhen. Insbesondere bietet sich bei der Verwendung verlustbehafteter Kompression ein MDC² genanntes Verfahren an. Bei diesem wird ein Multimedia Datenstrom mit unterschiedlichen Kompressionsraten in mehrere Datenströme kodiert, über ein Netzwerk übertragen und anschließend dekodiert. Gehen Pakete aus einem der Datenströme verloren, können die Inhalte der ursprünglichen Quelle dennoch wiederhergestellt werden.

In [GSG12] verwendeten die Autoren einen Video Datenstrom. Dieser wurde verlustbehaftet komprimiert und über ein MANet verschickt. Gleichzeitig wurde ein zweiter Strom erzeugt, der aufbauend auf den Daten des Ersteren, eine höherqualitative Repräsentation der Videodaten enthält. Traten in dem Netzwerk keine Paketverluste auf, so konnten

¹ International Telecommunication Union (Internationale Fernmeldeunion)

² Multiple Description Coding (Kodierung mit mehreren "Beschreibungen")

aus der Kombination der Daten hochqualitative Videodaten dekodiert werden. Gingen Pakete aus dem zweiten Datenstrom verloren, konnten die Videodaten immer noch kontinuierlich, wenngleich in schlechterer Qualität wiedergegeben werden.

Zur Demonstration von iFec kommt ein ähnliches Verfahren zum Einsatz. Statt eines Videodatenstroms werden Audiodaten in Echtzeit in zwei Datenströme unterschiedlicher Datenrate, und damit auch unterschiedlicher Qualität, kodiert. Diese dienen auch als Vorgabe für die Priorität. Um eine kontinuierliche Ausgabe an den Empfangsgeräten zu erreichen wurden die Daten niedrigerer Qualität höher priorisiert als Daten hoher Qualität.

5.1 TESTBEDIMPLEMENTIERUNG IN ANDROID

Als Basis des Demonstrators dient das Framework Ad-hoc3, das bereits in Kapitel 4 verwendet wurde um Versuche durchzuführen. Eine detaillierte Beschreibung von Ad-hoc3 kann in Abschnitt 2.3 nachgelesen werden.

Durch den modularen Aufbau von Ad-hoc3 kann die Software einfach um die benötigte Funktionalität erweitert werden. Dazu musste eine Applikation (*VoiceApp*) innerhalb des Frameworks entworfen werden, die aus einem Teil zur Benutzerinteraktion (*VoiceHMI*), sowie einem Teil zur Generierung und Bearbeitung von Paketdaten (*VoiceService*), besteht. Für diese Paketdaten wurde eine eigene Paketformate (*VoicePacket* und *VoicePacketLQ*) definiert. Zur Realisierung der selektiven Weiterleitung wurde ein Weiterleitungsprotokoll (*BayesForwarder*) vorgesehen. Analog zum Systemmodell des Frameworks aus Abbildung 2.4 auf Seite 28 ergibt sich für die Beispielapplikation zur Sprachübertragung ein Modell welches in Abbildung 5.1 dargestellt ist.

Darauf sind die Elemente *VoiceApp*, *FecSender* und *-Receiver* sowie der *NaiveBayesForwarder* (*Bayes*) zu sehen. Die Blöcke zur Bereitstellung und Steuerung der Vorwärtsfehlerkorrektur wurden bereits in Kapitel Kapitel 4 ausführlich beschrieben. Daher werden im Folgenden die Einheiten zur Behandlung der Audiodaten und zur selektiven Weiterleitung genauer beschrieben.

5.1.1 *VoiceApp*

Die Funktionalität zur Behandlung der Audiodaten von der Aufnahme bis zur Paketisierung bzw. der Dekodierung und der anschließenden Wiedergabe wurde als Applikation (*VoiceApp*) in Ad-hoc3 implementiert. In Abschnitt 2.3 wurde bereits erläutert aus welchen Komponenten diese bestehen. Im Fall der *VoiceApp* sind das folgende Teile:

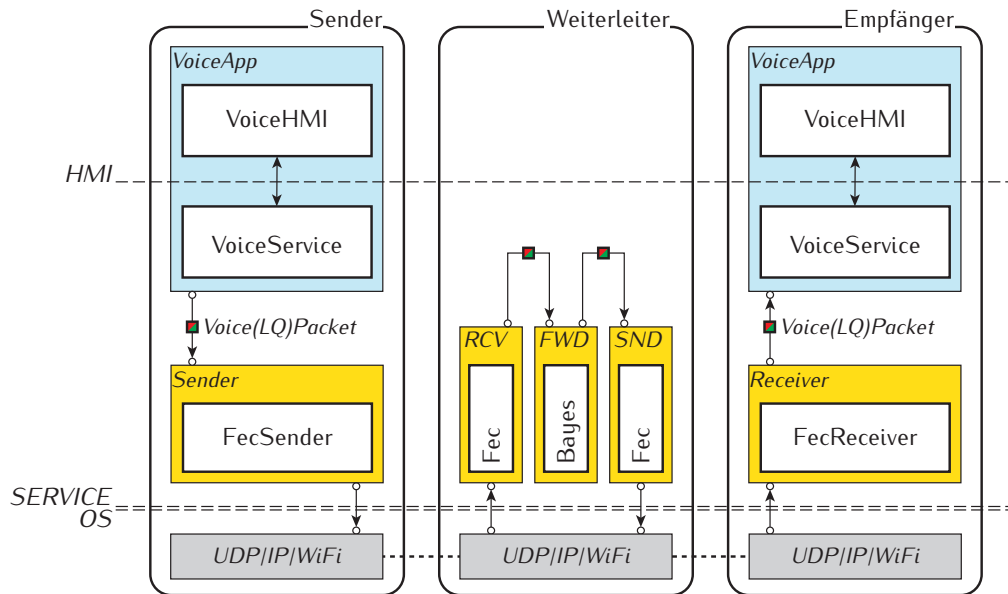


Abbildung 5.1: Systemmodell der prototypischen Implementierung der Beispiellapplikation zur Sprachübertragung.

HMIAPPLICATION Die Klasse **VoiceHMI** bildet die Schnittstelle für Benutzereingaben. Es werden zwei Schaltflächen zur Steuerung der Aufnahme sowie der Wiedergabe dargestellt. Ein weiterer Block zeigt eine Statistik zu den versandten Paketen oder Informationen zur aktuellen Wiedergabe - abhängig von Aufnahme oder Wiedergabezustand. Das Erfassen sowie die Wiedergabe von Audiodaten sollte per Definition auch zum HMI gehören. Es wurde jedoch festgestellt, dass der Transport von Audiodaten und die zugehörige Kodierung und Dekodierung über die IPC³ zu viele Ressourcen benötigt, als dass das Programm noch in Echtzeit lief. Die entsprechende Funktionalität wurde daher im Hintergrunddienst implementiert. Ein Bildschirmfoto des Benutzerinterface ist in Abbildung A.1 zu sehen.

SERVICEAPPLICATION In der Klasse **VoiceService** sind alle Funktionen zusammengefasst, die Audiodaten behandeln. Dazu gehören neben Aufnahme und Wiedergabe auch die verlustbehaftete Kompression sowie die Paketisierung.

INSTRUCTIONS Zur Kommunikation zwischen dem Benutzerinterface und dem Hintergrunddienst wurden zwei Anweisungen definiert: **VoiceInstruction** Mit Hilfe dieser Anweisung kann von der Benutzeroberfläche aus die Aufnahme und Wiedergabe gestartet werden.

³ Inter-Process Communication (Inter-Prozess-Kommunikation)

VoiceInstructionStatistics Informationen zu Paketraten, dem Anteil hoher Qualitätsdaten, PLR, etc. können damit an das HMI übertragen werden.

PACKETS Zur Datenkommunikation über das MANet wurden zwei Pakettyten definiert:

VoicePacket Die Audiopakete dienen dem Transport der Audiodaten zu den anderen Geräten im Netzwerk. Neben den Nutzdaten sind Felder für Sequenznummer, Abtastrate und verwendetem Kompressionscodec⁴ vorgesehen. Das vollständige Paketlayout kann in Abbildung A.2 eingesehen werden.

LQVoicePacket Neben den hochqualitativen Audiodaten sollen auch höher komprimierte Daten übertragen werden. Zur effizienteren Übertragung wurde das *VoicePacket* Format abgewandelt und die Felder für die Abtastrate und dem verwendeten Codec entfernt. Bei eingehenden Paketen wird davon ausgegangen, dass diese identisch zum hochqualitativen Datenstrom ist.

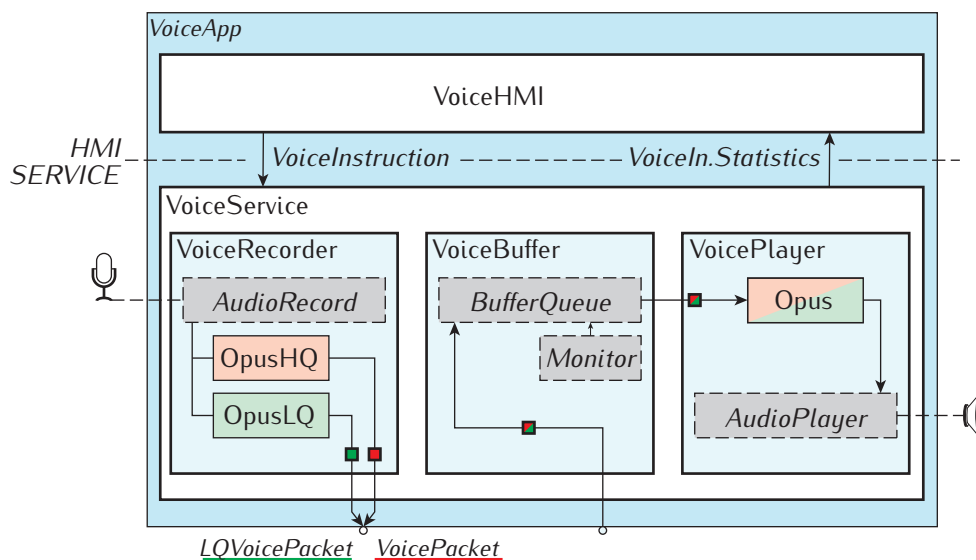


Abbildung 5.2: Detailansicht der Voice Applikation.

Das Zusammenspiel der einzelnen Komponenten ist in Abbildung 5.2 dargestellt. Die Funktionalität zur Behandlung der Audiodaten ist in der Klasse *VoiceService* implementiert. Dort übernimmt die Komponente *VoiceRecorder* die Funktionen auf der Senderseite. Auf Empfangsseite sind die Klassen *VoiceBuffer* und *VoicePlayer* für das korrekte Abspielen verantwortlich.

⁴ Als Codec bezeichnet man die Kombination aus Kodierer und entsprechendem Dekodierer.

SENDERSEITE Um kontinuierliche Audiodaten vom verschiedenen Quellen unter Android zu erhalten, kann eine Instanz des *AudioRecord* verwendet werden. Korrekt initialisiert liefert diese in regelmäßigen Abständen PCM⁵ kodierte Audiodaten des internen Mikrofons. Um diese Datenblöcke zu komprimieren wurde der Audiocodec Opus⁶ als nativer Programmteil implementiert⁷. Als Basis diente die Referenzimplementierung von Jean-Marc Valin und Koen Vos⁸.

In der Klasse *OpusHQ* werden die PCM Daten mit Hilfe des Opus Codecs komprimiert. Dabei kann die Zielbitrate in einem Bereich zwischen 4.000 und 600.000 bps eingestellt werden (ein Datenstrom von Mono-daten mit einer Auflösung von 16 bit und 48 kHz erzeugt 768.000 bps). Die gleichen Audiodaten werden im *OpusLQ* Block mit einer konstanten Bitrate von 4000 bps zu einem niederqualitativen Datenstrom komprimiert.

Die hochqualitativen Daten werden in *VoicePacket* Pakete gekapselt und eine fortlaufende Sequenznummer vergeben. Niedrigqualitative Daten werden entsprechend in *LQVoicePacket* Pakete gekapselt und erhalten die gleiche Sequenznummer. Beide Pakete werden anschließend dem *Sender* zum Versand übergeben.

EMPFANGSSEITE Ein großes Problem bei einem paketisierten, kontinuierlichen Datenstrom von Echtzeitdaten stellt die Schwankung der Ankunftszeiten der Pakete (Jitter) dar. Diese ergibt sich aus den unterschiedlichen Paketlaufzeiten zwischen Sender und Empfänger über unterschiedliche Pfade. Insbesondere bei der Vorwärtsfehlerkorrektur wird dieses Problem noch verschärft, da zur Rekonstruktion eines früheren Paketes eine gewisse Anzahl an nachfolgenden Paketen angekommen sein muss.

Daraus lässt sich auch erkennen, dass Pakete nicht mehr in der Reihenfolge ankommen müssen in der sie versandt wurden. Soll das System auch zur interaktiven Sprachkommunikation geeignet sein, darf eine gewisse Verzögerung zwischen der gesprochenen Sprache und dem Abspielen beim Empfänger ("mouth-to-ear") nicht überschritten werden. Die ITU empfiehlt in [ITU 03] einen Grenzwert von 400 ms für eine ausreichende bzw. 150 ms für eine zufriedenstellende Kommunikation.

⁵ Puls Code Modulation

⁶ Erklärung zu Opus

⁷ Nativ bezeichnet unter Android bzw. Java Teile des Programms, die außerhalb der Java virtual Machine laufen und daher auch in Programmiersprachen wie C und C++ geschrieben sein können. Damit lässt sich auch hardwarenäherer, performanterer Code nutzen.

⁸ Url: <http://downloads.xiph.org/releases/opus/opus-1.1.1.tar.gz> (abgerufen am 17.5.2016)

Die Klasse *VoiceBuffer* (vgl. Abbildung 5.2) implementiert einen Paketpuffer, der die Pakete wieder in die richtige Reihenfolge bringt. Außerdem kann eine maximale Verzögerung vorgegeben werden innerhalb der fehlende Pakete noch für die Wiedergabe bereitgestellt werden können. Um diese Funktionalität umzusetzen wurden mehrere Möglichkeiten getestet:

MAXIMALE ANZAHL Anhand der vorgegebenen Anzahl an Paketen pro Sekunde kann bei bekannter Verzögerungsdauer berechnet werden wie viele Pakete maximal im Puffer sein können. Kommt ein neues Paket an wird es gemäß der Sequenznummer einsortiert und das Paket mit kleinster Nummer aus dem Puffer entnommen. Dieses wird dann dem *VoicePlayer* zur Wiedergabe übergeben. Kommen für einen längeren Zeitraum keine Pakete an führt dies zu einem Puffer Unterlauf in der Wiedergabeeinheit und die Wiedergabe bricht ab, obwohl noch Pakete im Puffer wären.

ENTNAHME BEI BEDARF Die oben genannten Probleme des Puffer Überlaufs führen zu einem weiteren Ansatz bei dem Pakete aus dem Puffer entnommen werden wenn ein Unterlauf im *VoicePlayer* droht. Dazu wird die Wiedergabe am Empfangsgerät um die entsprechende Pufferverzögerung später begonnen. Wenn bei drohendem Unterlauf das Paket mit der nächsten Sequenznummer fehlt, wird es durch einen ,der Dauer entsprechenden, Block Stille ersetzt. Die von Android bereitgestellte Funktionalität zur Unterlauferkennung erwies sich in unseren Versuchen als unzuverlässig und führte zu stockender Wiedergabe oder gänzlichem Abbruch.

REGELMÄSSIGE ENTNAHME Da die Unterlauferkennung von Android nicht zuverlässig funktionierte, wurde diese durch eine regelmäßig Entnahme durch eine zusätzliche Ausführungseinheit ersetzt. Mit Hilfe der Paketrage lässt sich die mittlere Periodendauer berechnen in der Pakete entnommen werden müssen. Da es sich bei Android (bzw. dem darunterliegenden Linux Kernel) um ein Präemptives Multitasking System handelt, können Ausführungseinheiten unterbrochen werden. Selbst mit der höchsten Priorität war es nicht möglich die Ausführung in den geforderten kurzen Zeitintervallen zu erreichen⁹.

BERECHNUNG ABSPIELZEITPUNKT Um die längeren Unterbrechungen bei der regelmäßigen Entnahme auszugleichen müssen mehrere Pakete aus dem Puffer entnommen werden. Dazu wurden beim Empfang der Audiodaten der Abspielzeitpunkt berechnet. Der

⁹ Insbesondere die Ausführung des so genannten "Garbage Collectors" (Müllsammelner) unterbrach die Anwendung für die Dauer mehrerer Pakete.

dazu benötigte angenommene Anfangszeitpunkt des Datenstroms wird bei Empfang des ersten Pakets zurückgerechnet. Da dieser durch Laufzeitunterschiede schon nicht korrekt sein muss und die Samplingfrequenzen nicht exakt identisch sind, wird der Zeitpunkt kontinuierlich, tiefpassgefiltert nachgeführt. Die Pakete werden nun gemäß ihres Abspielzeitpunktes im Puffer einsortiert. Zum Zeitpunkt der Entnahme, werden alle Pakete deren Abspielzeitpunkt verstrichen ist, aus dem Puffer an den *VoicePlayer* weitergegeben. Dieser Ansatz erwies sich als zuverlässigster und wurde für alle folgenden Versuche verwendet.

Ein weiterer Vorteil der Pufferlösung ist die einfache Integration der MDC-Funktionalität. So werden gleichermaßen LQ- und VoicePakete an den Puffer gegeben. Anhand der Sequenznummer wird überprüft, ob das entsprechende Paket schon im Puffer liegt. Falls ja, werden LQVoice-Pakete durch die höherqualitativen VoicePakete ersetzt.

Die Einheit *VoicePlayer* dient dem Abspielen der VoicePakete. Dazu werden die Pakete entgegengenommen und der Datenteil extrahiert. Im Block *Opus* werden die komprimierten Daten wieder zu PCM-Daten dekodiert. Anschließend werden diese zur Wiedergabe durch das Android System an eine Instanz des *AudioPlayers* gegeben.

5.1.2 *NaiveBayesForwarder*

Der Weiterleitungsalgorithmus, der in Kapitel 3 entwickelt wurde, ist ebenfalls in **Adhoc3** implementiert worden. Dazu musste ein neues *ForwarderProtocol* (vgl. Abschnitt 2.3) geschrieben werden.

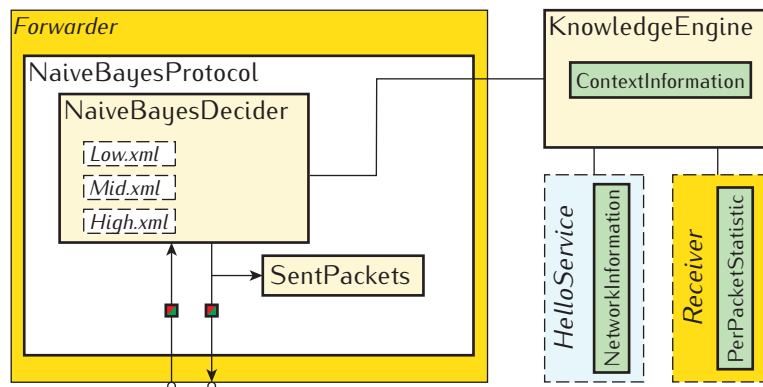


Abbildung 5.3: Detailansicht des NaiveBayes Weiterleiters.

Abbildung 5.3 zeigt das entsprechende Detail aus Abbildung 5.1. Das zentrale Element des Protokolls ist der *NaiveBayesDecider*. Dieser lädt zur Laufzeit drei Konfigurationsdateien, in der Abbildung mit “Low.xml”,

“Mid.xml” und “High.xml” bezeichnet. Darin sind die Ergebnisse der Lernphase des Naive Bayes-Entscheidungers aus Unterabschnitt 3.3.3 hinterlegt.

Da viele Attribute auf den Android Geräten nicht, oder nur mit tiefer gehenden Eingriffen ins System zur Verfügung stehen (wie z. B. die Position innerhalb von Gebäuden oder die Signalstärke), wurde nur eine Auswahl von Attributen verwendet (vgl. Unterabschnitt 3.3.2):

- Stromversorgung
- Datenverkehr
- Anzahl Duplikate
- Minimaler Grad der Nachbarn
- Verbleibende Nachbarn nach einem Paket
- Verbleibende Nachbarn
- Anteil Verbleibender Nachbarn
- Anteil 2-hop Nachbarn

Mit diese Untermenge an Features wurde wie in Unterabschnitt 3.3.3 beschrieben eine Optimierung der Attribut-Auswahl hinsichtlich der Erkennungsrate des Entscheiders durchgeführt. Die Gewichtungsvektoren für die Prioritäten wurden dabei identisch zu 3.3.3 gewählt.

Als Entscheidungsbasis für den *NaiveBayesDecider* dient zum einen die Klasse *SentPackets*, in der alle weitergeleiteten Pakete eingetragen werden. Damit kann verhindert werden, dass das gleiche Paket erneut weitergeleitet wird. Zum anderen benötigt der Entscheider die Werte der Attribute für den Naive Bayes Algorithmus. Wie bereits in OMNeT++ wurde hierfür eine zentrale Informationsstelle, die *KnowledgeEngine* vorgesehen. Im Gegensatz zur Implementierung im Simulator werden in Ad-hoc3 keine Informationsupdates über eine einheitliche Schnittstelle eingefügt, sondern aus Performancegründen dort gespeichert wo die Informationen anfallen. In Abbildung 5.3 ist dies durch die Verbindung zu den gestrichelt dargestellten Klassen und den entsprechenden grün hinterlegten Speichereinheiten *ContextInformation*, *NetworkInformation* und *PerPacketStatistic* (vgl. Unterabschnitt 2.2.2) dargestellt.

5.2 MESSUNGEN

Zur Demonstration der prototypischen Implementierung wurde ein häusliches Szenario, wie es z. B. für ein “Smart Home” denkbar ist,

gewählt. Darin wurde untersucht wie sich die Anwendung und das Übertragungsprotokoll in zwei verschiedenen Testfällen verhält. Zum einen ein statischer Fall, in dem sich insbesondere das Lernverhalten von iFec zeigen lässt. Zum anderen ein mobiler Fall um die Adaptivität der Protokollkombination zu zeigen.

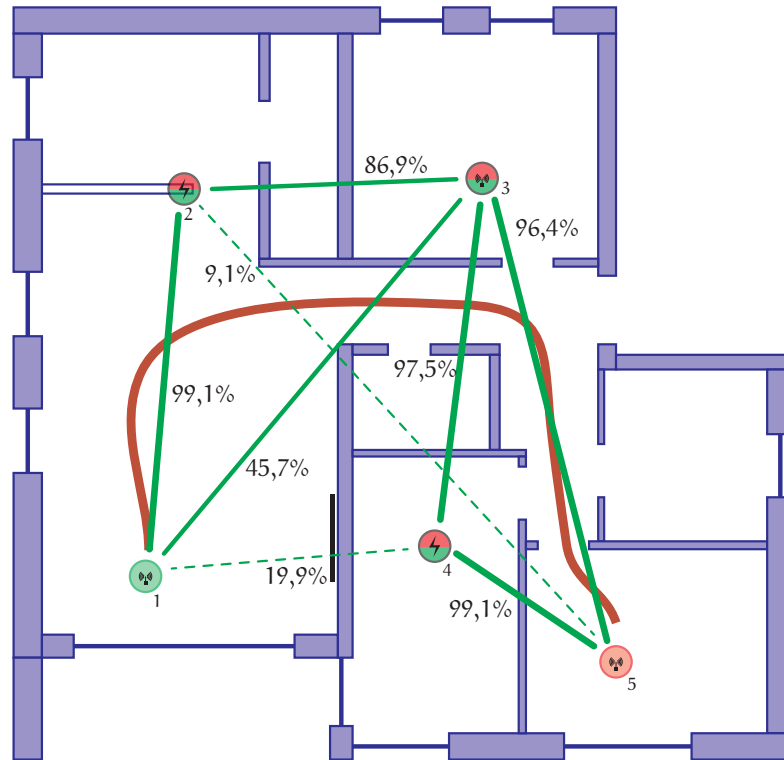


Abbildung 5.4: Versuchsanordnung zur Evaluation der prototypischen Implementierung. Eingetragen sind die durchschnittlichen Zustellraten zwischen den Knoten in % und die Strecke für den Mobilitätstest in rot.

In der Testumgebung, einer 4-Zimmer Wohnung mit $\approx 100 \text{ m}^2$, wurden 5 Smartphones mit **Adhoc3** verteilt. Abbildung 5.4 zeigt den Grundriss der Wohnung. Neben den Standorten der Geräte sind auch die PDR der einzelnen Verbindungen (grün) eingetragen. Die Knoten ② und ④ wurden über ein Netzteil mit Energie versorgt. Der grün eingefärbte Knoten ① diente für die Versuche als Sender. ⑤ in rot als Empfänger. Der im mobilen Szenario zurückgelegte Weg ist als roter Pfad eingezeichnet.

Ein Wert, der in Abbildung 5.4 auffällt, ist die Zustellrate zwischen ① und ④. Trotz räumlicher Nähe gehen viele Pakete verloren. Dies liegt zum einen an einer tragenden Wand, zum anderen am LCD-Fernseher (in schwarz angedeutet), die sich zwischen Sender und Empfänger befinden.

5.2.1 Statischer Fall

Zuerst wurde die Leistungsfähigkeit des Protokolls in einer statischen Testumgebung untersucht. Dazu wurden 5 Smartphones mit der Testapplikation wie im vorherigen Abschnitt beschrieben in einer 4 Zimmer Wohnung platziert. Mit Hilfe des "ExperimenterService" (siehe Unterabschnitt 27 auf Seite 27.) wie folgt festgelegt: Über einen Zeitraum von 300 s nimmt die zuvor beschriebene Audio-Kommunikations-Applikation Audiodaten auf und generiert daraus pro Sekunde 50 Pakete hoher Qualität (HQ), die mit mittlerer Priorität und 50 Pakete niedriger Qualität, die mit hoher Priorität versandt werden.

Die Knoten ②, ③ und ④ wurden als Weiterleiter konfiguriert, die das NB-Protokoll verwenden. Auf allen Knoten wird iFec verwendet um ausgehende Paketströme mit Redundanz zu versehen.

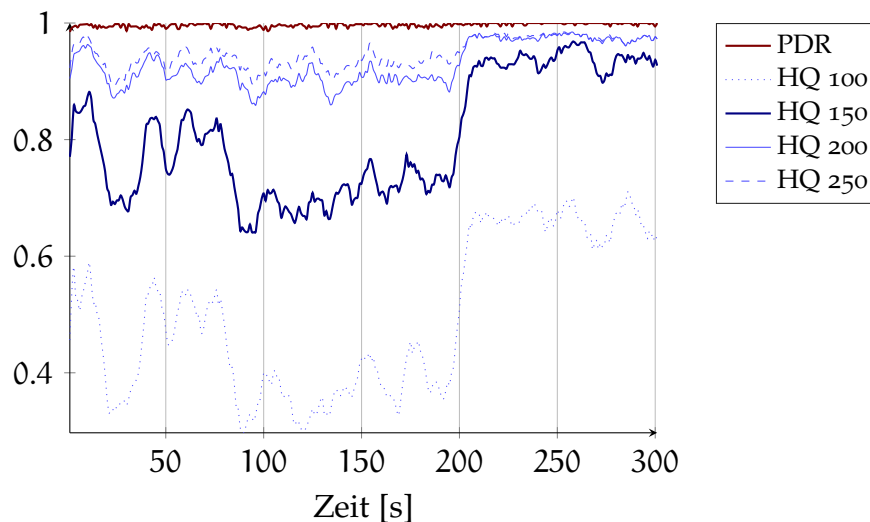


Abbildung 5.5: Zustellrate und Anteil des höher-qualitativen Datenstroms (HQ) für unterschiedliche Puffergrößen (100 - 250 ms).

Knoten ⑤ diente bei diesem Versuch als Empfänger auf dem die ankommenden Datenströme ausgewertet werden. Abbildung 5.5 zeigt die erreichte Zustellrate (in rot) des originalen Audio-Stroms¹⁰. Wie in den vorangegangenen Abschnitten beschrieben wird dieser durch die Rekombination der hoch- und nieder-qualitativen Audiodaten gebildet. Pakete niedriger Audio-Qualität werden dabei von höher-qualitativen

¹⁰ Viele der Diagramme in diesem und den folgenden Abschnitten zeigen einen zeitlichen Verlauf verschiedener Aspekte des Protokolls. Die Werte sind dabei immer entweder als Mittelwerte der vorangegangenen 10 s zu verstehen, oder wurden über dieses Zeitintervall gebildet (z. B. für die PDR).

Daten verdrängt. Dadurch konnte eine sehr hohe Zustellrate von durchschnittlich 99,67 %¹¹ erreicht werden.

In Abbildung 5.5 ist ebenfalls der Anteil an hoch-qualitativen Audio-Paketen (HQ-Rate) in blau dargestellt. Für die Auswertung wurden verschiedene Puffergrößen (vgl. Abbildung 5.1.1) zwischen 100 und 250 ms angenommen. Wie man sieht steigt die HQ-Rate mit längeren Pufferzeiten. Dies liegt zum einen am bevorzugten Versand hoch-priorer Pakete (im Fall der Audio Applikation den LQ-Paketen). Diese werden so früh wie möglich versendet bzw. weitergeleitet, wohingegen mittel-priorer Pakete (HQ) 30 ms verzögert werden¹². Zum anderen wird dem Fehlerkorrekturmechanismus bei höherer Puffergröße mehr Zeit gegeben um Pakete wiederherzustellen.

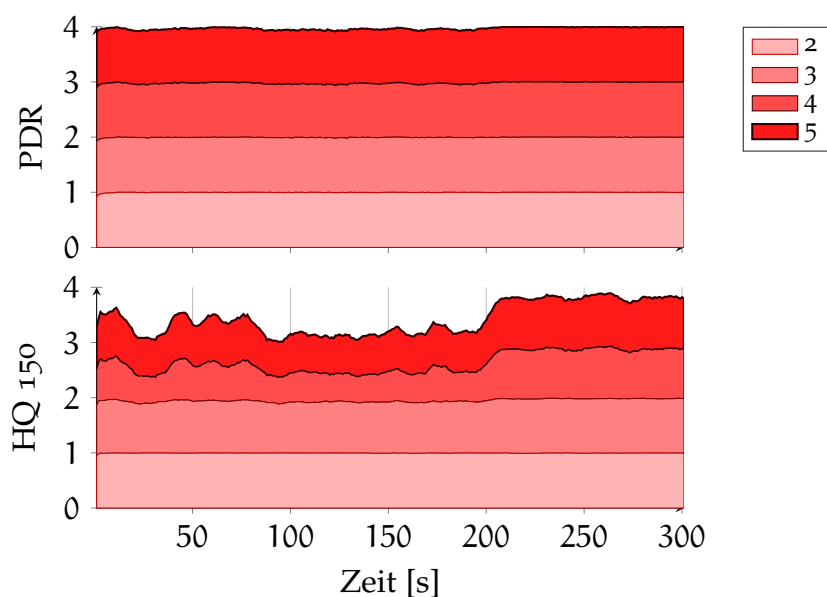


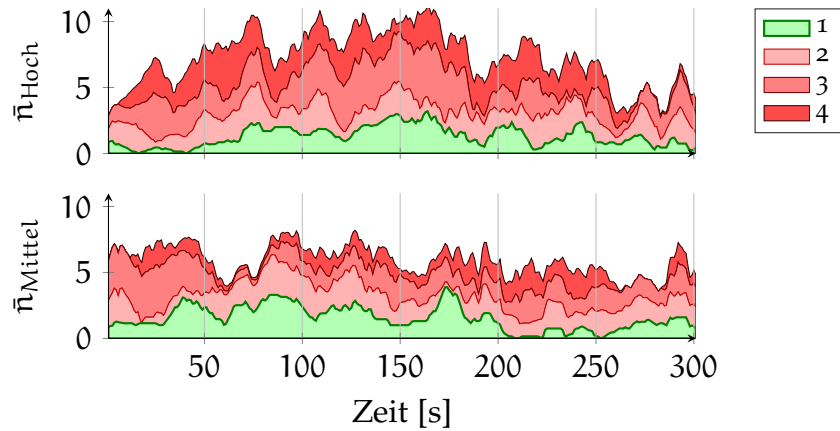
Abbildung 5.6: Zustellrate und Anteil des höher-qualitativen Datenstroms (HQ) für alle Empfänger des Netzwerks als Stapeldiagramm.

Betrachtet man die erreichte HQ-Rate, so fällt auf, dass ab ca. 200 s Versuchszeit der Anteil an hoch-qualitativen Paketen insbesondere für die niedrigeren Puffergrößen deutlich zunimmt. Eine Erklärung für dieses Verhalten findet sich weder im Weiterleitungsverhalten noch in einer starken Zunahme der gewählten Redundanz. Vielmehr kann man an Abbildung 5.6 erkennen, dass Knoten ④ um 200 s deutlich mehr HQ-Pakete empfangen kann. Obwohl sich im Laufe des Versuchs die Umgebung nicht offensichtlich verändert hat, scheinen sich die Empfangsbedingungen zwischen Knoten ① und ④ nach 200 s verbessert zu haben. Da sich

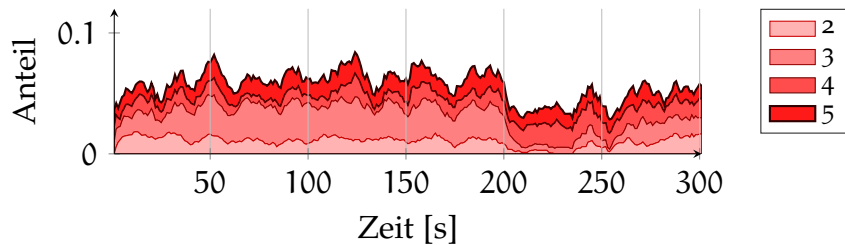
¹¹ Die Puffergröße wurde für die PDR-Bestimmung zu 150 ms angenommen.

¹² Dieses Verhalten dient zum einen als minimale Datenverkehrssteuerung, zum anderen können dadurch Burst-artige Störeinflüsse vermindert werden.

dadurch der Weg der Pakete zwischen ① und ⑤ verkürzt, können auch bei kürzeren Pufferzeiten mehr hoch-qualitative Audio-Daten abgespielt werden.



(a) $\bar{n} = n - 6$ des iFec Protokolls für hoch- und mittel-priore Pakete



(b) Anteil wiederhergestellter Pakete

Abbildung 5.7: Zeitlicher Verlauf der n des iFec Protokolls und der Anteil wiederhergestellter Pakete als Stapeldiagramme. Zur besseren Übersicht wurde n um 6 (keine Redundanz) verschoben.

In Abbildung 5.7a ist der zeitliche Verlauf des Redundanzgrades n des iFec Protokolls dargestellt. Zur besseren Übersicht ($6 \leq n \leq 12$) wurde n um 6 verschoben und für alle Sender des Netzwerks als Stapeldiagramm dargestellt. Man kann erkennen, dass n über den gesamten Versuchslauf auf einem verhältnismäßig niedrigen Niveau liegt. Für hoch-priorisierte Pakete ist n durchschnittlich 7,81, für mittel-priorisierte 7,47.

Wie man an Abbildung 5.6 erkennen kann, scheint dieser Redundanzgrad ausreichend zu sein um eine hohe Zustellrate zu erzielen. Abbildung 5.7b zeigt den Anteil an wiederhergestellten Paketen. Diese korrespondieren gut mit den in Abbildung 5.4 dargestellten Empfangsbedingungen. Lediglich Knoten ③ muss eine nennenswerte Anzahl (durchschnittlich 2,6 % in den ersten 200 s) Pakete wiederherstellen.

Wie bereits zuvor argumentiert, scheint ab 200 s eine Verbesserung der Empfangsbedingungen einzutreten. So sinkt zum einen der Redundanz-

grad für hoch-priore Pakete, zum anderen kann Knoten ④ eine höhere Anzahl an Paketen wiederherstellen.

5.2.2 Mobiler Fall

Als zweites Experiment wurde ein mobiles Szenario untersucht. Dabei wurde der Versuchsaufbau des statischen Falls übernommen. Lediglich Knoten ⑤ wurde während der Durchführung bewegt. Das Gerät wurde dabei von einer Versuchsperson in Brusthöhe ca. 40 cm vor dem Körper getragen. Der abgeschriftene Pfad ist in Abbildung 5.4 in rot eingetragen. Zu Beginn des Versuchs befand sich Knoten ⑤ an Position des Senders ①. Während des 300 s dauernden Experiments wurde der Pfad mehrfach in Schrittgeschwindigkeit abgeschriftet.

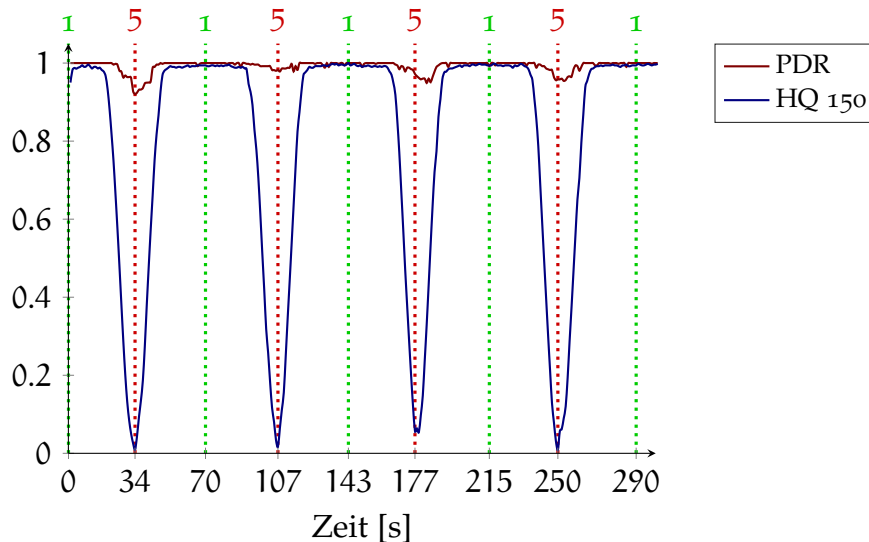
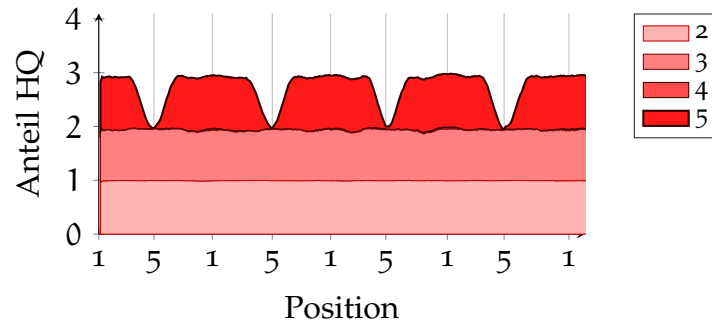


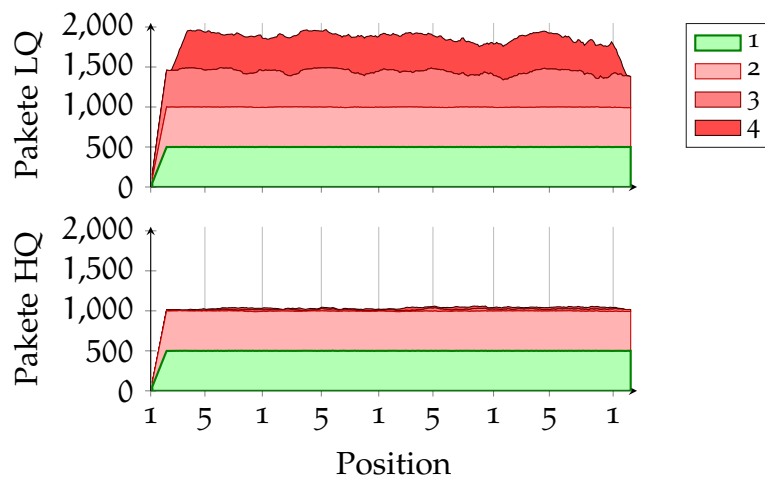
Abbildung 5.8: Zustellrate und Anteil des höher-qualitativen Datenstroms für 150 ms Puffergröße.

Abbildung 5.8 zeigt die erzielte Zustellrate und den Anteil hochqualitativer Datenpakete an den Audiodaten. Man erkennt, dass durchgehend eine hohe Zustellrate ($> 97\%$) erreicht wurde. Bei genauerem Hinsehen erscheinen die Senken an der weitest entfernten Position 5 schmäler und flacher als zu Beginn des Experiments.

Im Gegensatz zum statischen Setup kann bei diesem Versuch kein durchgängig hoher Anteil an hochqualitativen Audio Paketen erzielt werden. Wie man in Abbildung 5.8 erkennt, geht er auf bis zu 0 % zurück, wenn sich der Empfänger an der entferntesten Position 5 befindet. Um dieses Verhalten zu analysieren ist in Abbildung 5.9 zum einen die HQ-Rate für jeden Empfänger aufgeführt, zum anderen die versandten "Voice-Pakete" in hoher und niedriger Qualität.



(a) Anteil des höher-qualitativen Datenstroms für 150 ms Puffergröße auf allen Empfängern.



(b) Gesendete Pakete hoher (HQ) und niedriger (LQ) Qualität

Abbildung 5.9: Anteil hochqualitativer Audiopakete und gesendete "Voice-Pakete" für den mobilen Testfall. Beide Datensätze sind als Stapeldiagramme dargestellt.

Wie man in Abbildung 5.9a sieht, empfangen Knoten ② und ③ einen sehr hohen Anteil an hoch-qualitativen Paketen. In Abbildung 5.9b sind die gesendeten (weitergeleiteten) "Voice-Pakete" dargestellt. Betrachtet man die Anzahl hochqualitativer Audiodaten, so scheint Knoten ③ diese Pakete nicht weiterzuleiten. Als Konsequenz sinkt die HQ-Rate auf dem mobilen Gerät (Knoten ⑤) sobald sich dieses außerhalb der Reichweite von ① oder ② befindet.

Im Gegensatz dazu werden Pakete mit hoher Priorität (und niedriger Qualität) von Knoten ③ weitergeleitet. Daran erkennt man gut, dass das NB-Protokoll tatsächlich für unterschiedliche Prioritäten unterschiedliche Entscheidungen trifft.

Da die individuellen Weiterleitungsentscheidungen vom aktuellen Kontext und eventuell bereits empfangenen Duplikaten abhängt, fällt eine genaue Analyse, wie es zu der geringen Anzahl weitergeleiteter,

hochqualitativer Pakete kam, schwer. Ein Attribut das in die Entscheidung eingeht ist die Stromversorgung des Geräts. Im Gegensatz zu Knoten ① und ② besitzt der Teilnehmer ③ keine externe Stromversorgung. Wie bereits in Abbildung 3.11, Unterabschnitt 3.3.3 gezeigt wurde, leiten Knoten bei einer höheren Gewichtung des Energieverbrauchs (was bei den hochqualitativen Paketen der Fall ist) Pakete eher weiter wenn sie eine externe Stromversorgung besitzen. Bei den Paketen niedriger Qualität fällt das Attribut "Stromversorgung" weniger ins Gewicht.

5.3 ZUSAMMENFASSUNG

In diesem Kapitel wurde eine Echtzeit-Sprachapplikation vorgestellt, die das **NBIfec** -Protokoll nutzt um zuverlässige Sprachkommunikation in einem MANet zur Verfügung zu stellen. Die Applikation auf Basis des **Adhoc3** -Frameworks verwendet MDC zur weiteren Steigerung der Zuverlässigkeit der Sprachübertragung. Den Paketen der beiden resultierenden Datenströme werden unterschiedliche Prioritäten zugewiesen: hochqualitative Audiodaten werden mittel, niedrigqualitative hoch priorisiert.

Zu Beginn des Kapitels wurde der Aufbau der Sprachapplikation und deren Einbindung in das **Adhoc3** -Framework beschrieben. Neben der Beschreibung der Komponenten wurden auch die verschiedenen getesteten Ansätze zur Implementierung eines Paketpuffers zur Kompensation des Jitters und Rekombination der MDC-Datenströme vorgestellt.

Zur vollständigen Implementierung des **NBIfec** -Protokolls musste noch das NB-basierte Weiterleitungsprotokoll umgesetzt werden. Dazu wurde eine "Knowledge Engine" mit einer Auswahl der Features aus Kapitel 3 in **Adhoc3** implementiert. Mit diesem Featuresatz wurde erneut verschiedene Bayes-Klassifizierer trainiert, die die Entscheidungsbasis des NB-Protokolls bilden.

Als Testumgebung für das **NBIfec** -Protokoll wurde eine 100 m² große Wohnung gewählt. Darin wurden 5 Geräte mit der Demonstratorsoftware verteilt. In zwei unterschiedlichen Szenarien, einem statischen und einem mobilen, konnte die Leistungsfähigkeit des Protokolls für die Echtzeit-Sprachkommunikation gezeigt werden.

Im statischen Versuchsaufbau wurde eine durchgängig hohe Zustellrate auf allen Geräten des Netzwerks erreicht. Wurde den Paketen eine längere Ende-zu-Ende Verzögerung zugestanden (über einen größeren Paketpuffer), so konnte auch ein Anteil > 90 % an hochqualitativen Sprachpaketen erzielt werden. Für den mobilen Testfall wurde ebenfalls eine sehr hohe Zustellrate erzielt. Jedoch konnte nicht immer die hohe Sprachqualität wiedergegeben werden. Dieses Verhalten konnte auf einen Empfänger zurückgeführt werden, der zwar hochprior Pa-

kete weiterleitete, mittelpriore jedoch verwarf. Daran konnte gut die Berücksichtigung der unterschiedlichen Prioritäten gezeigt werden.

ZUSAMMENFASSUNG UND AUSBLICK

Auch 18 Jahre nach der technologischen Vision von Roel Pieper, der “Ambient Intelligence” (vgl. [Piep98]), sind diese intelligenten Umgebungen noch keine Realität. Viele der zugrunde liegenden Technologien sind mittlerweile verfügbar und decken unter den Stichworten “Internet of Things” oder “Smart Home” einige der Aspekte von AmI ab. Dies erfolgt aber weitestgehend durch die Steuerung mittels zentraler Server im Heimnetz oder Internet.

Dem Gedanken der umgebenden Intelligenz werden dezentrale Ansätze eher gerecht. Bei diesen kommunizieren die Geräte untereinander und benötigen keine stetige Verbindung zu zentralen Systemen. Oft werden aufgrund der limitierten Ressourcen bei den Geräten dieser Szenarien infrastrukturlose Netzwerke verwendet, deren Konfiguration sich an die umherlaufenden Menschen und der damit einhergehenden wechselnden Sichtbarkeit der Geräte dynamisch anpasst.

Eine grundlegende Technologie bei dieser Art drahtloser Netzwerke ist die Broadcast-Kommunikation. Zum einen bildet diese Form der Datenübertragung die Grundlage vieler Uni- und Multicast-Protokolle, zum anderen bietet sie ein großes Potential für die energieeffiziente Übertragung von Daten. Bei drahtlosen Übertragungen empfangen alle Geräte, die sich innerhalb des Senderradius¹ befinden, die gesendeten Daten. Durch konsequentes Ausnutzen dieses Effekts lassen sich sehr energiesparende Protokolle realisieren.

Zur Realisierung von AmI wurde daher in dieser Arbeit als weiterer Baustein ein Broadcast-Übertragungsprotokoll entwickelt, das optimal an die Anforderungen solcher infrastrukturlosen Netzwerke angepasst ist. Zu Beginn dieser Arbeit wurden folgende Forderungen an das Protokoll formuliert:

PRIORITÄTSBEWUSST Bei der Generierung von Daten kann eine “Priorität” definiert werden, die die Wichtigkeit der Daten beschreibt. Diese kann entweder vom Applikationsdesigner vorgegeben werden, oder durch Algorithmen bestimmt werden. Das in dieser Arbeit entwickelte Protokoll soll die Priorität der Daten beachten und die Protokollparameter entsprechend wählen.

¹ Hierbei ist die Fläche gemeint innerhalb derer ankommende Pakete erfolgreich dekodiert werden können.

EFFIZIENT Insbesondere die energetische Effizienz ist von großer Bedeutung für mobile Netzwerke. Das Übertragungsprotokoll soll daher die Energieversorgung der Netzwerkteilnehmer beachten und ebenfalls in Abhängigkeit der Priorität der Daten einen möglichst geringen Energieverbrauch realisieren.

HOHE ZUVERLÄSSIGKEIT Je nach Wichtigkeit der Daten ist eine unterschiedliche Zuverlässigkeit in der Übermittlung der Daten erforderlich. Das zu entwickelnde Protokoll soll einerseits möglichst zuverlässig Daten übertragen, andererseits in Abhängigkeit von Energiebedarf und Priorität Einschränkungen machen dürfen.

OPTIMISTISCH Die Grundannahmen bei der Entwicklung selbstorganisierender Systeme für AmI sollte optimistisch sein. Applikationen bei denen eine absolut zuverlässige Datenübertragung vorausgesetzt wird, sollten nicht Teil einer AmI-Umgebung sein ([Herro6]). Das in dieser Arbeit entwickelte Protokoll richtet sich an Applikationen, die optimistische Annahmen über die Übertragungseigenschaften treffen können.

INTELLIGENT Intelligente Umgebungen werden von den Nutzern als solche wahrgenommen, indem sie möglichst adaptiv und vorausschauend agieren und aus vergangenen Erfahrungen selbstständig lernen. Das zu entwickelnde Protokoll soll sich ebenfalls möglichst adaptiv verhalten und aus einer Vielzahl von erfassten Attributen eine optimale Strategie entwickeln.

EIGENENTWICKLUNGEN

Die Ergebnisse der Protokollentwicklung wurden zum einen mittels Simulationen, zum anderen mit Hilfe eines realen Testbeds aus Mobilgeräten verifiziert. Zur Netzwerksimulation kam der Simulator OMNeT++ zum Einsatz. Dieses Framework bietet bereits einen kompletten Netzwerkstack von der Simulation des Übertragungskanal über eine IEEE-802.11 Implementierung bis hin zu Applikationsprotokollen.

Als schichtübergreifende Datenschnittstelle wurde OMNeT++ um eine s.g. "Knowledgeengine" erweitert. Diese nimmt über Updatemechanismen Informationen aus verschiedenen Modulen entgegen und kann damit eine Vielzahl verschiedener Anfragen des Übertragungsprotokolls beantworten. Dazu gehören "Pro-Paket"-Informationen, Informationen zum umliegenden Netzwerk und schließlich auch noch Informationen zum Kontext des Geräts. Diese bilden die Entscheidungsgrundlage für die Algorithmen des Broadcast-Protokolls.

Zur Realisierung eines Testbeds, das aus Android-Smartphones gebildet wird, wurde eine eigene Android-Applikation entwickelt. Die Software **Adhoc3** realisiert ein modulares Framework, das zum einen leicht erweiterbar ist, zum anderen das komplette Management des infrastrukturlosen IEEE-802.11 Netzwerkes übernimmt. Neben einer Referenzimplementierung des Protokolls wurden auch eine Beispielapplikation zur Echtzeit-Audioübertragung implementiert.

ERGEBNISSE

Um den Anforderungen, die AmI an die Netzwerkübertragung stellt gerecht zu werden, wurde für das in dieser Arbeit entwickelte Protokoll **NBIfec** ein zweigeteilter Ansatz gewählt: Ein Naive-Bayes-basiertes Weiterleitungsprotokoll leistet eine ressourceneffiziente Verteilung von Broadcast-Daten. Darauf aufbauend wird die Zuverlässigkeit der Übertragung durch den Einsatz eines intelligenten Vorwärtsfehlerkorrektur-Verfahrens (**iFec**) verbessert.

SELEKTIVE WEITERLEITUNG Mit Hilfe eines Mehrzieloptimierungsverfahrens wurde in einem gestaffelten Prozess eine optimale Weiterleitungsstrategie hinsichtlich dreier Kriterien, Zustellrate, Energieverbrauch und Ende-zu-Ende Verzögerung, hergeleitet. In einer Vielzahl von Simulationsläufen wurden für viele verschiedene Szenarien optimiert und die Resultate an 21 Stellen der Paretofront ausgewertet. Dabei wurde jede Weiterleitungsentscheidung inklusive dem aktuellen Kontext mitprotokolliert.

Aus den Protokolldateien und Netzwerkkonfigurationen wurden drei Naive Bayes Klassifizierer trainiert, die unterschiedlichen Gewichtungen der drei Optimierungszielen entsprechen. Ein genetischer Algorithmus zur Auswahl der relevanten Attribute konnte die Klassifizierungsgenauigkeit noch einmal steigern. Die drei Klassifizierer wurden zu einem Protokoll zusammengefasst, das je nach Priorität der Daten eine hohe Zuverlässigkeit oder eine hohe Energieeffizienz zum Ziel hat.

Das Naive-Bayes-basierte Protokoll leistet insbesondere für hoch priorisierte Daten eine hohe Zuverlässigkeit bei vergleichsweise geringem Energieverbrauch. Für die anderen Prioritäten kann ebenfalls eine hohe Zustellrate erzielt werden, jedoch benötigt das Protokoll in Szenarien mit hohem Datenaufkommen und hohen Knotendichten mehr Energie als das angepasste **SBA** Protokoll, das durch seinen dynamischen Ansatz Paketduplikate effektiver vermeiden kann.

INTELLIGENTE VORWÄRTSFEHLERKORREKTUR Zur weiteren Steigerung der Zuverlässigkeit wurde ein adaptives, prioritätsabhängiges

Verfahren zur Übertragung von Broadcast-Daten entwickelt. Dazu wurden ein selbst-lernendes System implementiert, das proaktiv die Parameter des Reed-Solomon Fehlerkorrekturverfahrens setzen kann. Über eine Gewichtungsfunktion kann prioritätsabhängig eingestellt werden, wie die Kosten-Nutzen-Abwägung zwischen dem Datenmehraufwand durch die Redundanz und der geschätzten Zustellrate zu bewerten ist.

Für das selbst-lernende System wurden verschiedene maschinelle Lernverfahren getestet. Dabei wurden Parametervariationen evaluiert, um eine möglichst hohe Lerngeschwindigkeit bei stabilen Endresultaten zu erhalten. Die Fähigkeit des Naive Bayes-Klassifizierers, unbekannte Parameterkombinationen aus den zugrundeliegenden Normalverteilungen zu schätzen, sowie die damit einhergehende Glättung der Werte, ließen die Wahl auf diesen Algorithmus fallen.

Die Kombination aus Gewichtungsfunktion, Attributen und Lernverfahren wurde zu einem prioritätsabhängigen Fehlerkorrekturprotokoll, genannt **iFec**, zusammengefasst. Dieses wurde zuerst auf Android-basierten Smartphones in einfachen Netzwerken evaluiert. Insbesondere in einem Netzwerk mit zwei Weiterleitern konnte **iFec** den Redundanzgrad sehr gut steuern und sorgte für hohe Zustellraten. Ebenso konnte die Abhängigkeit von den vorgegebenen Prioritäten gezeigt werden. Das Protokoll gewährte den hochpriorären Daten mehr Redundanz und erreichte somit eine höhere aPDR.

In einem letzten Schritt wurde die Leistungsfähigkeit der Kombination von **iFec** mit dem zuvor entwickelten Naive Bayes-basierten Weiterleitungsprotokoll evaluiert. Dazu wurde das Fehlerkorrekturprotokoll in OMNeT++ implementiert und in komplexen Netzwerken mit verschiedenen Paketraten getestet. Es konnte gezeigt werden, dass die Kombination aus dem NB-Protokoll mit **iFec** sehr gute Zustellraten bei geringem Energiebedarf erzielt. Insbesondere benötigt das Fehlerkorrekturprotokoll nur wenig zusätzliche Energie um die aPDR deutlich zu steigern.

PROTOTYPISCHE IMPLEMENTIERUNG Um die Fähigkeiten des entwickelten Protokolls zu zeigen, wurde eine Echtzeit-Sprachapplikation entwickelt, die das **NBIfec**-Protokoll nutzt um zuverlässige Sprachkommunikation in einem MANet zur Verfügung zu stellen. Die Applikation auf Basis des **Adhoc3**-Frameworks verwendet MDC zur weiteren Steigerung der Zuverlässigkeit der Sprachübertragung. Den Pakete der beiden resultierenden Datenströme werden unterschiedliche Prioritäten zugewiesen: hochqualitative Audiodaten werden mittel, niedrigqualitative hoch priorisiert.

Im statischen Versuchsaufbau, einer 100 m² Wohnung, wurde eine durchgängig hohe Zustellrate auf allen Geräten des Netzwerks erreicht.

Wurde den Paketen eine längere Ende-zu-Ende Verzögerung zugestanden (über einen größeren Paketpuffer), so konnte auch ein Anteil $> 90\%$ an hochqualitativen Sprachpaketen erzielt werden. Für den mobilen Testfall wurde ebenfalls eine sehr hohe Zustellrate erzielt. Jedoch konnte nicht immer die hohe Sprachqualität wiedergegeben werden. Dieses Verhalten konnte auf einen Empfänger zurückgeführt werden, der zwar hochpriorre Pakete weiterleitete, mittelpriorre jedoch verwarf. Daran konnte gut die Berücksichtigung der unterschiedlichen Prioritäten gezeigt werden.

AUSBLICK

Auch wenn das entwickelte Protokoll bereits sehr gute Ergebnisse in Simulationen und im realen Einsatz zeigen konnte, kann die Leistungsfähigkeit weiter verbessert werden. Auf der einen Seite können Protokollinterne Verbesserungen durchgeführt werden. Dies betrifft insbesondere die Entscheidungsbasis des NB-basierten Protokolls. Durch eine genaue Analyse des Einsatzszenarios können die Parameter der Netzwerksimulation im Optimierungsschritt möglichst realistisch angepasst werden. Damit sind die in Folge trainierten Naive-Bayse-Entscheider optimal auf den Anwendungszweck angepasst. Ebenfalls für das Training des ML-Verfahrens können weitere Features helfen die Vorhersagen des Algorithmus zu verbessern. Diese können zum einen von Experten aus beispielsweise graphentheoretischen Überlegungen, geometrischen Betrachtungen oder der Analyse von Empfangsbedingungen vorgegeben werden. Zum anderen existieren Ansätze bei denen Features durch ML-Verfahren selbstständig extrahiert werden können.

Auf der anderen Seite kann das Protokoll durch weitere Techniken erweitert werden um Verbesserungen hinsichtlich Zuverlässigkeit und Effizienz zu erzielen. Eine sehr interessante Möglichkeit bieten Datenverkehrssteuerungsverfahren und kontrollierter Medienzugriff. Bei beiden Methoden werden Paketeigenschaften, wie z. B. die Priorität berücksichtigt um ein für die Applikationen erforderlichen oder gewünschten QoS zu erreichen. Damit ist insbesondere bei hohen Kanalauslastungen, bei denen **NBIfec** teilweise Schwächen zeigte, eine Verbesserung der Zuverlässigkeit hochpriorer Pakete zu erwarten.

ANHANG

A.1 ADHOC3

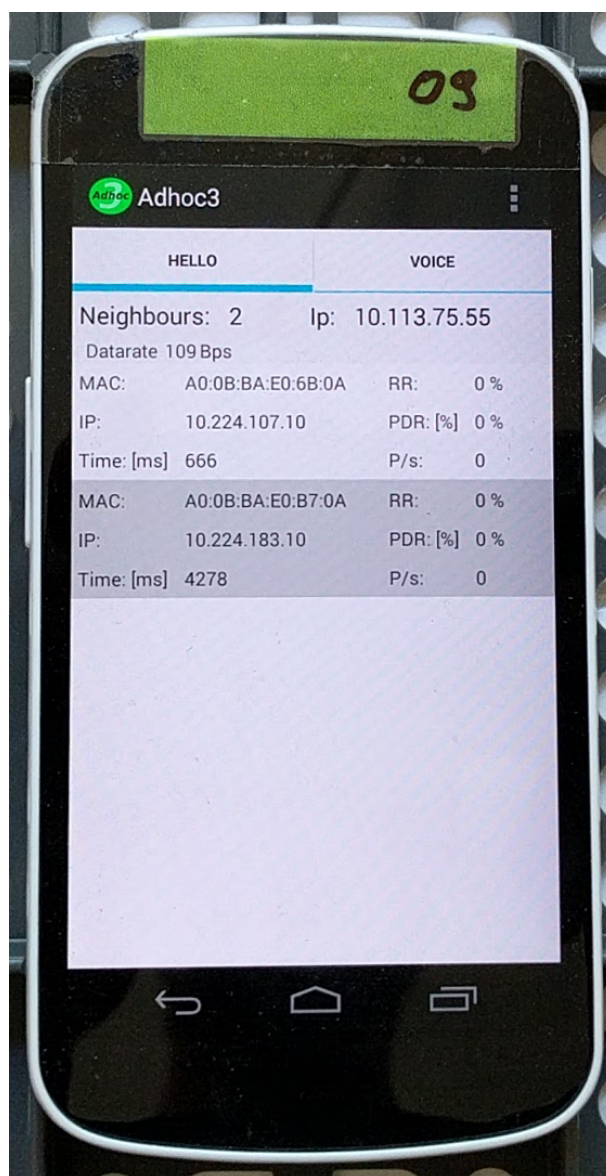


Abbildung A.1: Screenshot der Applikation Adhoc3.

A.2 PAKETLAYOUT

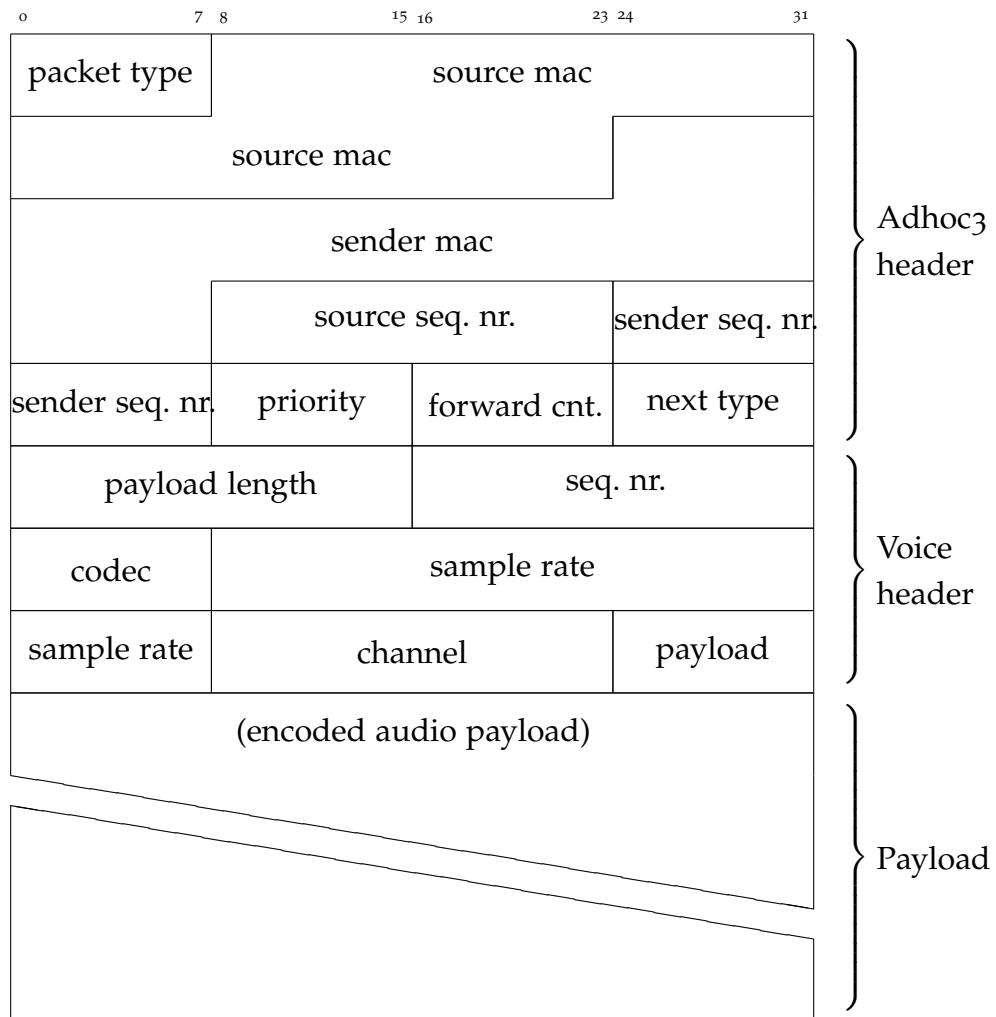


Abbildung A.2: Paket-Layout des VoicePacket.

A.3 PROTOKOLLPARAMETER

Tabelle A.1: Gewählte Parameter für die Protokolle des Stand der Technik.

	τ_A
AAG	99%
	Grenzwert
ABS	0,5
	Grenzwert
DBS	20 m
	$T_{\max}$
SBA	1 ms

A.4 RESULTATE

Tabelle A.2: Resultate für 5 stationäre und 15 batteriegespeiste Knoten im Vergleich mit dem Ergebnis des Optimierungsverfahrens.

	1pps PDR	1pps power	1pps delay	50pps PDR	50pps power	50pps delay
don't forward	0,133	0,00	0,3	0,133	0,00	0,3
flooding	0,676	1,00	2,2	0,295	1,00	3,5
aag	0,540	0,47	1,9	0,298	0,60	2,4
abs	0,659	0,85	2,7	0,297	0,90	3,6
dbb	0,661	0,90	2,7	0,296	0,91	3,6
aid	0,650	0,87	2,7	0,295	0,96	4,0
sba	0,578	0,52	2,4	0,301	0,60	2,8
asba	0,586	0,49	66,1	0,297	0,58	61,5
diba	0,671	0,93	2,7	0,295	0,97	4,0
eca	0,658	0,90	10,4	0,285	0,87	10,2
pmlb	0,675	0,99	2,3	0,240	0,63	1,5
moga	0,703	0,58	9,3	0,331	0,44	5,7

Tabelle A.3: Auswahl der Attribute für den Naive Bayes Klassifizierer mittels eines genetischen Algorithmus. Die Gewichte w werden in der Form $(w_{\text{PDR}}; w_{\text{Leistung}}; w_{\text{Delay}})$ angegeben. Die Attribute mit vorangestelltem “r” kennzeichnen die korrespondierenden “Anteil” Werte.

w	(1;0;0)	(0,7;0,3;0)	(0,5;0,5;0)
Externe Stromversorgung	•	•	•
NumberOfRepetitions	•	•	•
Rss			
Traffic	•	•	•
Distance			
Distance25	•		
Distance50	•	•	•
Distance75	•		•
Distance90	•	•	•
McdsRate			
MinDistance			
AdditionalCoverage	•		
MinNeighbourDegree	•		
rNeighbours90°	•		•
rNeighbours180°	•	•	•
rNeighbours270°	•		
rTwoHop		•	
rRemainingAfterReception			
rRemainingNeighbours	•	•	•

Tabelle A.4: Resultate für 5 statische und 15 mobile Knoten. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen.

	1pps PDR	1pps power	1pps delay	5pps PDR	5pps power	5pps delay	10pps PDR	10pps power	10pps delay	20pps PDR	20pps power	20pps delay
don't forward	0,24	0,00	0,3	0,25	0,00	0,3	0,22	0,00	0,3	0,19	0,00	0,3
flooding	0,80	1,00	1,9	0,74	1,00	2,2	0,62	1,00	2,4	0,49	1,00	2,6
aag	0,67	0,39	1,5	0,63	0,41	1,6	0,54	0,43	1,7	0,43	0,42	1,7
aid	0,73	0,73	2,0	0,70	0,76	2,3	0,59	0,80	2,5	0,48	0,85	2,7
sba	0,73	0,52	1,9	0,69	0,52	2,1	0,58	0,52	2,2	0,46	0,50	2,1
asba	0,72	0,38	38,3	0,68	0,38	44,1	0,58	0,41	48,6	0,46	0,39	47,3
pmlb	0,75	0,83	1,7	0,70	0,85	2,0	0,60	0,88	2,2	0,48	0,95	2,5
dbb	0,80	0,96	2,2	0,74	0,96	2,5	0,62	0,96	2,7	0,49	0,96	3,0
abs	0,77	0,76	2,1	0,72	0,77	2,3	0,60	0,77	2,5	0,48	0,79	2,6
diba	0,76	0,74	2,1	0,72	0,75	2,3	0,60	0,77	2,5	0,48	0,82	2,7
naiveBayes20	0,77	0,63	2,4	0,72	0,72	2,7	0,60	0,74	2,8	0,49	0,74	3,0
naiveBayes14	0,73	0,51	2,2	0,69	0,62	2,5	0,58	0,63	2,7	0,46	0,64	2,6
naiveBayes5	0,71	0,45	2,2	0,68	0,57	2,5	0,57	0,60	2,6	0,46	0,62	2,7
eca	0,78	0,86	7,0	0,72	0,83	7,7	0,60	0,83	8,3	0,46	0,80	8,1

Tabelle A.5: Resultate für 10 statische und 10 mobile Knoten. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen.

	1pps PDR	1pps power	1pps delay	5pps PDR	5pps power	5pps delay	10pps PDR	10pps power	10pps delay	20pps PDR	20pps power	20pps delay
don't forward	0,22	0,00	0,3	0,21	0,00	0,3	0,19	0,00	0,3	0,18	0,00	0,3
flooding	0,76	1,00	1,9	0,66	1,00	2,1	0,57	1,00	2,3	0,45	1,00	2,3
aag	0,64	0,44	1,6	0,55	0,43	1,6	0,48	0,43	1,6	0,39	0,43	1,5
aid	0,70	0,74	2,0	0,61	0,77	2,3	0,54	0,81	2,4	0,43	0,86	2,4
sba	0,71	0,56	2,0	0,59	0,53	2,1	0,51	0,52	2,0	0,41	0,51	1,9
asba	0,69	0,42	41,4	0,59	0,41	52,5	0,51	0,42	47,8	0,41	0,40	40,8
pmlb	0,70	0,85	1,7	0,61	0,85	1,9	0,52	0,84	2,0	0,43	0,90	2,1
dbb	0,76	0,95	2,2	0,66	0,96	2,5	0,56	0,95	2,6	0,44	0,95	2,6
abs	0,74	0,76	2,1	0,63	0,76	2,3	0,54	0,75	2,3	0,44	0,78	2,3
diba	0,72	0,73	2,1	0,63	0,75	2,3	0,54	0,79	2,4	0,44	0,82	2,4
naiveBayes20	0,74	0,68	2,5	0,64	0,75	2,7	0,55	0,77	2,8	0,44	0,75	2,6
naiveBayes14	0,71	0,56	2,4	0,62	0,63	2,5	0,53	0,65	2,6	0,42	0,65	2,3
naiveBayes5	0,70	0,51	2,3	0,61	0,62	2,5	0,53	0,63	2,6	0,42	0,61	2,3
eca	0,74	0,87	7,4	0,64	0,83	8,4	0,54	0,81	8,4	0,42	0,80	7,4

Tabelle A.6: Resultate für 20 statische und 30 mobile Knoten. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen.

	1pps PDR	1pps power	1pps delay	5pps PDR	5pps power	5pps delay	10pps PDR	10pps power	10pps delay	20pps PDR	20pps power	20pps delay
don't forward	0,22	0,00	0,3	0,22	0,00	0,3	0,20	0,00	0,3	0,19	0,00	0,4
flooding	0,98	1,00	2,7	0,82	1,00	19,0	0,53	1,00	57,8	0,32	1,00	76,8
aag	0,88	0,37	2,1	0,82	0,59	5,0	0,60	0,79	20,5	0,37	0,86	37,8
aid	0,97	0,78	2,8	0,83	0,91	14,2	0,54	0,98	53,6	0,32	0,99	75,7
sba	0,96	0,71	2,7	0,84	0,82	10,7	0,57	0,91	39,4	0,35	0,94	60,2
asba	0,96	0,34	26,4	0,92	0,48	52,1	0,70	0,76	111,5	0,41	0,86	139,9
pmlb	0,97	0,99	2,7	0,82	0,99	18,7	0,53	0,99	57,3	0,32	0,99	75,5
dbs	0,98	0,94	3,0	0,83	0,96	16,5	0,54	0,98	53,9	0,32	0,99	74,7
abs	0,97	0,70	2,7	0,84	0,81	8,7	0,57	0,93	38,2	0,34	0,96	62,6
diba	0,97	0,68	2,7	0,85	0,82	10,5	0,55	0,95	47,9	0,33	0,98	72,8
naiveBayes20	0,97	0,66	3,0	0,83	0,84	12,0	0,57	0,89	40,2	0,36	0,91	56,1
naiveBayes14	0,97	0,62	3,0	0,83	0,79	11,2	0,57	0,84	36,3	0,36	0,87	51,8
naiveBayes5	0,97	0,57	3,0	0,83	0,76	10,3	0,57	0,80	33,2	0,36	0,85	49,9
eca	0,98	0,90	6,5	0,82	0,87	19,7	0,56	0,93	48,3	0,34	0,96	71,2

Tabelle A.7: Resultate der prioritätsbasierten Broadcast-Protokolle für unterschiedliche Szenarien. Mittelwerte von je 5 Versuchsläufe mit unterschiedlichen Anordnungen. Die besten Resultate für die PDR sind in **Fett** dargestellt.

	(0,10) PDR	(0,10) Power	(5,5) PDR	(5,5) Power	(0,20) PDR	(0,20) Power	(5,15) PDR	(5,15) Power	(10,10) PDR	(10,10) Power	(20,30) PDR	(20,30) Power
<i>1 pps</i>												
flooding	0.54	1.00	0.43	1.00	0.82	1.00	0.77	1.00	0.74	1.00	0.92	1.00
don't fw	0.27	0.00	0.22	0.00	0.29	0.00	0.24	0.00	0.22	0.00	0.22	0.00
aagP	0.45	0.29	0.37	0.34	0.67	0.28	0.62	0.32	0.59	0.36	0.79	0.34
sbaP	0.48	0.39	0.39	0.42	0.78	0.52	0.71	0.53	0.69	0.58	0.91	0.74
naiveBayesP	0.50	0.43	0.41	0.47	0.78	0.49	0.72	0.51	0.70	0.59	0.92	0.69
<i>20 pps</i>												
flooding	0.36	1.00	0.31	1.00	0.35	1.00	0.34	1.00	0.32	1.00	0.17	1.00
don't fw	0.20	0.00	0.18	0.00	0.20	0.00	0.19	0.00	0.18	0.00	0.17	0.00
aagP	0.34	0.43	0.28	0.54	0.36	0.48	0.34	0.46	0.30	0.46	0.23	0.87
sbaP	0.35	0.46	0.29	0.60	0.38	0.57	0.36	0.53	0.32	0.55	0.20	0.94
naiveBayesP	0.34	0.42	0.29	0.57	0.37	0.67	0.35	0.68	0.33	0.70	0.20	0.91

ABKÜRZUNGSVERZEICHNIS

AIDL	Android Interface Definition Language (Android Schnittstellendefinitionssprache)
AMI	Ambient Intelligence ("Umgebende" Intelligenz)
AOSP	das Android Open Source Project
APDR	PDR auf Applikationsebene, bzw. nach Anwendung des FEC Verfahrens
API	Application Programming Interface (Programmierschnittstelle)
AUC	Area Under Curve (Fläche unter der [ROC] Kurve)
CSMA	Carrier Sense Multiple Access (Mehrfachzugriff mit Trägerprüfung)
CSMA/CA	CSMA mit Kollisionsvermeidung
CSMA/CD	CSMA mit Kollisionserkennung
FEC	Forward Error Correction (Vorwärtsfehlerkorrektur)
GPL	GNU General Public License
GPS	Globales Positionierungssystem
HMI	Human Machine Interface (Mensch-Maschinen-Schnittstelle)
HP	"Hallo"-Paket
IBSS	Independent Basic Service Set ("802.11 - Ad-hoc Modus")
IOT	Internet of Things (Das Internet der Dinge)
IP	Internet Protocol
IPC	Inter-Process Communication (Inter-Prozess-Kommunikation)
ITU	International Telecommunication Union (Internationale Fernmeldeunion)
KE	Knowledge Engine
LTE	Long Term Evolution (auch 3.9G genannt)

MANET	Mobile Ad-hoc Network (Mobiles ad-hoc Netzwerk)
MCDS	Minimum Connected Dominating Set (Kleinste Menge verbindender Knoten)
MDC	Multiple Description Coding (Kodierung mit mehreren "Beschreibungen")
ML	Maschinelles Lernen
MLA	Message Loss Avoidance (Vermeidung von Paketverlust)
MLP	Multi Layer Perceptron (Mehrschicht Perzeptron)
MOGA	Multi-Objective Genetic Algorithm (Multi-objektiver genetischer Algorithmus)
NB	Naive-Bayes
OFDM	Orthogonal Frequency-Division Multiplexing (Orthogonales Frequenzmultiplexverfahren)
PCM	Puls Code Modulation
PDR	Packet Delivery Rate (Paketzustellrate)
PLR	Packet Loss Rate (Paketverlustrate)
PRNG	Pseudo Random Number Generator (Pseudozufallszahlen Generator)
QoS	Quality of Service ("Dienstgüte")
RAD	Random Assessment Delay (Zufällige Verzögerung zur Beurteilung)
ROC	Receiver Operating Characteristic Kurve (auch Grenzwertoptimierungskurve)
RS	Reed-Solomon [-Code]
RTS/CTS	Ready to Send - Clear to send (Sendebereitschaft - Sendefreigabe)
SPC	Single Parity Check (Einfache Paritätskontrolle)
TTL	Time to Live (Gültigkeitsdauer)
UDP	User Datagram Protocol
VANET	Vehicular Ad-hoc Network (Fahrzeug-ad-hoc-Netzwerk)

wlan Wireless Local Area Network (drahtloses Nahbereichsnetzwerk)

wsn Wireless Sensor Network

LITERATURVERZEICHNIS

- [ABC+12] Wahabou Abdou, Christelle Bloch, Damien Charlet, Dominique Dhoutaut und François Spies. "Designing Smart Adaptive Flooding in MANET Using Evolutionary Algorithm". English. In: *Mobile Wireless Middleware, Operating Systems, and Applications*. Hrsg. von Nalini Venkatasubramanian, Vladimir Getov und Stephan Steglich. Bd. 93. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2012, S. 71–84. DOI: 10.1007/978-3-642-30607-5_7.
- [AHB+11] Wahabou Abdou, Adrien Henriet, Christelle Bloch, Dominique Dhoutaut, Damien Charlet und François Spies. "Using an evolutionary algorithm to optimize the broadcasting methods in mobile ad hoc networks". In: *Journal of Network and Computer Applications* 34.6 (2011), S. 1794–1804.
- [ABVR09] Md Tanvir Al Amin, Sukarna Barua, Sudip Vhaduri und Ashikur Rahman. "Load aware broadcast in mobile ad hoc networks". In: *Communications, 2009. ICC'09. IEEE International Conference on*. IEEE, 2009, S. 1–5.
- [AIS+08] Ahmad Al Hanbali, Mouhamad Ibrahim, Vilmos Simon, Endre Varga und Iacopo Carreras. "A Survey of Message Diffusion Protocols in Mobile Ad Hoc Networks". In: *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools* (2008), S. 82–97. DOI: 10.4108/icst.valuetools2008.4510.
- [ABF+06] Enrique Alba, B. Dorronsoro, F. Luna, A.J. Nebro, P. Bouvry und L. Hogie. "A Cellular Multi-Objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs". In: *Computer Communications* 30.4. IEEE, 2006, S. 685–697.
- [Ariz] A. Ariza Quintana. *Inetmanet branch of inet-framework 2.0*. Abgerufen am 17.5.2016. URL: <https://github.com/aarizaq/inetmanet-2.0>.
- [BGL11] M. Bakhouya, J. Gaber und P. Lorenz. "An adaptive approach for information dissemination in Vehicular Ad hoc Networks". In: *Journal of Network and Computer Applications*

- 34.6 (2011), S. 1971–1978. DOI: doi:10.1016/j.jnca.2011.06.010.
- [Bakh13] Mohamed Bakhouya. “Broadcasting Approaches for Mobile Ad hoc Networks”. In: *High Performance Computing and Simulation (HPCS), 2013 International Conference on*. IEEE, 2013, S. 705–707.
- [BKSWo8] Boto Bako, Frank Kargl, Elmar Schoch und Michael Weber. “Advanced adaptive gossiping using 2-hop neighborhood information”. In: *Global Telecommunications Conference*. IEEE, 2008, S. 1–6.
- [BaWe11] Boto Bako und Michael Weber. “Efficient information dissemination in VANETs”. In: *Advances in Vehicular Networking Technologies* (Apr. 2011). Hrsg. von Dr. Miguel Almeida, S. 45–64.
- [BBA10] Muneer Bani Yassein, Mustafa Bani Khalaf und Ahmed Y. Al-Dubai. “A new probabilistic broadcasting scheme for mobile ad hoc on-demand distance vector (AODV) routed networks”. In: *The Journal of Supercomputing* 53.1 (März 2010), S. 196–211. DOI: 10.1007/s11227-010-0408-0.
- [BMS+] Zoltan Bojthe, Levente Meszaros, Benjamin Seregi, Rudolf Hornig und Andras Varga. *What Is INET Framework*. Abgerufen am 18.1.2016. URL: <https://inet.omnetpp.org/Introduction.html>.
- [BrSh10] Thomas Breuel und Faisal Shafait. “Automlp: Simple, effective, fully automated learning rate and size adjustment”. In: *The Learning Workshop*. Bd. 4. Utah. 2010, S. 51.
- [CFLZ05] Giacomo Cabri, Luca Ferrari, Letizia Leonardi und Franco Zambonelli. “The LAICA project: Supporting ambient intelligence via agents and ad-hoc middleware”. In: *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)*. IEEE, 2005, S. 39–44.
- [CMV+08] A. Cadiz, B. Mejias, J. Vallejos, K. Mens, P. Van Roy und W. De Meuter. “PALTA: Peer-to-peer Adaptable Topology for Ambient intelligence”. In: *Chilean Computer Science Society, 2008. SCCC '08. International Conference of the*. Nov. 2008, S. 100–109. DOI: 10.1109/SCCC.2008.16.

- [ČHE02] Mario Čagalj, Jean-Pierre Hubaux und ChristianENZ. "Minimum-energy Broadcast in All-wireless Networks: NP-completeness and Distribution Issues". In: *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*. MobiCom '02. Atlanta, Georgia, USA: ACM, 2002, S. 172–182. DOI: 10.1145/570645.570667.
- [Cola05] Michael D Colagrosso. "A Classification Approach to Broadcasting in a Mobile Ad Hoc Network". In: *Communications, IEEE International Conference on*. Bd. 2. IEEE, 2005, S. 1112–1117.
- [Cola07] Michael D. Colagrosso. "Intelligent Broadcasting in Mobile Ad Hoc Networks: Three Classes of Adaptive Protocols". In: *EURASIP Journal on Wireless Communications and Networking* 1 (2007), S. 1–16. DOI: 10.1155/2007/10216.
- [DaBh97] Bevan Das und Vaduvur Bharghavan. "Routing in ad-hoc networks using minimum connected dominating sets". In: *ICC'97 Montreal, Towards the Knowledge Millennium*. Bd. 1. IEEE, 1997, S. 376–380.
- [DPAM02] K. Deb, A. Pratap, S. Agarwal und T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Trans. Evol. Computat.* 6.2 (2002), S. 182–197. DOI: 10.1109/4235.996017.
- [DeAg94] Kalyanmoy Deb und Ram B Agrawal. "Simulated binary crossover for continuous search space". In: *Complex Systems* 9.3 (1994), S. 1–15.
- [DeCh10] Deepak Dembla und Yogesh Chaba. "Performance Modeling of Efficient and Dynamic Broadcasting Algorithm in MANETs Routing Protocols". In: *Computer Research and Development, 2010 Second International Conference on*. IEEE, 2010, S. 421–425.
- [DoPa96] P Domingo und M Pazzani. "Beyond independence: conditions for the optimality of the simple bayesian classifier". In: *Proc. 13th International Conference on Machine Learning*. 1996, S. 105–112.
- [DBS+01] Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten und Jean-Claude Burgelman. *Scenarios for ambient intelligence in 2010*. Office for official publications of the European Communities Luxembourg, 2001.

- [DuNe11] Juan J Durillo und Antonio J Nebro. "jMetal: A Java framework for multi-objective optimization". In: *Advances in Engineering Software* 42.10 (2011), S. 760–771. DOI: doi:10.1016/j.advengsoft.2011.05.014.
- [ESKSo6] Stephan Eichler, Christoph Schroth, Timo Kosch und Markus Strassberger. "Strategies for context-adaptive message dissemination in vehicular ad hoc networks". In: *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*. IEEE, 2006, S. 1–9.
- [Fawco4] Tom Fawcett. "ROC graphs: Notes and practical considerations for researchers". In: *Machine learning* 31.1 (2004), S. 1–38.
- [Förso7] Anna Förster. "Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey". In: *Intelligent Sensors, Sensor Networks and Information*. IEEE, 2007, S. 365–370.
- [FHTo9] Jerome Friedman, Trevor Hastie und Robert Tibshirani. *The elements of statistical learning*. 2. Aufl. Springer Series in Statistics. Berlin: Springer, Feb. 2009.
- [GGK+12] Jens Gebert, Andreas Georgakopoulos, Dimitrios Karvounas, Vera Stavroulaki und Panagiotis Demestichas. "Management of opportunistic networks through cognitive functionalities". In: *Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on*. IEEE, 2012, S. 113–118.
- [GSG12] N Gomathi, P Seethalakshmi und A Govardhan. "An integrated cross layer approach for multimedia streaming using multipath and multiple description coding in mobile ad-hoc networks". In: *WSEAS TRANSACTIONS on COMMUNICATIONS* (2012), S. 57–69.
- [Günt10] Yvonne Günter. *Hierarchisches TDMA-Verfahren mit Clusterbasiertem Medienzugriff zum Einsatz in Fahrzeug-Fahrzeug-Netzen*. Der Andere Verlag, 2010.
- [Hart13] Hannes Hartenstein. *Why is Vehicular (still) Challenging? Unresolved issues and future directions in WiVeC research*. Keynote IEEE WiVeC 2013. 2013.
- [Herro6] Klaus Herrmann. "Self-Organizing Infrastructures for Ambient Services". Diss. Technische Universität Berlin, 2006.

- [HOG13] Daniel Hiranandani, Katia Obraczka und JJ Garcia-Luna-Aceves. "MANET protocol simulations considered harmful: The case for benchmarking". In: *Wireless Communications, IEEE* 20.4 (2013), S. 82–90.
- [HSC95] Hugh W Holbrook, Sandeep K Singhal und David R Cheriton. "Log-based receiver-reliable multicast for distributed interactive simulation". In: *ACM SIGCOMM Computer Communication Review* 25.4 (1995), S. 328–341.
- [Hutc14] Frederick Hutchinson. "A Literature Review of Modern Ambient Intelligence in Smarthomes". University of Auckland, Abgerufen am 17.5.2016. 2014. URL: <https://www.cs.auckland.ac.nz/courses/compsci705s2c/assignments/reports/2014/Ambient%20Intel.pdf>.
- [HTT11] TP Huynh, YK Tan und KJ Tseng. "Energy-aware wireless sensor network with ambient intelligence for smart LED lighting system control". In: *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2011, S. 2923–2928.
- [IEEE03] IEEE - LAN/MAN Standards Committee of the IEEE Computer Society. *IEEE Standard for Information Technology-Telecommunications and information exchange between systems-LANs and MANs-Specific requirements-Part 11: WLAN MAC and PHY Specifications-Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*. IEEE, 2003.
- [ITU 03] ITU - International Telecommunication Union. *Recommendation G. 114 - One-way transmission time*. ITU - International Telecommunication Union, 2003.
- [JSAC01] Christine E Jones, Krishna M Sivalingam, Prathima Agrawal und Jyh Cheng Chen. "A survey of energy efficient network protocols for wireless networks". In: *wireless networks* 7.4 (2001), S. 343–358.
- [Khelo7] Abdelmajid Khelil. "A generalized broadcasting technique for mobile ad hoc networks". Diss. Universität Stuttgart, 2007.
- [KBME13] Lyes Khoukhi, Hakim Badis, Leila Merghem-Boulahia und Moez Essegghir. "Admission control in wireless ad hoc networks: a survey". English. In: *EURASIP Journal on Wireless Communications and Networking* 1 (2013), S. 1–13. DOI: 10.1186/1687-1499-2013-109.

- [KOA+99] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, Thad Starner und Wendy Newstetter. "The Aware Home: A Living Laboratory for Ubiquitous Computing Research". In: *Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*. CoBuild '99. London, UK: Springer, 1999, S. 191–198.
- [KoSi12] Andras Kokuti und Vilmos Simon. "Adaptive multihop broadcast protocols for ad hoc networks". In: *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*. IEEE, 2012, S. 1–6.
- [KHW09] D. Koutsonikolas, Y.C. Hu und Chih-Chun Wang. "Pacifier: High-Throughput, Reliable Multicast without "Crying Babies" in Wireless Mesh Networks". In: *INFOCOM 2009, IEEE*. Apr. 2009, S. 2473–2481. DOI: 10.1109/INFCOM.2009.5062175.
- [KCC05] Stuart Kurkowski, Tracy Camp und Michael Colagrosso. "MANET simulation studies: the incredibles". In: *ACM SIGMOBILE Mobile Computing and Communications Review* 9.4 (2005), S. 50–61.
- [KCG06] Pradeep Kyasanur, Romit Roy Choudhury und Indranil Gupta. "Smart gossip: An adaptive gossip-based broadcasting service for sensor networks". In: *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*. IEEE, 2006, S. 91–100.
- [LSR03] J Lee, S Singh und Y Roh. "Interlayer interactions and performance in wireless ad hoc network". In: *IRTF ANS Working Group, Internet-Draft* (2003).
- [LSS07] Jin-Shyan Lee, Yu-Wei Su und Chung-Chou Shen. "A comparative study of wireless protocols: Bluetooth, UWB, Zig-Bee, and Wi-Fi". In: *Industrial Electronics Society, IECON*. IEEE, 2007, S. 46–51.
- [LSG02] Sung-Ju Lee, William Su und Mario Gerla. "On-demand multicast routing protocol in multihop wireless mobile networks". In: *Mobile networks and applications* 7.6 (2002), S. 441–453.
- [LXHS12] Liqun Li, Guoliang Xing, Qi Han und Limin Sun. "ASM: Adaptive Voice Stream Multicast over Low-Power Wireless Networks". In: *Parallel and Distributed Systems, IEEE Transactions on* 23.4 (2012), S. 626–633.

- [LYC+12] Y. Li, K. Ying, P. Cheng, H. Yu und H. Luo. "Cooperative data dissemination in cellular-VANET heterogeneous wireless networks". In: *High Speed Intelligent Communication Forum (HSIC), 2012 4th International*. Mai 2012, S. 1–4. DOI: 10.1109/HSIC.2012.6212991.
- [LSK09] Dimitrios Liarokapis, Ali Shahrabi und Andreas Komninos. "DibA: An Adaptive Broadcasting Scheme in Mobile Ad Hoc Networks". In: *2009 Seventh Annual Communication Networks and Services Research Conference*. Institute of Electrical und Electronics Engineers, Mai 2009, S. 224–231. DOI: 10.1109/CNSR.2009.42.
- [LQZ13] Federico Librino, Giorgio Quer und Michele Zorzi. "Network-aware retransmission strategy selection in ad hoc wireless networks". In: *Communications Workshops (ICC)*. IEEE, 2013, S. 358–363.
- [LiKioo] Hyojun Lim und Chongkwon Kim. "Multicast tree construction and flooding in wireless ad hoc networks". In: *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. ACM, 2000, S. 61–68.
- [LiLi12] Adam Lipowski und Dorota Lipowska. "Roulette-wheel selection via stochastic acceptance". In: *Physica A: Statistical Mechanics and its Applications* 391.6 (2012), S. 2193–2196.
- [MRL+11] Geneviève Mange, Christophe Rosik, Stéphanie Leveil, Ulrico Celentano, Olasunkanmi Durowoju und Kamran Arshad. "Cognitive resource management for QoS support in mobile opportunistic communications". In: *Future Network & Mobile Summit (FutureNetw)*. IEEE, 2011, S. 1–9.
- [MaAno9] Kenneth C Mansfield Jr und James L Antonakos. *Computer Networking for LANs to WANs: Hardware, Software and Security*. Cengage Learning, 2009.
- [MaNi98] Makoto Matsumoto und Takuji Nishimura. "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), S. 3–30.
- [MLGo3] Prasant Mohapatra, Jian Li und Chao Gui. "QoS in mobile ad hoc networks". In: *IEEE Wireless Communications* 10.3 (2003), S. 44–53.

- [Open] OpenSim Ltd., Hrsg. *OMNeT++ Discrete Event Simulator - Introduction*. Abgerufen am 17.5.2016. URL: <https://omnetpp.org/intro>.
- [PSCo7] Carlo Parata, Vincenzo Scarpa und Gabriella Convertino. "Flex-WiFi: a mixed infrastructure and ad-hoc IEEE 802.11 network for data traffic in a home environment". In: *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. IEEE, 2007, S. 1–6.
- [PeLuo0] Wei Peng und Xi-Cheng Lu. "On the reduction of broadcast redundancy in mobile ad hoc networks". In: *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. IEEE, 2000, S. 129–130.
- [Piep98] Roel Pieper. "From Devices to "Ambient Intelligence": The Transformation of Consumer Electronics". In: *Digital Living Room Conference* (1998).
- [Rapp+02] Theodore S Rappaport u. a. *Wireless communications: principles and practice*. 2. Aufl. Prentice Hall PTR New Jersey, 2002.
- [ANY13] Hasan AA Al-Rawi, Ming Ann Ng und Kok-Lim Alvin Yau. "Application of reinforcement learning to routing in distributed wireless networks: A review". In: *Artificial Intelligence Review* 43.3 (2013), S. 381–416.
- [ReSo60] Irving S Reed und Gustave Solomon. "Polynomial codes over certain finite fields". In: *Journal of the society for industrial and applied mathematics* 8.2 (1960), S. 300–304.
- [RTL+13] D.G. Reina, S.L. Toral, J.M. Leon-Coca, F. Barrero, N. Besis und E. Asimakopoulou. "An Evolutionary Computational Approach for Optimizing Broadcasting in Disaster Response Scenarios". In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. Juli 2013, S. 94–100. DOI: 10.1109/IMIS.2013.25.
- [RMRo7] Celso C. Ribeiro, Simone L. Martins und Isabel Rosseti. "Metaheuristics for optimization problems in computer communications". In: *Computer Communications* 30.4 (2007), S. 656–669. DOI: 10.1016/j.comcom.2006.08.027.
- [RRS12] F. J. Ros, P. M. Ruiz und I. Stojmenovic. "Acknowledgment-Based Broadcast Protocol for Reliable and Efficient Data Dissemination in Vehicular Ad Hoc Networks". In: *IEEE Transactions on Mobile Computing* 11.1 (2012), S. 33–46. DOI: 10.1109/TMC.2010.253.

- [RoTo99] Elizabeth M Royer und Chai-Keong Toh. "A review of current routing protocols for ad hoc mobile wireless networks". In: *IEEE personal communications* 6.2 (1999), S. 46–55.
- [Rub98] Dan Rubenstein. "Real-time reliable multicast using proactive forward error correction". In: *Computer Science Department Faculty Publication Series - University of Massachusetts at Amherst* (1998).
- [SGK05] Norman M Sadeh, Fabien L Gandon und Oh Byung Kwon. "Ambient intelligence: The mycampus experience". Carnegie Mellon University Pittsburgh - School Of Computer Science, Abgerufen am 17.5.2016. 2005. URL: <http://reports-archive.adm.cs.cmu.edu/anon/isri2005/CMU-ISRI-05-123.pdf>.
- [SBDK10] Elmar Schoch, Boto Bako, Stefan Dietzel und Frank Kargl. "Dependable and secure geocast in vehicular networks". In: *Proceedings of the seventh ACM international workshop on Vehicular Internetworking*. ACM. 2010, S. 61–68.
- [SDP+12] Pablo Serrano, Antonio De La Oliva, Paul Patras, Vincenzo Mancuso und Albert Banchs. "Greening wireless communications: Status and future directions". In: *Computer Communications* 35.14 (2012), S. 1651–1661.
- [SlMa10] M. Slavik und I. Mahgoub. "Stochastic Broadcast for VANET". In: *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. Jan. 2010, S. 1–5. DOI: 10.1109/CCNC.2010.5421816.
- [SlMa11] Michael Slavik und Imad Mahgoub. "Applying machine learning to the design of multi-hop broadcast protocols for VANET". In: *Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2011, S. 1742–1747.
- [TLZ15] Fei Tao, Yuanjun Laili und Lin Zhang. "Brief History and Overview of Intelligent Optimization Algorithms". In: *Configurable Intelligent Optimization Algorithm*. Springer, 2015, S. 3–33.
- [TaGu13] Supriya Tayal und Vinti Gupta. "A Survey of Attacks on Manet Routing Protocols". In: *International Journal of Innovative Research in Science, Engineering and Technology* 2.6 (2013), S. 2280–2285.
- [Thom07] Captain Ryan W. Thomas. "Cognitive Networks". Diss. Virginia Polytechnic Institute und State University, 2007.

- [TCC+15] Alvaro Torres, Carlos T Calafate, Juan-Carlos Cano, Pietro Manzoni und Yusheng Ji. "Evaluation of flooding schemes for real-time video transmission in VANETs". In: *Ad Hoc Networks* 24 (2015), S. 3–20.
- [TNCS02] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen und Jang-Ping Sheu. "The Broadcast Storm Problem in a Mobile Ad Hoc Network". English. In: *Wireless Networks* 8.2-3 (2002), S. 153–167. DOI: 10.1023/A:1013763825347.
- [VaWi09] Hado Van Hasselt und Marco A Wiering. "Using continuous action spaces to solve discrete problems". In: *Neural Networks IJCNN*. IEEE, 2009, S. 1149–1156.
- [Varg01] András et al. Varga. "The OMNeT++ discrete event simulation system". In: *Proceedings of the European simulation multiconference (ESM-2001)*. Bd. 9. S 185. 2001, S. 65.
- [VSW11] Endre Sándor Varga, Vilmos Simon und Bernát Wiandt. "On-line genetic programming of multi-hop broadcast protocols in ad hoc networks". In: *Periodica Polytechnica Electrical Engineering* 55.1-2 (2011), S. 65. DOI: 10.3311/pp.ee.2011-1-2.07.
- [VMK12] B Vinodhini, M Marikkannan und S Karthik. "A Review Of Various Issues In Routing And Broadcasting Methods For Designing Efficient Method In Mobile Adhoc Network". In: *International Journal of Computer Networking* 2.3 (2012).
- [VuMa12] Vladimir Vukadinovic und Stefan Mangold. "Performance of Collaborative GPS Localization in Pedestrian Ad Hoc Networks". In: *Proceedings of the third ACM international workshop on Mobile Opportunistic Networks*. ACM. 2012, S. 45–52.
- [West79] D. H. D. West. "Updating mean and variance estimates: An improved method". In: *Communications of the ACM* 22.9 (1979), S. 532–535.
- [WSV12] Bernát Wiandt, Vilmos Simon und Endre Sándor Varga. "Efficient Multihop Broadcast with Distributed Protocol Evolution". In: *Information and Communication Technologies*. Springer. 2012, S. 309–320.
- [Wieg12] Bernhard Wiegel. "Quality of Service in Fahrzeug-Fahrzeug-Netzen – dezentrale und schichtübergreifende Steuerung des Nachrichtenaufkommens". Diss. Universität Ulm, 2012.

- [WNEo0] Jeffrey E Wieselthier, Gam D Nguyen und Anthony Ephremides. "On the construction of energy-efficient broadcast and multicast trees in wireless networks". In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Bd. 2. IEEE, 2000, S. 585–594.
- [WNEo2] Jeffrey E Wieselthier, Gam D Nguyen und Anthony Ephremides. "Energy-efficient broadcast and multicast trees in wireless networks". In: *Mobile networks and applications* 7.6 (2002), S. 481–492.
- [WiCao2] Brad Williams und Tracy Camp. "Comparison of broadcasting techniques for mobile ad hoc networks". In: *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM. 2002, S. 194–205.
- [XWZ10] Zhiming Xu, Yu Wang und Jingguo Zhu. "A reliable multicast routing protocol for high-speed mobile ad hoc networks: R-ODMRP". In: *Journal of software* 5.1 (2010), S. 20–27.
- [YMG08] Jennifer Yick, Biswanath Mukherjee und Dipak Ghosal. "Wireless Sensor Network Survey". In: *Comput. Netw.* 52.12 (2008), S. 2292–2330. DOI: 10.1016/j.comnet.2008.04.002.
- [YLN03] J. Yoon, M. Liu und B. Noble. "Random waypoint considered harmful". In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*. Bd. 2. März 2003, S. 1312–1321. DOI: 10.1109/INFCOM.2003.1208967.
- [ZhZho8] Qian Zhang und Ya-Qin Zhang. "Cross-Layer Design for QoS Support in Multihop Wireless Networks". In: *Proceedings of the IEEE* 96.1 (Jan. 2008), S. 64–76. DOI: 10.1109/JPROC.2007.909930.
- [ZKLY10] Xiaoying Zhang, Thomas Kunz, Li Li und Oliver Yang. "An energy-efficient broadcast protocol in manets: Design and evaluation". In: *Communication Networks and Services Research Conference (CNSR), 2010 Eighth Annual*. IEEE, 2010, S. 199–206.