



Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Verteilte Systeme

Masterarbeit

Effizientes Mehrkanal-Routing

Ronny Scholz
19. Dezember 2008

Kurzbeschreibung

Diese Arbeit beschäftigt sich mit der Entwicklung zweier Mehrkanal-Routingverfahren für drahtlose Mesh-Netzwerke. Im Fokus der Arbeit steht die Maximierung des Durchsatzes, ohne dabei die Grundlagen der Zuverlässigkeit einzuschränken. Entscheidend für die Erhöhung des Durchsatzes ist die Ermittlung eines Pfades, der die einzelnen Abschnitte eines Multi-Hop-Datenstroms möglichst gleichmäßig auf die verfügbaren Funkkanäle aufteilt. Da die Bestimmung eines derart optimalen Pfades zu aufwändig ist, gehen beide Verfahren verschiedene Kompromisse aus Berechnungsdauer und Optimalität der Pfadsuche ein.

Durch eine Evaluierung stellte sich heraus, dass die Anwendung eines heuristischen Verfahrens den effizienteren Kompromiss darstellt. Dieses Verfahren führt die Pfadsuche auf einem zuvor eingeschränkten Suchraum durch und erreicht auf zwei Funkkanälen eine Durchsatzsteigerung von bis zu 95% gegenüber der Nutzung eines einzigen Funkkanals. Weiterhin wurde nachgewiesen, dass die Robustheit des Netzwerks gegen Störungen, wie etwa Kanalausfälle, erhöht werden konnte.

Inhaltsverzeichnis

Kurzbeschreibung	iv
Abkürzungsverzeichnis	ix
Abbildungsverzeichnis	xii
Tabellenverzeichnis	xiii
Listings	xv
1 Einleitung	1
1.1 Motivation	1
1.2 Problemexposition	2
1.3 Aufgabenstellung	2
1.4 Ergebnisse	2
1.5 Gliederung	3
2 Grundlagen und verwandte Arbeiten	5
2.1 Einordnung und Abgrenzung	5
2.2 WLAN	6
2.2.1 Funkkanäle	6
2.2.2 Interferenzen	8
2.3 Routing in drahtlosen Mesh-Netzwerken	9
2.3.1 Linkmetriken für drahtlose Mesh-Netzwerke	10
2.3.2 Spezielle Effekte in Multi-Hop-Netzwerken	12
2.3.3 Ad-hoc Wireless Distribution Service	13
2.4 Mehrkanal-Routingverfahren	14
2.4.1 Klassifizierung	15
2.4.2 Ausgewählte Mehrkanal-Routingverfahren	18
3 Konzept	23
3.1 Vorbetrachtungen	23
3.1.1 Maßnahmen zur Ermöglichung zuverlässiger Kommunikation	23
3.1.2 Verarbeitung von Paketen	25
3.1.3 Teilnehmer mit zu wenig Netzwerkkarten	28
3.2 Diagonale Links	29
3.2.1 Topologie	29

3.2.2	Aufwandsabschätzung	33
3.2.3	Gruppierung von Knoten	34
3.2.4	Optimalität des Pfades	34
3.3	Heuristik	35
3.3.1	Vorauswahl	35
3.3.2	Brute-Force-Schritt	36
3.3.3	Aufwandsabschätzung	39
3.3.4	Optimalität des Pfades	41
4	Implementierung	43
4.1	Einführung in relevante Teile von AWDS	43
4.1.1	Laufzeitumgebung	43
4.1.2	Module	44
4.1.3	Das awdsrouting-Modul	45
4.2	Allgemeine Anpassungen von AWDS	48
4.2.1	Modulstruktur	48
4.2.2	Nutzung der ETX-Metrik	49
4.2.3	Paketversand	49
4.3	Diagonale Links	50
4.3.1	Erweiterung der Topologie	50
4.3.2	Pfadsuche	53
4.3.3	Topologie-Export	54
4.4	Heuristik	56
4.4.1	Verwaltung der Kanalkosten	56
4.4.2	Anpassung der Datenpakete	58
4.4.3	Paketversand	59
4.4.4	Verwaltung der Routingtabelle	59
4.4.5	Pfadsuche	60
4.4.6	Erweiterungen	63
4.5	Zusammenfassung	66
5	Evaluierung	67
5.1	Hardware	67
5.2	Steigerung des Durchsatzes	67
5.2.1	Versuchsaufbau	68
5.2.2	Erwartungen	69
5.2.3	Auswertung	70
5.3	Dauer der Pfadsuche	73
5.3.1	Versuchsaufbau	73
5.3.2	Erwartungen	73
5.3.3	Auswertung	74
5.4	Einfluss der Wichtung zwischen Latenz und Durchsatz	75
5.4.1	Versuchsaufbau	75
5.4.2	Erwartungen	76

5.4.3	Auswertung	76
5.5	Verhalten im Fehlerfall	78
5.5.1	Versuchsaufbau	78
5.5.2	Erwartungen	78
5.5.3	Auswertung	79
5.6	Diskussion der Ergebnisse	80
6	Zusammenfassung und Ausblick	81
	Literaturverzeichnis	85

Inhaltsverzeichnis

Abkürzungsverzeichnis

ALIX	AMD LX based system boards
AMD	Advanced Micro Devices
API	Application Programming Interface
AWDS	Ad-hoc Wireless Distribution Service
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DHCP	Dynamic Host Control Protocol
ETT	Estimated Transmission Time
ETX	Expected Transmission Count
GEA	Generic Event-based API
HWMP	Hybrid Wireless Mesh Routing Protocol
IEEE	Institute of Electrical and Electronics Engineers
IPv6	Internet Protocol Version 6
ISM	Industrial, Scientific, Medical
LAN	Local Area Network
LIBRA	Load and Interference Balanced Routing Algorithm
MAC	Media Access Control
MANET	Mobile Ad-hoc Network
MCR	Multichannel Routing Protocol
MIC	Metric of Interference and Channel Switching
MR-LQSR	Multi-Radio Link-Quality Source-Routing
POSIX	Portable Operating System Interface
SSCH	Slotted Seeded Channel Hopping
STL	Standard Template Library
TTL	Time To Live
WCETT	Weighted Cumulative ETT
WLAN	Wireless LAN
WRAP	Wireless Router Application Platform
XML	Extensible Markup Language

Inhaltsverzeichnis

Abbildungsverzeichnis

2.1	Klassifizierung drahtloser Netzwerke	5
2.2	Beispiel für drei überlappungsfreie WLAN-Funkkanäle im ISM-Band.	7
2.3	Übertragungsbereich (durchgehender Kreis) und Interferenzbereich (gestrichelter Kreis) eines Teilnehmers.	9
2.4	Multi-Hop-Übertragung von Teilnehmer A zu Teilnehmer E.	12
2.5	Vermeidung konkurrierender Medienzugriffe zwischen Teilnehmern unterschiedlicher Datenströme durch Routing über dünn besetzte Bereiche.	13
2.6	Vermeidung konkurrierender Medienzugriffe der Teilnehmer entlang eines Pfades durch die Nutzung verschiedener Funkkanäle. K gibt den Funkkanal einer Verbindung an.	14
2.7	Klassifizierung von Mehrkanal-Routingverfahren anhand der Kanalnutzung.	16
2.8	Topologie bei identischer Kanaluordnung (links) und individueller Kanaluordnung (rechts) bei Teilnehmern mit zwei Netzwerkkarten. K gibt den Funkkanal einer Verbindung an.	17
3.1	Sub-Topologien in einem Netzwerk mit zwei Funkkanälen.	30
3.2	Topologie mit vertikalen Links (gestrichelte Pfeile)	31
3.3	Topologie mit diagonalen Links (gestrichelte Pfeile)	32
3.4	Möchte Teilnehmer A Daten zu C senden, reduziert er zunächst das Netzwerk auf einen einzelnen Teilnehmer-Pfad zum Ziel. Dieser Schritt dient als Vorstufe für die eigentliche Pfadsuche.	36
3.5	Im Brute-Force-Schritt wird auf dem vorselektierten Graphen der optimale Pfad bestimmt (links). Dazu werden alle Kanalkombinationen in einen Entscheidungsbaum eingetragen und miteinander verglichen (rechts).	37
3.6	Die Kosten des bereits durchlaufenen Pfades werden im Paket-Header gespeichert und vom Empfänger ausgewertet (schwarze Pfeile). Sie dienen zur Initialisierung des Entscheidungsbaumes.	39
4.1	Systemarchitektur mit GEA.	44
4.2	Vereinfachte Darstellung der Modulstruktur von AWDS.	45
4.3	Logische Komponenten des awdsrouting-Moduls im Zusammenspiel mit dem tapiface- und rawbasic-Modul.	45
4.4	Vereinfachte Darstellung der lokalen Sicht von Teilnehmer A auf die Topologie.	47

Abbildungsverzeichnis

4.5	Nach der Umstrukturierung der AWDS-Module kann das awdsrouting-Modul beliebig viele rawbasic-Module verwalten.	48
4.6	Ausschnitt der lokalen Sicht von Teilnehmer A auf eine Mehrkanal-Topologie mit diagonalen Links.	51
4.7	Die Klasse GroupCache.	52
4.8	Bei der Erzeugung diagonalen Links bildet der Zielknoten des realen Links den Endpunkt, die unbeteiligten Knoten des Senders bilden die Startpunkte.	53
4.9	Konstruktion der Kanten-Endpunkte zwischen zwei Teilnehmern (links) und Anzeige im TopoViewer (rechts).	55
4.10	Die Klasse VectorWeight.	57
4.11	Reduzierter Suchraum für die Pfadsuche von Teilnehmer A zu Teilnehmer C.	61
4.12	Suche nach der optimalen Kanalkombination.	62
4.13	Liste der Teilnehmer-Pfade von Teilnehmer A zu Teilnehmer D.	64
5.1	Topologie während der Durchsatzmessungen.	68
5.2	Vergleich der Nutzung der Verbindungskapazität durch verschiedene Routingverfahren.	70
5.3	Dauer der Pfadsuche.	75
5.4	Schematische Darstellung der Latenzmessung.	76
5.5	Einfluss von β	77
5.6	Auswirkung einer Kanalstörung auf eine Datenübertragung im effizienten Modus (links) und im zuverlässigen Modus (rechts).	79

Tabellenverzeichnis

4.1	Genereller Aufbau eines Topologie-Pakets. Anpassungen zur Speicherung mehrerer Nachbarschaften sind hervorgehoben.	52
4.2	Struktur einer Nachbarschaft innerhalb eines Topologie-Pakets.	52
4.3	Nächste Hops von Teilnehmer A.	54
4.4	Header eines Datenpakets. Ergänzungen zur Speicherung des Paketverlaufs sind hervorgehoben.	59
5.1	Start- und Ziel-Teilnehmer für verschiedene Pfadlängen.	69
5.2	Ermittelte optimale Pfade mit einer Länge von einem Hop.	71
5.3	Ermittelte optimale Pfade mit einer Länge von zwei Hops.	71
5.4	Ermittelte optimale Pfade mit einer Länge von drei Hops.	72
5.5	Ermittelte optimale Pfade mit einer Länge von vier Hops.	72
5.6	Ermittelte optimale Pfade für verschiedene Werte von β	77

Tabellenverzeichnis

Listings

4.1	Die Methode <code>VectorWeight::getValue()</code>	58
4.2	Pseudo-Quelltext der rekursiven Pfadsuche.	65

Listings

1 Einleitung

1.1 Motivation

Drahtlose Mesh-Netzwerke ermöglichen die Kommunikation in Gebieten, die aufgrund ihrer geografischen Lage oder bestehender Hindernisse nicht mit konventionellen Kabelnetzwerken versorgt werden können. Die Kommunikation zwischen den Teilnehmern erfolgt hierbei komplett über Funk, beispielsweise durch WLAN nach IEEE 802.11. Größere Entfernungen werden von Zwischenknoten überbrückt, welche die Daten aufbereiten und neu versenden. Gewöhnlich sind die Zwischenknoten stationär. Sie spannen ein drahtloses Kern-Netz auf, das von mobilen Endgeräten genutzt wird, um sich untereinander auszutauschen.

Weil gewisse Freiheiten in der Mobilität der Endgeräte bestehen, eignen sich drahtlose Mesh-Netzwerke auch für den industriellen Einsatz, beispielsweise zur Ausleuchtung von Fertigungsstraßen oder für die Verwendung mobiler Bediengeräte. Die stationäre Eigenschaft des Kern-Netzes erleichtert darüber hinaus die Anwendung von Verfahren, die sich mit der Bereitstellung gewisser Dienstegüthen beschäftigen, etwa der Garantie eines bestimmten Durchsatzes oder der Zusicherung von Echtzeitfähigkeit. Ein wichtiges Kriterium der Echtzeitfähigkeit ist die Eigenschaft der Zuverlässigkeit, welche die Resistenz des Netzwerks gegen Ausfälle von Komponenten beschreibt.

Durch die allseitige Ausbreitung von Funksignalen beeinflussen sich alle Teilnehmer in unmittelbarer Umgebung gegenseitig. Bei Multi-Hop-Übertragungen hat dies zur Folge, dass stets mehrere Teilnehmer um den Zugriff auf das Medium konkurrieren. Weil jedoch immer nur ein Teilnehmer gleichzeitig senden darf, können lange Wartezeiten auf den Medienzugriff entstehen, die eine Verringerung des maximal erreichbaren Durchsatzes, d.h. der Kapazität der Verbindung, zur Folge haben.

Des Weiteren sind drahtlose Netzwerke sehr anfällig gegen externe Störsender, die das Medium dauerhaft blockieren können. So können räumlich kompakte Einkanal-Netzwerke bereits durch einen einzelnen Störer komplett zum Erliegen gebracht werden.

Durch die Nutzung mehrerer Funkkanäle kann die Kapazität der Verbindungen gesteigert werden. Denn Teilnehmer, die auf verschiedenen Kanälen senden, konkurrieren nicht mehr um den Medienzugriff, so dass sie häufiger Pakete verschicken können. Weiterhin lassen sich bestehende Verbindungen redundant auslegen, wodurch die Zuverlässigkeit des Netzwerks erhöht werden kann.

1.2 Problemexposition

Die effiziente Nutzung der zusätzlichen Kapazität in Mehrkanal-Netzwerken gelingt durch die Vermeidung konkurrierender Medienzugriffe benachbarter Teilnehmer, indem diese unterschiedliche Funkkanäle benutzen. Dies lässt sich nur mithilfe eines optimierten Routingverfahrens umsetzen, das die einzelnen Abschnitte eines Pfades verschiedenen Funkkanälen zuordnet. Denn unterhalb der Routingschicht haben Teilnehmer keine globale Sicht auf die Funkkanäle, so dass die Kanalnutzung nicht optimal erfolgen kann. Oberhalb der Mesh-Routingschicht ist das Ausreizen der Kanäle ebenfalls nicht sinnvoll, weil ihre Nutzung von der Routingschicht abstrahiert wird.

Die simple Nutzung bestehender Einkanal-Routingverfahren ist nicht zweckmäßig, weil diese keine gleichmäßige Auslastung des Funkmediums anstreben. Sie sind dadurch nicht in der Lage, in Mehrkanal-Topologien den optimalen Pfad zu finden. Um die maximale Durchsatzsteigerung zu erzielen, ist der Einsatz spezieller Mehrkanal-Routingverfahren erforderlich, welche neben der Minimierung der Pfadlänge auch die Verteilung der Übertragungen auf die verfügbaren Funkkanäle berücksichtigen. Dann ist die Pfadsuche allerdings nicht mehr trivial lösbar. So kann gezeigt werden, dass der optimale Pfad nur durch einen Vergleich aller möglichen Pfade ermittelt werden kann. Da die Anzahl der Pfade mit wachsender Topologie exponentiell steigt, gilt die optimale Pfadsuche in Mehrkanal-Netzwerken als unpraktikabel.

1.3 Aufgabenstellung

Das Ziel dieser Masterarbeit ist die Entwicklung von Mehrkanal-Routingverfahren für drahtlose Mesh-Netzwerke, welche eine maximale Durchsatzsteigerung erreichen, ohne die Zuverlässigkeit des Netzwerks zu beeinträchtigen.

Um einen praktikablen Einsatz der Pfadsuche zu ermöglichen, sollen hierfür zwei verschiedene Kompromisse aus Berechnungsdauer und Optimalität eingegangen und bewertet werden.

Eines der Verfahren soll durch eine geeignete Anpassung der Topologie Kanalwechsel bei der Pfadsuche begünstigen. In einem zweiten Verfahren hat die Pfadsuche durch die Anwendung einer Heuristik zu erfolgen, die den Suchraum bereits im Vorfeld einschränkt.

Die Implementierung der Protokolle soll als Erweiterung des bestehenden Routingverfahrens AWDS erfolgen und mit geeigneten Mitteln evaluiert werden.

1.4 Ergebnisse

Im Rahmen dieser Masterarbeit wurden zwei Mehrkanal-Routingprotokolle entwickelt, deren korrekte Funktionsweise durch eine Evaluierung in einer realen Testumgebung

nachgewiesen wurde. Mit beiden Routingverfahren wurde eine deutliche Steigerung des Durchsatzes gegenüber Einkanal-Routing erreicht. Als effizienter stellte sich das heuristische Verfahren heraus, das den Suchraum im Vorfeld auf eine kleine Menge von Alternativen einschränkt. Die Heuristik benötigt zwar etwas mehr Zeit für die Pfadsuche, als das Verfahren der diagonalen Links, das die Pfadsuche mittels eines Shortest-Path-Verfahrens auf einer angepassten Topologie durchführt. Sie kann jedoch eine stärkere Optimierung des Pfades erreichen und dadurch einen höheren Durchsatz erzielen.

Die Heuristik bietet zudem die Möglichkeit, die Präferenz nach hohem Durchsatz oder geringer Latenz in die Pfadsuche einzubeziehen. Die korrekte Funktionsweise dieser Option konnte ebenfalls bestätigt werden.

Beide Verfahren wurden darauf konzipiert, auf einer Topologie mit starker Redundanz der Verbindungen eingesetzt zu werden. Diese Redundanz soll dazu genutzt werden, um im Fehlerfall schnell auf alternative Pfade zurückgreifen zu können. Dazu wurde eines der Verfahren auf das Verhalten im Fehlerfall analysiert. Es wurde gezeigt, dass bei der Störung eines kompletten Funkkanals die Kommunikation zunächst zusammenbricht, nach einer Aktualisierung der Pfade jedoch fortgeführt werden kann. Die Fehlererkennung dauert jedoch noch deutlich zu lang. Es konnte gezeigt werden, dass durch redundanten Paketversand die störungsbedingten Ausfälle einer Datenübertragung vermieden werden können.

1.5 Gliederung

Die vorliegende Arbeit gliedert sich in sechs Kapitel.

Kapitel 2 gibt einen Einblick in Routingverfahren für drahtlose Mesh-Netzwerke. Es erläutert die speziellen Anforderungen, die an Mehrkanal-Routingverfahren gestellt werden und stellt verwandte Arbeiten vor.

Im dritten Kapitel wird erläutert, auf welche Weise Voraussetzungen für zuverlässige Kommunikation geschaffen werden sollen. Davon ausgehend werden zwei Routingverfahren für Mehrkanal-Netzwerke entwickelt, die verschiedene Ansätze zur effizienten Pfadsuche nutzen.

Kapitel 4 beschreibt die Implementierung der entwickelten Protokolle in das bestehende Routingverfahren AWDS.

In Kapitel 5 erfolgt die Evaluierung der Routingverfahren in einer realen Testumgebung. Das sechste Kapitel enthält eine Zusammenfassung der Arbeit sowie einen Ausblick.

1 Einleitung

2 Grundlagen und verwandte Arbeiten

Dieses Kapitel bietet einen Einstieg in die Thematik von Mehrkanal-Routingverfahren. Es beginnt mit einer Klassifizierung von Funk-Netzwerken, beschreibt die Fokussierung der Arbeit auf drahtlose Mesh-Netzwerke und führt in die relevanten WLAN-Standards ein. Anschließend werden Einkanal-Routingverfahren für drahtlose Mesh-Netzwerke vorgestellt und Interferenzen als deren maßgebliche Ursache für eine eingeschränkte Leistungsfähigkeit beschrieben. Der letzte Abschnitt erläutert die Schwierigkeiten der optimalen Pfadsuche in Mehrkanal-Topologien und stellt dafür geeignete Routingprotokolle vor.

2.1 Einordnung und Abgrenzung

Diese Arbeit beschäftigt sich mit der Entwicklung von Routingverfahren für drahtlose Mesh-Netzwerke, die auf dem WLAN-Funkstandard nach IEEE 802.11 basieren. Diese sind ein Spezialfall mobiler Ad-hoc-Netzwerke. Abbildung 2.1 zeigt eine grobe Klassifizierung drahtloser Netzwerke, anhand derer eine Abgrenzung gegenüber anderen Netzen erfolgen soll.

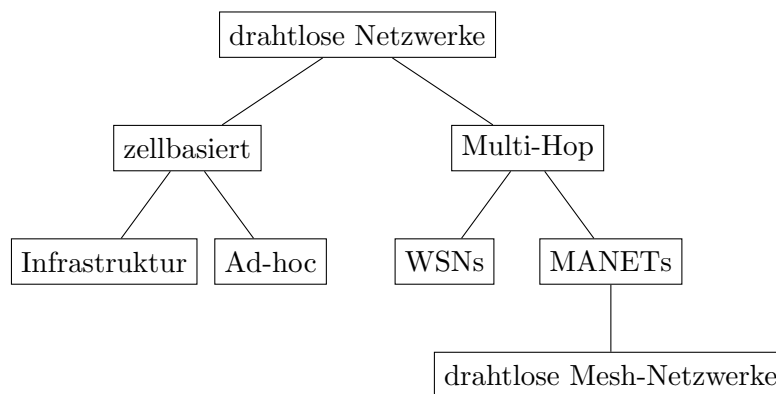


Abbildung 2.1: Klassifizierung drahtloser Netzwerke

Zu den *zellbasierten Netzwerken* zählen die in IEEE 802.11 standardisierten Betriebsmodi des Infrastruktur-Modus und des Ad-hoc-Modus. Beiden Varianten gemein ist die begrenzte Reichweite des Netzwerks, weil sich alle Teilnehmer in unmittelbarer Empfangsreichweite zueinander befinden müssen.

Multi-Hop-Netzwerke erlauben dagegen eine größere räumliche Ausdehnung des Netzwerks. Können sich zwei Teilnehmer nicht direkt erreichen, übernehmen andere Teilnehmer die Rolle eines Vermittlers und leiten die Daten entsprechend weiter.

Drahtlose Sensor-Netzwerke (Wireless Sensor Networks, WSNs) dienen der großflächigen Erfassung von Umgebungsparametern mit einer Vielzahl von Sensoren. Diese sind in der Regel stationär, so dass eine Veränderung der Topologie nur selten stattfindet. Die Teilnehmer besitzen meist eine autonome Energieversorgung, weshalb die Berücksichtigung der Energieeffizienz eine bedeutende Rolle spielt.

Mobile Ad-hoc Networks (MANETs) bestehen aus mobilen Teilnehmern, die untereinander kommunizieren. Die Mobilität der Teilnehmer bewirkt häufige Veränderungen der Topologie, was den Ausfall bestehender Pfade zur Folge hat. Eine der wesentlichen Anforderungen an Routingverfahren ist daher die frühzeitige Erkennung von Verbindungsabbrüchen, um schnell auf alternative Pfade ausweichen zu können.

Drahtlose Mesh-Netzwerke sind ein Spezialfall von MANETs. Ihre Teilnehmer werden in Mesh-Router und Mesh-Clients eingeteilt. Mesh-Router sind stationär und verfügen über eine permanente Energieversorgung. Sie spannen ein drahtloses, statisches Kern-Netz auf, das meist über Gateways an externe Netzwerke angebunden ist. Die mobilen Mesh-Clients sind die Anwendungsträger des Netzwerks und können ebenso als Vermittler für andere Teilnehmer fungieren.

Die in dieser Arbeit entwickelten Routingverfahren sollen auf den Einsatz in drahtlosen Mesh-Netzwerken ausgelegt sein. Diese bieten durch die eingeschränkte Mobilität der Teilnehmer im Kern-Netz die besten Voraussetzungen für Maßnahmen zur Erhöhung der Zuverlässigkeit. Darüber hinaus ist AWDS, das als Ausgangspunkt der Implementierung dienen soll, ein Routingverfahren für drahtlose Mesh-Netzwerke. Die Unterstützung von MANETs ist wünschenswert, soll aber nicht im Fokus dieser Arbeit stehen.

2.2 WLAN

Der Wireless LAN-Standard (WLAN) nach IEEE 802.11 spezifiziert die Übertragung in lokalen Funknetzen. Er wurde 1997 verabschiedet und erfreut sich seitdem immer größerer Beliebtheit. Ein großer Vorteil gegenüber kabelgebundenen Netzwerken ist der geringe Installationsaufwand, weil prinzipiell eine komplizierte Netzplanung sowie das Verlegen von Kabeln entfallen.

2.2.1 Funkkanäle

IEEE 802.11 selbst sowie die Erweiterungen IEEE 802.11b und IEEE 802.11g spezifizieren die Kommunikation im ISM-Band (industrial, scientific, medical) im 2.4GHz-Bereich. Das kontinuierliche Spektrum wird in mehrere diskrete Bereiche eingeteilt, die sogenannten Funkkanäle. Der WLAN-Standard definiert einen Kanalabstand von

5MHz. Ein Kanal ist jedoch 20 MHz breit, so dass sich nah beieinander liegende Kanäle gegenseitig stören. In Mehrkanal-Netzwerken ist es daher sinnvoll, wenn die eingesetzten Kanäle *überlappungsfrei* sind, d.h. einen gewissen Mindestabstand aufweisen. Auf dem in Deutschland freigegebenem Frequenzband lassen sich insgesamt 13 Funkkanäle betreiben, davon bis zu drei überlappungsfrei (Abb. 2.2). Die exakten Frequenzvorgaben sind jedoch regional unterschiedlich. So stehen in den USA elf Funkkanäle (davon drei überlappungsfrei) und in Japan 14 Kanäle (vier überlappungsfrei) zur Verfügung. Das ISM-Band wird von einer Vielzahl weiterer Anwendungen genutzt. Dazu zählen Bluetooth-Geräte, Mikrowellenöfen und Funktelefone.

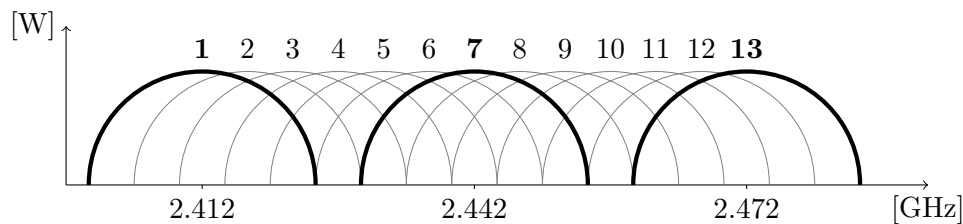


Abbildung 2.2: Beispiel für drei überlappungsfreie WLAN-Funkkanäle im ISM-Band.

Mit der Verabschiedung von IEEE 802.11a wurde die Nutzung des 5GHz-Bandes standardisiert. Auch in diesem Band ist die konkrete Frequenzvergabe regional unterschiedlich, so dass in Europa 19, in den USA zwölf und in Japan vier Kanäle zur Verfügung stehen, die jedoch alle überlappungsfrei sind.

Seit 2002 ist in Europa die Verwendung von IEEE 802.11a mit hohen Sendeleistungen sowie dessen Nutzung außerhalb von Gebäuden erlaubt, sofern zusätzlich IEEE 802.11h genutzt wird. Dieser Standard sieht die Reduktion der Sendeleistung bzw. einen Wechsel des Kanals vor, falls andere Dienste, beispielsweise militärische Radarstationen oder Satellitenfunk, bereits den aktuellen Kanal belegen. Aufgrund der höheren technischen Anforderungen sind entsprechende Geräte allerdings eher selten verbreitet, weswegen die Anzahl der Störquellen deutlich geringer ist, als im 2.4GHz-Band.

Aufgrund der Frequenzunterschiede besitzen Signale auf verschiedenen Frequenzbändern unterschiedliche Ausbreitungseigenschaften. So werden beispielsweise WLAN-Signale im 5GHz-Band wesentlich stärker durch Reflektion, Dämpfung und Beugung beeinflusst, als solche im 2.4GHz-Band [LC]. Dies hat Auswirkungen auf die Erreichbarkeit mehrkanal-fähiger Teilnehmer, besonders innerhalb von Gebäuden. So ist grundsätzlich davon auszugehen, dass sich Topologien auf verschiedenen Funkkanälen voneinander unterscheiden.

2.2.2 Interferenzen

Interferenzen entstehen, wenn gleichzeitig mehrere Teilnehmer auf das Medium (d.h. den gleichen Funkkanal) zugreifen. Dabei überlagern und verfälschen sich ihre Signale, wodurch sie im Empfänger nicht mehr korrekt ausgewertet werden können. Die Vermeidung von Interferenzen wird im WLAN-Standard weitgehend durch das Medienzugriffsverfahren CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) sichergestellt. Ein sendebereiter Teilnehmer beobachtet zunächst das Medium, um festzustellen, ob es gerade durch einen anderen Teilnehmer blockiert wird. Ist dies der Fall, wird die eigene Übertragung verzögert.

Die Tatsache, dass alle nahe gelegenen Teilnehmer miteinander interferieren können, ist eine Eigenschaft von Funknetzwerken, die drahtgebundene Netzwerke nicht haben. Sie führt zu einer deutlichen Einschränkung der Pfad-Kapazität drahtloser Mesh-Netzwerke (weitere Erläuterungen liefert Abschnitt 2.3.2).

Der Interferenzbereich

Durch die Dämpfung der Funksignale sinkt die Güte einer Verbindung mit steigender Entfernung der Teilnehmer. Man teilt den Bereich um den Sender daher in verschiedene Bereiche ein, die Auskunft über die Verbindungsqualität geben (Abb. 2.3).

Teilnehmer, zwischen denen die Kommunikation in beide Richtungen möglich ist, befinden sich in *Übertragungsbereich* r . Außerhalb des Übertragungsbereichs treten zu hohe Fehlerraten auf, welche die fehlerfreie Dekodierung des Signals im Empfänger verhindern. Das Signal ist jedoch bis zur Überschreitung der *Interferenzbereich* c noch messbar. Durch das CSMA/CA-basierte Medienzugriffsverfahren werden Sendevorgänge anderer Teilnehmer innerhalb des *Interferenzbereichs* erkannt und ein gleichzeitiger Zugriff verhindert.

Die Größe des Interferenzbereichs hat einen wesentlichen Einfluss auf die Kapazität von Pfaden in drahtlosen Mesh-Netzwerken. Messungen haben gezeigt, dass die Interferenzbereich im Freien als auch innerhalb von Gebäuden etwa das Doppelte der Übertragungsbereich beträgt [Wew06]. So kann die Anzahl der Störer wesentlich größer sein, als die Anzahl der Teilnehmer, mit denen eine Kommunikation möglich ist.

Übersprechen innerhalb eines Teilnehmers

Obwohl theoretisch zwischen nicht-überlappenden Funkkanälen keine Interferenzen auftreten, trifft dies in der Praxis nicht immer zu. In Teilnehmern mit mehreren Netzwerkkarten wurde beobachtet, dass sich selbst Funkkanäle mit hinreichend großem Frequenzabstand gegenseitig beeinflussen [DPZ04, RGC04, RC05] und eine Verringerung des Durchsatzes hervorrufen können. Es wird vermutet, dass die Ursache zu

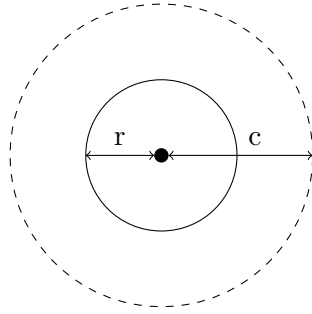


Abbildung 2.3: Übertragungsbereich (durchgehender Kreis) und Interferenzbereich (gestrichelter Kreis) eines Teilnehmers.

einem großen Teil in der mangelhaften Umsetzung der Frequenzfilter der Netzwerkkarten liegt. Dadurch werden auch Frequenzen außerhalb des zulässigen Bereichs ausgestrahlt, welche in den restlichen Netzwerkkarten des Teilnehmers zur Detektion eines blockierten Mediums führen.

Raniwala et al. untersuchen diesen Effekt des Übersprechens detailliert in [RGC04]. So wird gezeigt, dass er von der gegenseitigen Anordnung der Netzwerkkarten abhängt und weitgehend minimiert werden kann, wenn die Karten nebeneinander statt übereinander platziert werden. Weitere Verbesserungen können durch die Nutzung externer Antennen erreicht werden. Ein Übersprechen tritt nicht auf, wenn die eingesetzten Funkkanäle auf verschiedenen Frequenzbändern liegen.

Bei der Konzeption und dem Aufbau eines Mehrkanal-Netzwerks muss unbedingt auf die Minimierung des Übersprechens geachtet werden, um keine unnötigen Einschränkungen des Durchsatzes hinnehmen zu müssen.

2.3 Routing in drahtlosen Mesh-Netzwerken

Drahtlose Mesh-Netzwerke sind spezielle Multi-Hop-Netzwerke, welche die Kommunikation entfernter Teilnehmer durch die Weiterleitung der Daten über Zwischenstationen ermöglichen. Diese Aufgabe übernehmen Routingverfahren. Routingverfahren bestehen aus den Prozessen des *Routings*, d.h. der Pfadsuche und der Verwaltung der dazu notwendigen Informationen, sowie des *Forwardings*, d.h. der eigentlichen Übermittlung des Pakets durch das Netzwerk.

Für den Einsatz in drahtlosen Mesh-Netzwerken sind proaktive, topologie-basierte Routingverfahren zu bevorzugen [YWK05]. Ihre Funktionsweise wird im Folgenden erläutert. Ein spezielles proaktives, topologie-basiertes Routingverfahren ist AWDS. Es bildet die Basis der Implementierung dieser Arbeit und wird in Abschnitt 2.3.3 vorgestellt.

Topologie-basierte Routingverfahren sehen die Speicherung der Netzwerk-Topologie oder eines Teils davon in allen Teilnehmern vor, die dazu Informationen über die eige-

nen Nachbarschaftsbeziehungen (*link state*) durch das gesamte Netzwerk propagieren müssen. Sie werden daher auch als *Link-State-Verfahren* bezeichnet.

Die Speicherung der Topologie erfolgt als kantengewichteter Graph, dessen Knoten die Teilnehmer repräsentieren. Können sich Teilnehmer direkt erreichen, so werden die zugehörigen Knoten mit einer Kante verbunden. Das Gewicht der Kante entspricht den Kosten der Verbindung, die durch die *Metrik* festgelegt werden. Je nach eingesetzter Metrik lassen sich verschiedene Eigenschaften der Verbindung betrachten, gängige Metriken ziehen Kriterien heran, anhand derer sich der Durchsatz der Verbindung abschätzen lässt (weitergehende Erläuterungen liefert Abschnitt 2.3.1).

Die Suche nach einer optimalen Route durch das Netzwerk erfolgt als Suche nach dem kürzesten Pfad im Topologie-Graphen, d.h. nach dem Pfad mit den geringsten Kosten. Hierzu werden meist Shortest-Path-Verfahren wie der Algorithmus von Dijkstra [Dij59] oder Bellman-Ford [BC56, FJ56] eingesetzt. Gegenüber reaktiven Verfahren führen proaktive Verfahren periodisch eine Pfadsuche durch, selbst wenn keine Datenübertragungen stattfinden.

Für das Forwarding von Paketen bietet sich aufgrund seiner Robustheit gegenüber Veränderungen der Topologie ein *Hop-by-Hop-Verfahren* an [YWK05]. Dabei wird im Paket lediglich der Empfänger vermerkt. Jeder Teilnehmer entlang des Pfades leitet das Paket an denjenigen Nachbarn weiter, der aus seiner Sicht dem Empfänger am nächsten ist.

2.3.1 Linkmetriken für drahtlose Mesh-Netzwerke

Die Metrik gibt an, anhand welcher Kriterien die Kosten einer einzelnen Verbindung ermittelt werden. Sie spezifiziert außerdem, auf welche Weise die Kosten eines gesamten Pfades aus den Kosten der einzelnen Pfadabschnitte zusammengesetzt werden.

Yang et al. arbeiten in [YWK05] die Eigenschaften effizienter Routing-Metriken für Mesh-Netzwerke heraus. Dazu gehört unter anderem die Eigenschaft der *Isotonie*. Eine isotonische Metrik garantiert, dass die Relation zweier Pfadgewichte erhalten bleibt, wenn sie um eine weitere Verbindung ergänzt werden. Eine nicht-isotonische Metrik erlaubt, dass die Teilstücke eines kürzesten Pfades nicht zwangsläufig ebenfalls kürzeste Pfade sind. Diese Eigenschaft widerspricht den Voraussetzungen von Shortest-Path-Verfahren und erzwingt einen exponentiellen Aufwand für die Suche nach optimalen Pfaden. Des Weiteren kann die Nutzung einer nicht-isotonischen Metrik zur Entstehung von Routing-Schleifen führen. Routing-Schleifen bezeichnen eine Situation, in der zwei Teilnehmer jeweils den anderen als nächsten Hop für ein bestimmtes Ziel vorsehen. Ein an dieses Ziel gerichtetes Paket wird ständig zwischen beiden Teilnehmern hin- und hergeleitet, sofern keine Gegenmaßnahmen getroffen werden.

Hop-Count

Die Hop-Count-Metrik wird in vielen kabelgebundenen Netzwerken eingesetzt und fand daher auch in den ersten Routingverfahren für Mesh-Netzwerke Verwendung [JM96, PR99]. Sie weist jeder Verbindung ein Gewicht von 1 zu. Demnach entsprechen die Kosten eines Pfades durch das Netzwerk der Anzahl der Verbindungen (Hops), die ein Paket vom Sender zum Empfänger durchlaufen muss. Dadurch wird dem Umstand Rechnung getragen, dass lange Pfade eine schlechtere Leistung bieten, als kurze Pfade. Die Bewertung von Verbindungen mittels Hop-Count lässt sich trivial aus der Topologie bestimmen. Die Hop-Count-Metrik beachtet allerdings die unterschiedlichen Übertragungs- und Verlustraten der Verbindungen nicht. Für den Einsatz in drahtlosen Netzwerken ist sie daher wenig geeignet.

ETX

Die ETX-Metrik (Expected Transmission Count) [DABM03] ist eine Weiterentwicklung von Hop-Count. Sie betrachtet zusätzlich die Verlustraten einer Verbindung. Der ETX-Wert wird als der Durchschnittswert der Übertragungen definiert, die nötig sind, um ein Paket erfolgreich auszuliefern. Da in Link-State-Verfahren ohnehin periodisch Pakete mit allen Nachbarn ausgetauscht werden, entsteht durch die Nutzung der ETX-Metrik kein zusätzlicher Aufwand. Die spezifischen Übertragungsraten der einzelnen Verbindungen im Netzwerk werden jedoch immer noch nicht betrachtet.

ETT

Die ETT-Metrik (Estimated Transmission Time) [Bic05] erweitert die ETX-Metrik um die Betrachtung der Übertragungsrate einer Verbindung. Sie entspricht der geschätzten Dauer, die für die erfolgreiche Übertragung eines Paketes benötigt wird. Die Beziehung zwischen dem ETT-Wert einer Verbindung v und ihrem ETX-Wert wird als

$$ETT_v = ETX_v \frac{s}{b_v},$$

definiert, wobei b_v die Übertragungsrate der Verbindung und s die Paketgröße angibt. Die ETT-Metrik erfordert ebenfalls den regelmäßigen Austausch von Paketen mit allen Nachbarn. Im Gegensatz zur ETT-Metrik reicht der Austausch von Broadcast-Paketen jedoch nicht aus, weil Broadcast-Pakete stets mit der niedrigsten Senderate verschickt werden. Eine Bewertung der Verbindungen mittels ETT-Metrik verlangt daher den periodischen Versand von Unicast-Paketen an alle Nachbarn, wodurch das Netzwerk zusätzlich belastet wird.

2.3.2 Spezielle Effekte in Multi-Hop-Netzwerken

Aufgrund der allseitigen Ausbreitung von Funksignalen wird das Medium in drahtlosen Netzwerken von vielen Teilnehmern geteilt. Besonders in drahtlosen Multi-Hop-Netzwerken besteht eine hohe Wahrscheinlichkeit, dass viele der Teilnehmer gleichzeitig senden möchten, was durch den CSMA-basierten Medienzugriff jedoch verhindert wird. Die Folge sind lange Wartezeiten auf den Medienzugriff, die mit einer Verringerung des Durchsatzes einhergehen. Durch die Wahl geeigneter Pfade können diese Effekte jedoch teilweise minimiert werden.

Gegenseitige Beeinflussung der Teilnehmer entlang des Pfades

Nachfolgende Teilnehmer entlang eines Multi-Hop-Pfades konkurrieren miteinander um den Medienzugriff, weil sich ihre Interferenzbereiche überschneiden.

Abbildung 2.4 (entnommen aus [LBDC⁺01]) soll diesen Sachverhalt verdeutlichen. Sie zeigt eine Multi-Hop-Übertragung von Teilnehmer A zu Teilnehmer E, die Interferenzreichweite (gestrichelter Kreis) sei doppelt so groß wie die Übertragungreichweite (durchgehender Kreis). Das erste Paket wird gerade von Teilnehmer D zu Teilnehmer E geschickt. Teilnehmer A versucht bereits, das zweite Paket abzuschicken, weil er sich außerhalb des Interferenzbereichs von D befindet und ein freies Medium feststellt. Beim Empfänger des zweiten Pakets, Teilnehmer B, überlappen sich jedoch die Interferenzbereiche von A und D. Es treten Interferenzen auf, die eine erfolgreiche Zustellung des zweiten Pakets verhindern. Erst nachdem das erste Paket bei Teilnehmer E angekommen ist, kann Teilnehmer A das zweite Paket erfolgreich abschicken. In diesem Szenario müssen stets drei Teilnehmer auf ein freies Medium warten, was den Durchsatz der Übertragung gegenüber einer direkten Verbindung theoretisch auf ein Viertel beschränkt. Tatsächlich ist mit weitaus größeren Durchsatzeinbußen zu rechnen, weil die Wahrscheinlichkeit des Medienzugriffs nicht für jeden Teilnehmer identisch ist [LBDC⁺01].

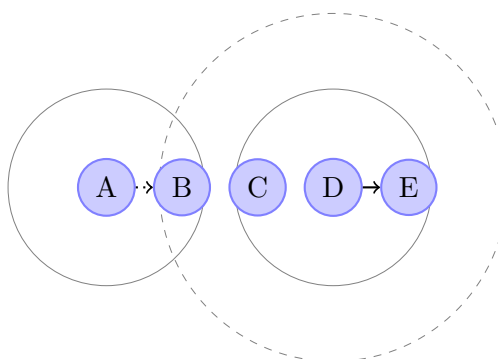


Abbildung 2.4: Multi-Hop-Übertragung von Teilnehmer A zu Teilnehmer E.

Konkurrierende Medienzugriffe entlang des Pfades können in Einkanal-Netzwerken nicht verhindert werden. Sie finden deshalb in Routingverfahren, die speziell für die Nutzung eines einzigen Funkkanals konzipiert wurden, keine Beachtung.

Gegenseitige Beeinflussung der Teilnehmer benachbarter Datenströme

Da sich alle Teilnehmer in Interferenzreichweite das Funkmedium teilen, wird eine Übertragung auch von benachbarten Datenübertragungen negativ beeinflusst. Die Häufigkeit benachbarter Datenströme kann in Einkanal-Netzwerken prinzipiell minimiert werden, indem die Daten über dünn besetzte Bereiche des Netzwerks geleitet werden (Abb. 2.5, entnommen aus [YWK05]), es konnten jedoch keine bestehenden Einkanal-Routingverfahren identifiziert werden, die derartige Berücksichtigungen in die Pfadsuche einfließen lassen.

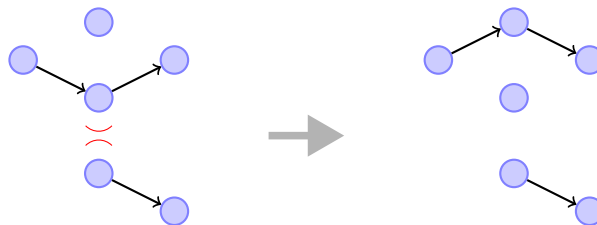


Abbildung 2.5: Vermeidung konkurrierender Medienzugriffe zwischen Teilnehmern unterschiedlicher Datenströme durch Routing über dünn besetzte Bereiche.

2.3.3 Ad-hoc Wireless Distribution Service

Ad-hoc Wireless Distribution Service (AWDS) [awd] ist ein proaktives Link-State-Routingverfahren für drahtlose Mesh-Netzwerke, das am Institut für Verteilte Systeme an der Universität Magdeburg entwickelt wird. AWDS arbeitet oberhalb der Netzzugangsschicht und bietet daher die Möglichkeit, Dienste wie DHCP oder IPv6 zu betreiben.

Der Aufbau der Topologie erfolgt durch die Propagierung der Nachbarschaften aller Teilnehmer. Zur Erkundung der eigenen Nachbarschaft senden alle Teilnehmer periodisch Hello-Pakete per Broadcast aus. Anhand der empfangenen Hello-Pakete wird ermittelt, welche anderen Teilnehmer sich in der eigenen Nachbarschaft befinden. Verbindungsabbrüche werden durch mehrmaliges Ausbleiben von Hello-Paketen eines Nachbarn erkannt.

Die Liste der eigenen Nachbarn wird periodisch durch das Netzwerk geflutet. Das Fluten eines Paketes geschieht durch den Versand als Broadcast-Paket. Empfängt ein

Teilnehmer ein solches Flutpaket zum ersten Mal, sendet er es ebenfalls per Broadcast weiter, so dass das Paket nach einiger Zeit alle Teilnehmer des Netzwerks erreicht. Jeder Teilnehmer erhält demnach Informationen über die Nachbarschaften aller anderen Teilnehmer und führt diese lokal zur gesamten Netztopologie zusammen.

Um Anwendungsdaten über das Netzwerk zu verschicken, erzeugt AWDS eine virtuelle Ethernet-Schnittstelle im Endgerät. Ist die Zieladresse bekannt, werden an diese Schnittstelle gesendete Daten per Unicast zum Ziel geroutet, andernfalls durch das gesamte Netzwerk geflutet. Am Ziel angelangt, werden die Daten von der dortigen Anwendung über die virtuelle Schnittstelle entgegengenommen. Aus Sicht der Anwendung sind alle dieser virtuellen Schnittstellen über Ethernet verbunden, AWDS arbeitet also transparent. Durch Bridging der virtuellen Schnittstelle mit anderen Netzwerk-Schnittstellen kann AWDS in bestehende Netzwerk-Infrastrukturen eingebunden werden.

AWDS ist modulbasiert. Es lässt sich durch Hinzunahme oder Austausch von Modulen bequem im Funktionsumfang erweitern bzw. anpassen. So existieren unter anderem Module für die Nutzung verschiedener Metriken, verschlüsselter Kommunikation mittels AES oder die direkte Manipulation über eine Kommandozeile.

2.4 Mehrkanal-Routingverfahren

Drahtlose Netzwerke sind sehr anfällig für Kanalstörungen. Darüber hinaus wirken sich konkurrierende Medienzugriffe der einzelnen Teilnehmern negativ auf den maximal erreichbaren Durchsatz aus.

Die Nutzung mehrerer Funkkanäle verspricht Verbesserungen in beiden Problemfeldern. So wirken sich Kanalstörungen nicht so fatal aus, weil jederzeit ein Ausweichkanal zur Verfügung steht. Weiterhin kann durch die abschnittsweise Aufteilung eines Datenstroms auf unterschiedliche Funkkanäle die Anzahl der Teilnehmer verringert werden, die gleichzeitig um das Medium konkurrieren (Abb. 2.6, entnommen aus [YWK05]). Dadurch entstehen geringere Wartezeiten auf den Medienzugriff, wodurch der Durchsatz gesteigert wird.

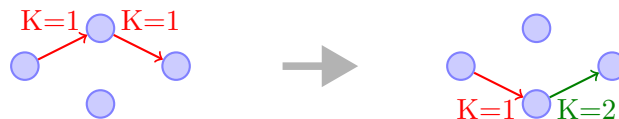


Abbildung 2.6: Vermeidung konkurrierender Medienzugriffe der Teilnehmer entlang eines Pfades durch die Nutzung verschiedener Funkkanäle. K gibt den Funkkanal einer Verbindung an.

Die simple Nutzung der in Einkanal-Routing eingesetzten Pfadsuche ist jedoch nicht zielführend, weil diese keine Lastverteilung auf die verschiedenen Funkkanäle durchführt. Das Problem konkurrierender Medienzugriffe bei Multi-Hop-Übertragungen tritt zwar auch in Einkanal-Netzwerken auf, es findet dort allerdings keine Berücksichtigung, weil es nur durch die Nutzung weiterer Funkkanäle gelöst werden kann. In Mehrkanal-Netzwerken muss daher die Definition des optimalen Pfades um die Forderung erweitert werden, dass dessen Abschnitte möglichst gleichmäßig über die verfügbaren Funkkanäle verteilt sind. Dann ist die optimale Pfadsuche jedoch nicht mehr trivial lösbar.

Da sich Signale verschiedener Funkkanäle nicht beeinflussen, müssen Pfadkosten, die durch die Nutzung unterschiedlicher Funkkanäle entstehen, unabhängig voneinander betrachtet werden. Weil dann die Teilstücke eines optimalen Pfades nicht zwangsläufig selbst optimal sind, wird das Bellman-Ford-Kriterium verletzt. Es ist jedoch Voraussetzung für die Konstruktion von Shortest-Path-Verfahren, die mit vertretbarem Aufwand lösbar sind. Ohne das Kriterium besteht keine Möglichkeit, frühzeitig schlechte Pfade zu erkennen und auszuschließen. Der optimale Pfad kann in Mehrkanal-Netzwerken daher nur gefunden werden, indem alle möglichen vollständigen Pfade bestimmt und miteinander verglichen werden. Da die Anzahl der Pfade mit wachsender Topologie exponentiell steigt, ist Nutzung der optimalen Pfadsuche in Mehrkanal-Topologien nicht praktikabel.

Weiterhin sind die Endgeräte drahtloser Mesh-Netzwerke meist stromsparende Geräte mit geringer Rechenleistung, so dass die Grenze der Berechenbarkeit bereits bei sehr kleinen Topologien überschritten wird. Um einen praktikablen Einsatz dennoch zu garantieren, muss ein Kompromiss aus Optimalität und der Dauer der Pfadsuche gefunden werden.

In den letzten Jahren wurde daher eine Vielzahl spezieller Mehrkanal-Routingprotokolle entwickelt, die in diesem Abschnitt exemplarisch vorgestellt werden sollen.

2.4.1 Klassifizierung

Die Nutzung der verschiedenen Funkkanäle wird in den bereits bestehenden Mehrkanal-Routingprotokollen auf sehr unterschiedliche Weise umgesetzt. Dieser Abschnitt arbeitet Eigenschaften von Mehrkanal-Routingverfahren heraus, die eine grobe Klassifikation ermöglichen. Abbildung 2.7 zeigt eine Übersicht.

Dynamik

Die Dynamik eines Mehrkanal-Routingverfahrens gibt an, ob die zu Beginn festgelegte Kanalzuweisung während des Betriebs verändert werden darf.

Statische Verfahren verändern die ursprüngliche Kanalzuweisung nicht. Dies vereinfacht die Synchronisation der Teilnehmer untereinander, erfordert jedoch eine gewisse

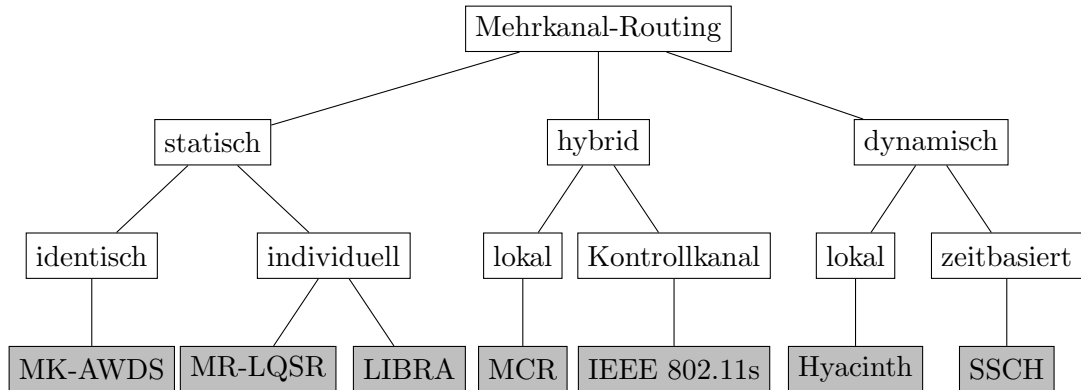


Abbildung 2.7: Klassifizierung von Mehrkanal-Routingverfahren anhand der Kanalnutzung.

Planung ihrer Platzierung, um während des Betriebs eine ausreichende Konnektivität zu garantieren.

Dagegen erlauben *hybride* und *dynamische Mehrkanal-Routingverfahren* die Anpassung der Funkkanäle während des Betriebs. Die Kanalzuweisung besteht aus zwei Problemstellungen. Zum Einen muss geklärt werden, welche Nachbarn miteinander direkt kommunizieren dürfen, d.h. einen gemeinsamen Kanal zugewiesen bekommen sollen. Zum Anderen muss dieser Verbindung eine konkrete Frequenz zugewiesen werden. Dabei können neben der Minimierung der Auslastung des Mediums auch spezielle Vorgaben der Dienstgüte eine Rolle spielen.

Während bei hybriden Verfahren ein Teil der Funkkanäle fest zugewiesen wird und nur der restliche Teil verändert werden darf, erlauben dynamische Verfahren die Änderung der Kanalzuordnung in allen Netzwerkkarten. Sie eignen sich damit auch für Netzwerke, deren Teilnehmer jeweils nur eine einzige Netzwerkkarte enthalten.

Es sei betont, dass statische Verfahren ein Spezialfall hybrider Verfahren sind. Hybride Verfahren sind wiederum ein Spezialfall dynamischer Verfahren, sofern diese den Einsatz mehrerer Netzwerkkarten pro Teilnehmer vorsehen.

Zuordnung der Kanäle zu den Netzwerkkarten

Statische Mehrkanal-Verfahren lassen sich danach klassifizieren, ob es eine feste Kanalzuordnung gibt, in der jedem Teilnehmer die gleiche Menge von Funkkanälen zur Verfügung stehen, oder ob Teilnehmer untereinander auf verschiedene Kanäle zugreifen dürfen (Abb. 2.8).

Wird allen Teilnehmern die gleiche Menge an Frequenzen zugewiesen (*identische Zuweisung*), erreicht das Netzwerk die größtmögliche Konnektivität. Jeder Teilnehmer kann seine Nachbarn sofort erreichen. Der zu erwartende Durchsatzgewinn fällt jedoch moderat aus, weil insgesamt nur wenige verschiedene Funkkanäle benutzt werden.

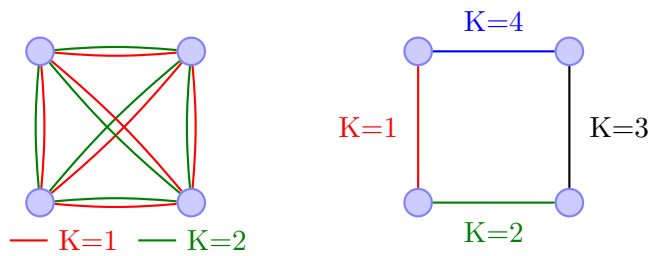


Abbildung 2.8: Topologie bei identischer Kanalzuordnung (links) und individueller Kanalzuordnung (rechts) bei Teilnehmern mit zwei Netzwerkkarten. K gibt den Funkkanal einer Verbindung an.

Eine andere Möglichkeit besteht darin, den Teilnehmern teilweise verschiedene Kanäle zuzuweisen (*individuelle Zuweisung*). Dadurch lassen sich insgesamt mehr Funkkanäle nutzen, als Netzwerkkarten pro Teilnehmer zur Verfügung stehen, was höhere Durchsatzraten erlaubt. Dabei verringert sich allerdings die Erreichbarkeit der Teilnehmer untereinander, weil diese nur noch mit anderen Teilnehmern kommunizieren können, die den gleichen Funkkanal benutzen. Teilnehmer, die neu in das Netzwerk aufgenommen werden sollen, müssen zunächst auf allen verfügbaren Funkkanälen nach Nachbarn suchen.

Verfahren, die eine individuelle Zuweisung erlauben, lassen sich ebenso in Netzwerken mit identischer Zuweisung einsetzen.

Koordination

Hybride und dynamische Verfahren erlauben während des Betriebs die Änderung einiger bzw. aller Funkkanäle. Man unterscheidet dabei die Art der Koordination der Teilnehmer.

Bei *lokalen Verfahren* findet keine Koordination der Teilnehmer statt. Jeder Teilnehmer beobachtet seine Nachbarschaft und führt anhand gewisser Kriterien, beispielsweise aufgrund eines stark ausgelasteten Mediums, eigenständig einen Kanalwechsel durch.

In hybriden Verfahren besteht dagegen auch die Möglichkeit der Nutzung eines *gemeinsamen Kontrollkanals*. Über den Kontrollkanal tauschen sich alle Teilnehmer Statusinformationen aus und verabreden die Nutzung eines bestimmten Funkkanals für eine anstehende Datenübertragung.

Dynamische Routingprotokolle erlauben den Einsatz *zeitbasierter Verfahren*. Dabei wechselt jeder Teilnehmer periodisch die Zuordnung der Funkkanäle zu seinen Netzwerkkarten gemäß einer individuellen Vorschrift. Dies erfordert eine strenge Zeitsynchronisation der Teilnehmer.

2.4.2 Ausgewählte Mehrkanal-Routingverfahren

Im Folgenden wird für jede der vorgestellten Klassifizierungen ein stellvertretendes Mehrkanal-Routingverfahren vorgestellt. Alle Verfahren wurden auf die Maximierung des Durchsatzes konzipiert, während der Recherchen zu verwandten Arbeiten konnten keine Routingprotokolle identifiziert werden, die zuverlässigkeits-relevante Ziele verfolgen.

Mehrkanal-AWDS (MK-AWDS)

Im Rahmen des wissenschaftlichen Teamprojektes des Master-Studiengangs Informatik an der Universität Magdeburg wurde bereits eine Machbarkeitsstudie für den Betrieb von AWDS auf mehreren Funkkanälen durchgeführt [Sch08]. Sie beschränkt sich auf die Nutzung zweier Funkkanäle, die je einer Netzwerkkarte fest zugewiesen werden. Für jeden Funkkanal wird eine eigene, logische Topologie aufgebaut, ein Wechsel des Kanals während einer Übertragung ist nicht erlaubt. Die Lastverteilung wird allein im Sender durchgeführt, indem mithilfe einer gewichteten, zufälligen Auswahl des Sendekanals die Auslastung beider Kanäle minimiert wird. Die prototypische Implementierung erzielte in einer realen Testumgebung eine gute Leistung, so dass AWDS nun zur Evaluierung dieser Masterarbeit eingesetzt werden soll.

MR-LQSR

Das Multi-Radio Link-Quality Source-Routing-Protokoll (MR-LQSR) [DPZ04] geht von einem vollkommen stationären Netzwerk aus. Die Zuweisung von Funkkanälen zu den Netzwerkkarten der Teilnehmer kann individuell verschieden sein. Die Besonderheit des Verfahrens ist eine Pfadmetrik namens Weighted Cumulative ETT (WCETT), die speziell die gegenseitige Beeinflussung der Teilnehmer entlang des Pfades berücksichtigt. Mittels der ETT-Metrik wird jeder Verbindung ein Gewicht zugewiesen, das sowohl Verlustraten als auch die Senderate berücksichtigt. Diese Gewichte werden für jeden Funkkanal gesondert gespeichert. Pfade werden zum Einen nach der Gesamtsumme der beteiligten Gewichte, zum Anderen nach dem Maximalwert der Kanalsummen bewertet. Der Einfluss dieser beiden Faktoren kann mittels eines Wichtungsparameters justiert werden. Durch eine gleichstarke Wichtung werden Pfade bevorzugt, die möglichst kurz sind und häufig die Kanäle wechseln.

Yang et al. weisen in [YWK05] die fehlende Isotonie-Eigenschaft der WCETT-Metrik nach. Diese hat zur Folge, dass optimale Pfade nur mit exponentiellem Aufwand gefunden werden können, weil die Pfadsuche mittels Shortest-Path-Verfahren nicht zuverlässig arbeitet.

LIBRA

Load and Interference Balanced Routing Algorithm (LIBRA) [YWK] ist ein proaktives Routingprotokoll. Es setzt eine statische Kanalzuordnung voraus, die jedoch individuell sein kann, d.h. von Teilnehmer zu Teilnehmer verschieden ist. Durch die Verwendung der Metric of Interference and Channel Switching (MIC) erweitert es WCETT um die Eigenschaft, die Beeinflussung einer Übertragung durch benachbarte Datenströme in die Berechnung des Gewichts einzubeziehen. Im Gegensatz zu MR-LQSR erlaubt LIBRA, die fehlende Isotonie-Eigenschaft der Metrik durch die Hinzunahme virtueller Knoten wiederherzustellen. So besteht jeder Teilnehmer aus je einem Empfangs- und Sendeknoten pro Funkkanal, sowie zwei weiteren Knoten, welche die Endpunkte ein- und ausgehender Datenströme darstellen. Die Empfangsknoten eines Teilnehmers werden mit seinen Sendeknoten verbunden. Findet dabei ein Kanalwechsel statt, erhält die Verbindung geringere Kosten als bei der Beibehaltung des Kanals. Pfade mit Kanalwechsel sind dadurch kürzer als Pfade ohne Kanalwechsel, wodurch sie bei der Pfadsuche bevorzugt werden.

MCR

Das Multichannel Routing Protocol (MCR) [Kya06] wählt einen hybriden Ansatz. Die Menge der verfügbaren Netzwerkkarten wird unterteilt in solche, die Daten empfangen und solche, die Daten versenden dürfen. Netzwerkkarten, die dem Empfang von Daten dienen, bekommen über einen längeren Zeitraum einen festen Kanal zugewiesen. Diese Zuweisung wird periodisch der Nachbarschaft des Teilnehmers bekannt gegeben. Anhand der Empfangskanäle der Nachbarn kann festgestellt werden, ob der eigene Empfangskanal stark ausgelastet ist und gegebenenfalls gewechselt werden muss. Es wird jedoch davon ausgegangen, dass ein Wechsel des Empfangskanal nur sehr selten durchgeführt werden muss. Neben der eher statischen Zuweisung der Empfangskanäle sind die Sendekanäle sehr dynamisch. Möchte ein Teilnehmer Daten übertragen, so wechselt er den eigenen Sende-Kanal auf den Empfangs-Kanal des nächsten Hops. Pakete, die aufgrund der momentanen Kanalnutzung nicht zugestellt werden können, werden zwischengespeichert und später versendet. Dadurch wird ein übermäßiges Wechseln des Kanals vermieden.

IEEE 802.11s

Der sich momentan in Entwicklung befindliche Standard IEEE 802.11s (Mesh Networking) sieht für die Unterstützung mehrerer Funkkanäle die Nutzung des Common Control Channel-Protokolls (CCC) [Ben] vor. Es definiert zwei logische Kanal-Typen, einen Kontrollkanal und einen oder mehrere Datenkanäle. Der Kontrollkanal dient ausschließlich dem Austausch von Kontrollnachrichten. Diese werden von den Teilnehmern eingesetzt, um für eine bevorstehende Datenübertragung einen Datenkanal

zu reservieren. Der Kontrollkanal wird gewöhnlich einem separaten Funkkanal zuge-
teilt. Teilnehmer mit nur einer Netzwerkkarte können ihn jedoch auch als Datenkanal
verwenden. Das CCC-Protokoll erfordert eine strenge zeitliche Synchronisation der
Teilnehmer. Dies stellt jedoch kein Problem dar, da das Protokoll in die MAC-Schicht
integriert ist.

Als eigentliches Routingverfahren sieht IEEE 802.11s den Einsatz des Hybrid Wireless
Mesh Routing Protocol (HWMP) [J⁺06] vor, das aus zwei verschiedenen Routing-
protokollen besteht. Für den mobilen Teil des Netzwerks wird ein reaktives Verfah-
ren eingesetzt, während für den stationären Teil des Mesh-Netzwerks ein proaktives,
baum-basiertes Verfahren Verwendung findet.

Hyacinth

Hyacinth [RC05] ist ein dynamisches Routingverfahren, das den Wechsel des Funk-
kanals in allen Netzwerkkarten des Teilnehmers während des Betriebs zulässt. Hya-
cynth geht von einem Anwendungsfall aus, in dem Endgeräte über das Mesh-Netzwerk
mit Gateways verbunden sind, die den Anschluss an ein kabelgebundenes Netzwerk
ermöglichen. Der Aufbau der Topologie erfolgt baum-basiert, das heißt es gibt vom
Endgerät zum Gateway genau einen Pfad. Die Verbindungen zu anderen Teilnehmern
werden daher in zwei Kategorien eingeteilt, Links in Richtung der Gateways und Links
in Richtung der Endgeräte. Diese Links werden von unterschiedlichen Netzwerkkarten
bedient.

Die konkrete Zuweisung von Kanälen zu den Netzwerkkarten ist ein zentrales Problem
des Verfahrens und wird anhand des erwarteten Durchsatzes durchgeführt. Durch eine
Analyse der erwarteten Datenströme wird zunächst das Datenaufkommen aller Verbin-
dungen geschätzt. Anschließend besucht Hyacinth jede Verbindung nach absteigender
Reihenfolge ihres Datenaufkommens und weist ihr denjenigen Kanal zu, der im Inter-
ferenzbereich der beteiligten Teilnehmer momentan am geringsten ausgelastet ist.

Hyacinth durchläuft mehrere Iterationen aus Kanalzuweisung und Pfadsuche. Die Ka-
nalzuweisung leitet anhand des erwarteten Datenaufkommens die Nutzung der Kanäle
ab. Dadurch verändern sich die Kapazitäten der Verbindungen, weswegen die Pfad-
suche nun andere Pfade als zuvor ermittelt. Dadurch verändert sich das erwartete
Datenaufkommen einer Verbindung, was einen neuen Durchlauf der Kanalzuweisung
erfordert. Dieser iterative Prozess wiederholt sich, bis ein stabiler Zustand erreicht
wird.

SSCH

Slotted Seeded Channel Hopping (SSCH) [BCD04] ist ein dynamisches Mehrkanal-
Routingverfahren, das nur eine Netzwerkkarte pro Teilnehmer voraussetzt. Das Ver-
fahren basiert auf Zeitschlitzten. Für die Dauer eines Zeitschlitzes bleibt die Kanal-
zuweisung in jedem Teilnehmer konstant, eine Änderung findet nur beim Übergang

in den nächsten Zeitschlitz statt. Dies geschieht nach einer gewissen Vorschrift, die sich in jedem Teilnehmer unterscheidet. Dadurch wird erreicht, dass ein Teilnehmer in verschiedenen Zeitschlitzten mit jeweils anderen Nachbarn kommunizieren kann.

Durch den regelmäßigen Austausch der Vorschriften ist bekannt, wann ein bestimmter Nachbar erreichbar ist. Um die Übertragung von Nutzdaten zu beschleunigen, kann die eigene Vorschrift an die des Nachbarn angepasst werden, so dass dieser öfter erreicht werden kann.

Da SSCH in die MAC-Schicht integriert wird, kann es mit beliebigen Einkanal-Routingverfahren kombiniert werden.

Bewertung der Kanalnutzung hinsichtlich Durchsatz und Zuverlässigkeit

Den größten Durchsatzgewinn erreichen Verfahren, die alle überlappungsfreien Funkkanäle gleichzeitig nutzen können. Dazu zählen neben dynamischen und hybriden Routingprotokollen auch statische Verfahren mit individueller Kanalzuweisung. Je mehr Funkkanäle jedoch benutzt werden, desto größer ist die Wahrscheinlichkeit, dass ein Nachbar auf einer anderen Frequenz sendet und daher nicht sofort erreichbar ist. Ein Wechsel des Kanals ist allerdings vergleichsweise teuer. So zeigen Chereddi et al. in [CKV06], dass die Umschaltzeit selbst unter Anwendung verschiedener Optimierungen noch einige Millisekunden beträgt. Dieser Nachteil fällt dann stark ins Gewicht, wenn häufige Kanalwechsel stattfinden. Sollen Daten über einen Funkkanal gesendet werden, der im Teilnehmer momentan nicht aktiv ist, müssen diese in einer Warteschlange zwischengespeichert werden. Dadurch entstehen weitere Latenzen. Außerdem benötigt die Aufnahme neuer Teilnehmer in das Netzwerk einige Zeit, weil diese zuerst alle möglichen Frequenzen nach potentiellen Nachbarn absuchen müssen.

Demgegenüber stehen statische Verfahren mit identischer Kanalzuweisung. Sie erreichen die höchste Konnektivität, weil alle Teilnehmer miteinander kommunizieren können, ohne zuvor den Funkkanal wechseln zu müssen. Im Falle eines Knoten- oder Kanalausfalls kann sofort eine neue Route berechnet werden. Das Verfahren Mehrkanal-AWDS stellt allerdings spezielle Bedingungen an die Topologie, weil mindestens ein durchgehender Pfad auf dem gleichen Funkkanal bis zum Empfänger existieren muss. Das Netzwerk ist dadurch sehr anfällig für Verbindungsausfälle.

Die Aufnahme neuer Teilnehmer erfolgt in Netzwerken mit identischer Kanalzuweisung vergleichsweise schnell, weil die aufwändige Suche nach möglichen Kanälen entfällt. Da die Anzahl der Funkkanäle an die Anzahl der Netzwerkkarten pro Teilnehmer gekoppelt ist, ist der Ausbau der Kapazität nur in begrenztem Umfang möglich. Dadurch fällt die zu erwartende Durchsatzsteigerung vergleichsweise gering aus.

Der Vorteil eines Kontrollkanals ist eher fraglich. Nutzt das Netzwerk insgesamt wenige Funkkanäle, bedeutet der Kontrollkanal viel Overhead, weil im Mehrkanal-Betrieb auf ihm keine Nutzdaten gesendet werden. In einem Netzwerk mit vielen Funkkanälen besteht dagegen die Gefahr, dass der Kontrollkanal durch das vermehrte Aufkommen von

2 Grundlagen und verwandte Arbeiten

Synchronisations-Nachrichten zum Flaschenhals wird und dadurch die eigentlichen Datenübertragungen ausgebremst werden. Zudem ist eine Störung des Kontrollkanals besonders kritisch, weil das Netzwerk ohne ihn zum Erliegen kommt.

Verfahren, die wie SSCH oder CCC eine strenge zeitliche Synchronisation der Teilnehmer voraussetzen, sind typischerweise in die MAC-Schicht integriert und müssen in Hardware implementiert werden. Dies schließt den Einsatz von günstiger "Standard"-Hardware aus.

3 Konzept

Dieses Kapitel erläutert, auf welche Weise die Grundlage für zuverlässige Kommunikation geschaffen werden soll. Darauf aufbauend werden mit dem Verfahren der diagonalen Links und einem heuristischen Verfahren zwei Mehrkanal-Routingprotokolle vorgestellt, die auf effiziente Weise gute Pfade finden.

3.1 Vorbetrachtungen

3.1.1 Maßnahmen zur Ermöglichung zuverlässiger Kommunikation

Zur Schaffung der Voraussetzungen für zuverlässige Kommunikation in drahtlosen Mehrkanal-Mesh-Netzwerken muss zunächst bekannt sein, welche Arten von Fehlern auftreten können. Dazu soll ein Fehlermodell aufgestellt werden.

Fehlermodell für drahtlose Mesh-Netzwerke

Es wird von fehlerfreiem Entwurf und fehlerfreier Herstellung der Komponenten ausgegangen, so dass lediglich *Betriebsfehler* betrachtet werden. Der Schwerpunkt liegt dabei auf der Berücksichtigung von *störungsbedingten Fehlern*, speziell des Ausfalls der Kommunikationskanäle. Der Einfluss von Störquellen soll nicht konstant sein und kann nach einer Weile wieder komplett verschwinden. Derartige Fehler sind *transient*, weil sie nur vorübergehend existieren und keine fortwährenden Symptome zeigen. Es werden ebenso *permanente Fehler* betrachtet, die ab ihrem Auftreten ständig präsent sind.

Die Grundlage des Fehlermodells soll ein *Crash-recovery-Fehlermodell* bilden [Gä01]. Es behandelt die folgenden Fehlerklassen:

1. Eine einzelne Verbindung wird auf einem Kanal gestört, zum Beispiel aufgrund eines Hindernisses. Sind die Teilnehmer oder das Hindernis mobil, wird die Verbindung nach einiger Zeit wieder verfügbar sein.
2. Eine einzelne Netzwerkkarte eines Teilnehmers fällt aus, so dass er sie nicht mehr einsetzen kann.

3 Konzept

3. Ein Teilnehmer fällt aufgrund eines Hardwaredefekts komplett aus. Dadurch bricht jegliche Verbindung zur Nachbarschaft ab.
4. Eine Gruppe mobiler Endgeräte verliert die Verbindung zum Kern-Netz, weil sie sich zu weit von ihm entfernt hat.
5. Ein Kanal wird komplett gestört, zum Beispiel durch ein benachbartes Funknetz. Es besteht die Möglichkeit, dass der Einfluss des Störers mit der Zeit schwindet, so dass der Kanal wieder genutzt werden kann.

Fehler der Klassen 1 und 5 sind transient. Sie werden durch äußere Einflüsse hervorgerufen, über deren Dauerhaftigkeit keine Aussage getroffen werden kann. Es soll daher nicht ausgeschlossen werden, dass lediglich eine temporäre Störung vorliegt. Die Fehlerklassen 2, 3 und 4 stehen für permanente Fehler.

Fehler der Klasse 1 können bereits in Einkanal-Netzwerken toleriert werden, indem betroffene Pfade über andere Teilnehmer umgeleitet werden. Dies setzt voraus, dass die Topologie auch ohne die ausgefallene Verbindung zusammenhängend ist. In Mehrkanal-Topologien besteht zusätzlich die Möglichkeit, dass die betroffenen Teilnehmer über einen anderen Funkkanal noch miteinander verbunden sind.

Der Ausfall einer Netzwerkkarte (Fehlerklasse 2) ist nur dann tolerierbar, wenn die Teilnehmer mit mehreren Netzwerkkarten ausgerüstet sind. Dies ist bei mehrkanalfähigen Teilnehmern gewöhnlich der Fall.

Der komplette Ausfall eines Teilnehmers (Fehlerklasse 3) ist nur dann tolerierbar, wenn es sich um eine Station des Kern-Netzes handelt und ein alternativer Pfad existiert. Der Ausfall von Endgeräten ist nicht kompensierbar, wenn diese nicht ebenfalls redundant ausgelegt sind.

Tritt ein Fehler der Klasse 4 auf, so ist eine Kommunikation ebenfalls nicht mehr möglich. Im Gegensatz zu Fehlerklasse 3 ist das betroffene Endgerät allerdings noch in der Lage, entsprechend zu reagieren und könnte beispielsweise ein akustisches Warnsignal aussenden. Innerhalb der Routingschicht besteht jedoch keine Möglichkeit, die Konnektivität wiederherzustellen.

Anders als Einkanal-Netzwerke können Mehrkanal-Netzwerke auch Fehler der Klasse 5 tolerieren. Dazu muss die Topologie auf den restlichen Funkkanälen zusammenhängend sein.

Diskussion möglicher Routingprotokolle

Es wurde gezeigt, dass Mehrkanal-Netzwerke mehr Fehler als Einkanal-Netzwerke tolerieren können. Dazu müssen sie jedoch entsprechend redundant ausgelegt sein. Während sich Fehler der Klassen 2 und 3 durch redundante Auslegung der Hardware (z.B. Netzwerkkarten, Teilnehmer) tolerieren lassen, gelingt die Tolerierung von Kanalausfällen nur durch eine redundante Auslegung der Kommunikationskanäle, d.h. die

Bereitstellung alternativer Funkkanäle. Weiterhin ist es vorteilhaft, wenn Verbindungsabbrüche schnell erkannt und neue Pfade rasch aufgebaut werden, um den Versand von Paketen über ineffiziente oder ungültige Pfade zu minimieren.

Das Auftreten eines beliebigen Fehlers führt zu einer Veränderung der Topologie und kann Auswirkungen auf aktive Pfade haben. Es ist daher sinnvoll, wenn das Routingverfahren bereits eine gewisse Robustheit gegen Veränderungen der Topologie aufweist. Hierzu bietet sich die Nutzung von Hop-by-Hop-Verfahren an, die das Paket abschnittsweise bis zum Empfänger leiten. Dies ermöglicht die Korrektur bestehender Pfade während einer Datenübertragung. In [YWK05] wird gezeigt, dass proaktive Link-State-Verfahren für den Einsatz in Mesh-Netzwerken am besten geeignet sind. Sie sollen daher auch in den in dieser Arbeit entwickelten Routingverfahren Verwendung finden.

Die Forderung nach starker Konnektivität und einer schnellen Einsatzbereitschaft im Fehlerfall kann durch die Verwendung einer statischen, identischen Kanaluweisung erreicht werden. Das bedeutet, dass jeder Teilnehmer die gleichen fest voreingestellten Funkkanäle nutzt. Dadurch ist er bereits im Normalfall mit allen Nachbarn verbunden und kann mit ihnen kommunizieren, ohne zunächst einer Netzwerkkarte einen neuen Funkkanal zuweisen zu müssen. Um den Schutz gegen breitbandige Störungen zu verbessern, bietet sich dabei der Einsatz unterschiedlicher Frequenzbänder an (d.h. 2.4GHz und 5GHz).

Hybride, dynamische oder statische Verfahren mit individueller Kanaluweisung versprechen zwar eine höhere Durchsatzsteigerung, dies geschieht jedoch auf Kosten der Zuverlässigkeit. Denn diese Verfahren können nicht garantieren, dass Teilnehmer in Kommunikationsreichweite einen Funkkanal gemeinsam nutzen und sich direkt erreichen können. Verwendet ein Teilnehmer einen anderen Funkkanal als sein direkter Nachbar, muss für eine erfolgreiche Datenübertragung zunächst einer der eigenen Netzwerkkarten ein neuer Funkkanal zugewiesen werden.

Zusammengefasst sollen also proaktive, topologie-basierte Hop-by-Hop-Routingverfahren entwickelt werden. Zur Gewährleistung der bestmöglichen Konnektivität werden jedem Teilnehmer die gleichen Funkkanäle fest zugewiesen. Die auf diese Weise bereitgestellte Redundanz stellt eine solide Basis für den Einsatz übergeordneter Verfahren dar, welche den Anwendungen tatsächlich Ausfallsicherheit zusichern können.

3.1.2 Verarbeitung von Paketen

Ein Link-State-Verfahren sieht die Nutzung von drei verschiedenen Paket-Typen vor:

- lokale Broadcast-Pakete zur Ermittlung der Nachbarschaften
- Flut-Pakete zum Propagieren der Nachbarschaften
- Unicast-Pakete für den Transport von Nutzdaten

3 Konzept

Für jeden der Paket-Typen soll im Folgenden der Einsatz im Mehrkanal-Betrieb erörtert werden.

Aufgrund der unterschiedlichen Ausbreitungseigenschaften verschiedener Funkkanäle muss davon ausgegangen werden, dass sich die funkkanal-spezifischen Nachbarschaften eines Teilnehmers unterscheiden. Hinzu kommt die Tatsache, dass sich Teilnehmer mit zu wenig Netzwerkkarten ohnehin nicht auf allen Funkkanälen gleichzeitig aufhalten können (detaillierte Erläuterungen gibt Abschnitt 3.1.3). Deswegen muss der Aufbau der Nachbarschaften, d.h. das Senden lokaler Broadcast-Pakete, separat pro Funkkanal erfolgen. Jeder Teilnehmer sendet dazu periodisch Broadcast-Pakete (*Hello-Pakete*) auf jedem Funkkanal aus. Die Auswertung der empfangenen Nachrichten lässt Rückschlüsse darauf zu, welche anderen Teilnehmer sich in der Nachbarschaft befinden und über welche Funkkanäle sie erreichbar sind.

Die funkkanal-spezifischen Nachbarschaften werden zusammengeführt und in einem *Topologie-Paket* gemeinsam periodisch durch das Netzwerk geflutet. Topologie-Pakete werden stets über alle Funkkanäle verschickt. So wird sichergestellt, dass alle Teilnehmer erreicht werden, selbst wenn sie mit dem Sender keinen Funkkanal gemeinsam haben. Als Konsequenz wird jeder Teilnehmer, der mehr als einen Kanal nutzt, Duplikate erhalten. Die Duplikaterkennung erfolgt durch den Einsatz von Sequenznummern. Nutzdaten werden als Unicast-Pakete verschickt. Ihr Versand kann in zwei verschiedenen Modi erfolgen, einem *effizienten Modus* oder einem *zuverlässigen Modus*. Dieser Modus soll von der Anwendung je nach Anforderung zur Laufzeit verändert werden können. Er muss im Header des Datenpakets gespeichert werden, damit Zwischenknoten wissen, auf welche Weise das Paket weitergeleitet werden soll.

Effizienter Modus

Im effizienten Modus sollen die redundanten Kommunikationskanäle im Normalbetrieb zur Steigerung des Durchsatzes eingesetzt werden, indem die einzelnen Abschnitte eines Multi-Hop-Datenstroms verschiedene Funkkanäle benutzen. Bei einem Kanalausfall sollen die betroffenen Datenströme auf die restlichen Funkkanäle aufgeteilt werden, so dass sie sich die zur Verfügung gestellte Bandbreite mit den ursprünglich unbetroffenen Datenströmen teilen müssen.

Effiziente Nutzung der Kanäle In Abschnitt 2.4 wurde erläutert, dass Mehrkanal-Netzwerke weitere Optimierungen des Pfades ermöglichen. Dabei gelten als Optimalitätskriterien:

1. Existiert ein beliebiger Pfad zwischen zwei Teilnehmern, so wird er vom Routing-Algorithmus gefunden.
2. Das Medium wird optimal ausgelastet, indem die Übertragungen gleichmäßig auf die verfügbaren Funkkanäle aufgeteilt werden.

Das erste Optimalitätskriterium bedeutet, dass ein Wechsel des Kanals auf jedem Hop erlaubt ist. Damit gelten Verfahren, die wie MK-AWDS die Kanalnutzung entlang eines Pfades einschränken, als nicht optimal.

Durch das zweite Optimalitätskriterium sollen die Wartezeiten auf ein freies Medium verringert und der Durchsatz erhöht werden. Wie bereits gezeigt wurde, ist der Aufwand für die Ermittlung optimaler Pfade jedoch exponentiell. Ein sinnvoller Ausweg besteht in der Beschleunigung der Pfadsuche, wobei Abzüge in der Optimalität in Kauf genommen werden müssen.

Während einer Pfadsuche gibt es mehrere Stellen, an denen dieser Kompromiss aus Berechnungsdauer und Optimalität eingegangen werden kann. Eine Variante ist es, Kanalwechsel entlang des Pfades durch eine neue Metrik und eine geeignete Anpassung der Topologie zu begünstigen. Das Verfahren der diagonalen Links wählt diesen Ansatz. Es wird in Abschnitt 3.2 beschrieben.

Eine andere Möglichkeit besteht in der Anpassung der Pfadsuche selbst. Diese ist deshalb so aufwändig, weil der Suchraum exponentiell mit der Größe der Topologie wächst. Durch eine Einschränkung des Suchraums im Vorfeld auf einige wenige Alternativen kann die Dauer der Pfadsuche jedoch deutlich verringert werden. Dieser Ansatz wird von einem heuristischen Verfahren verfolgt, das in Abschnitt 3.3 vorgestellt wird.

Durchsatzgewinn Die Kapazität des Netzwerks ist abhängig von der Anzahl der verfügbaren Funkkanäle. Da ein Verfahren mit identischer Kanalzuweisung eingesetzt werden soll, ist die Anzahl der Funkkanäle durch die Anzahl der Netzwerkkarten pro Teilnehmer beschränkt. Diese Einschränkung bedingt lediglich einen moderaten Ausbau der Kapazität, falls nicht auf Spezialhardware zurückgegriffen wird. Die Kapazität ist gleichzeitig eine obere Schranke für den maximal erreichbaren Durchsatz. Dieser fällt typischerweise geringer aus, weil zusätzlich zu den Nutzdaten auch Verwaltungsinformationen ausgetauscht werden müssen. Weitere Einschränkungen entstehen durch geringe Pfadlängen. Weil auf jedem Hop des Pfades nur ein einziger Funkkanal genutzt wird, lässt sich die Kapazität aller Funkkanäle erst ab einer Mindestanzahl an Hops ausreizen. Auf kürzeren Pfaden kann nur ein Bruchteil der verfügbaren Kapazität erreicht werden. Zum Beispiel kann eine Übertragung über zwei Hops maximal zwei Funkkanäle nutzen, selbst wenn das Netzwerk bis zu vier Kanäle unterstützt. Dadurch kann der zu erwartende Durchsatz maximal 50% der Gesamtkapazität des Netzwerks erreichen.

Zuverlässiger Modus

Im zuverlässigen Modus verschickt jeder Teilnehmer entlang des Pfades eine Kopie des Paketes auf allen Funkkanälen, die er mit dem nächsten Hop teilt. Sobald die erste Kopie im nächsten Hop eintrifft, sendet dieser sie auf allen Funkkanälen an dessen

3 Konzept

nächsten Hop usw. Alle weiteren empfangenen Kopien werden anhand der Sequenznummer des Pakets erkannt und verworfen.

Sind benachbarte Teilnehmer entlang des Pfades vollständig miteinander verbunden, konkurrieren sie auf allen Funkkanälen miteinander, so dass keine Durchsatzsteigerungen zu erwarten sind. Bei der Pfadwahl kommt es daher lediglich darauf an, dass die Summe aller Abschnitte unbeachtet der Kanalnutzung möglichst kurz ist. Deswegen genügt es, ein Shortest-Path-Verfahren zur Pfadsuche einzusetzen.

Der Vorteil des zuverlässigen Modus besteht darin, dass einige Pakete verloren gehen können, ohne die laufende Datenübertragung zu beeinflussen. So ist selbst bei einem Kanalausfall keine Unterbrechung der Datenströme zu erwarten, wenn auf allen Abschnitten des Pfades alternative Verbindungen bestehen.

3.1.3 Teilnehmer mit zu wenig Netzwerkkarten

Obwohl die Nutzung aller verfügbaren Funkkanäle die beste Anbindung an das Netzwerk verspricht, sollen dennoch auch Teilnehmer mit weniger Netzwerkkarten unterstützt werden. Denn besonders die Endgeräte des Netzwerks, wie Laptops oder PDAs, werden gewöhnlich mit einer einzigen Netzwerkkarte ausgestattet, lassen sich aufgrund ihrer kompakten Bauweise jedoch nur schwer erweitern. Seitens der Routing-schicht sind für solche Teilnehmer keine speziellen Anpassungen nötig, weil die fehlende Unterstützung eines Funkkanals wie die Nichterreichbarkeit auf diesem Kanal behandelt wird. Im Topologie-Graphen besitzt dieser Teilnehmer statt vieler paralleler Kanten zu seinen Nachbarn entsprechend weniger. Durch die geringere Redundanz wird allerdings die Zuverlässigkeit eingeschränkt, weswegen nach Möglichkeit stets Teilnehmer mit ausreichend Netzwerkkarten eingesetzt werden sollten.

Teilnehmer mit zu wenig Netzwerkkarten haben gewisse Freiheiten bei der Zuweisung der Funkkanäle. Sie können sich für diejenigen Funkkanäle entscheiden, welche die bestmögliche Konnektivität bieten. Dazu beobachten sie zunächst passiv die Nachbarschaften auf allen Funkkanälen und wählen anschließend die besten Funkkanäle aus. Die genaue Vorgehensweise ist wie folgt spezifiziert:

1. Wechsle auf den ersten Funkkanal.
2. Zähle die Anzahl der Nachbarn durch Auswertung ihrer Hello-Pakete. Sende jedoch keine eigenen Hello-Pakete, um unnötige Veränderungen der Topologie zu vermeiden.
3. Solange weitere Funkkanäle verfügbar sind, wechsle auf den nächsten Funkkanal und gehe zu Schritt 2.
4. Weise den Netzwerkkarten in absteigender Reihenfolge diejenigen Funkkanäle zu, über welche die meisten Nachbarn erreicht werden können.

Soll eine Zuweisung für mehrere Netzwerkkarten erfolgen, lässt sich die Ausführung der Schritte 2 bis 3 parallelisieren.

Die Suche nach der optimalen Zuweisung der Funkkanäle erfolgt stets nach dem Neustart des Knotens. Da sich die Topologie auch während des Betriebs ändern kann, soll auch zur Laufzeit eine Änderung der Zuweisung möglich sein. Während ein Teilnehmer eine seiner Netzwerkkarten zur Suche nach einem neuen Funkkanal einsetzt, kann er diese nicht nutzen, um an der Kommunikation im Netzwerk teilzunehmen. Dadurch werden alle Verbindungen zu den Nachbarn auf dem Funkkanal getrennt, welcher der Netzwerkkarte zuvor zugewiesen war. Bricht dabei die Verbindung eines Teilnehmers zu einem Nachbarn komplett ab, muss ein neuer Pfad gesucht werden, der jedoch schlechter ist als der bisherige. Im schlimmsten Fall kann der Nachbar gar nicht mehr erreicht werden. Eine wesentliche Voraussetzung der Kanalsuche ist daher, dass alle Nachbarn des Teilnehmers über die restlichen, weiterhin aktiv genutzten Kanäle erreichbar bleiben.

Teilnehmer mit nur einer Netzwerkkarte müssen sich jedoch während einer Kanalsuche zwangsläufig vom Netzwerk trennen. Sie sollten daher nur dann eine neue Suche einleiten, wenn sie über den bestehenden Kanal ohnehin keine Verbindung zum restlichen Netzwerk haben.

Nachdem die Funkkanäle zugewiesen wurden, kann der normale Betrieb aufgenommen werden. Der Teilnehmer beginnt mit dem Senden von Hello-Paketen auf jeder Netzwerkkarte und gliedert sich damit in die bestehende Topologie ein.

3.2 Diagonale Links

Wir betrachten eine Datenübertragung von Teilnehmer A an Teilnehmer B. Sofern keine anderen Teilnehmer gleichzeitig senden wollen, hat Teilnehmer A kontinuierlichen Zugriff auf das Medium und kann ununterbrochen senden. Leitet Teilnehmer B diese Daten allerdings unter Beibehaltung des Funkkanals an einen dritten Teilnehmer weiter, so teilen sich A und B das Medium, wodurch der Durchsatz auf etwa die Hälfte einbricht. Eine Weiterleitung auf einem anderen Funkkanal stört die Übertragung von Teilnehmer A dagegen nicht, so dass keine Durchsatzeinbußen auftreten. So ist es naheliegend, ein Routingverfahren zu entwickeln, das auf jedem Hop einen Wechsel des Funkkanals durchführt.

3.2.1 Topologie

In Einkanal-Netzwerken haben sich Shortest-Path-Verfahren für die Suche nach optimalen Pfaden durchgesetzt. Je nach Optimalitätskriterium werden dabei verschiedene Metriken eingesetzt, die den einzelnen Verbindungen entsprechende Kosten zuweisen. In Mehrkanal-Routing kann ebenfalls ein Shortest-Path-Verfahren eingesetzt werden, wenn eine angepasste Metrik genutzt wird. Zusätzlich ist jedoch eine geeignete Modellierung der Topologie notwendig.

3 Konzept

Die Modellierung einer Mehrkanal-Topologie als Graph mit parallelen Kanten (Multigraph) ist nicht zweckmäßig, weil Shortest-Path-Verfahren stets die kürzeste der parallelen Kanten auswählen würden. Die Forderung nach Kanalwechseln wird dabei nicht zwangsläufig umgesetzt.

Im ersten Schritt müssen daher die parallelen Kanten im Netzwerk eliminiert werden. Hilfreich ist dabei die Tatsache, dass sich Signale auf unterschiedlichen Funkkanälen nicht beeinflussen. Damit ist es ohne Weiteres möglich, aus der gesamten Topologie jeweils eine Sub-Topologie pro Funkkanal abzuleiten. Ein Knoten in einer Sub-Topologie entspricht dann nicht mehr einem Teilnehmer als Ganzes, sondern jeweils einer seiner Netzwerkkarten, die auf dem entsprechenden Kanal funkt (Abb. 3.1). Die Modellierung eines Teilnehmers als eine Menge eigenständiger Knoten in der Topologie erfordert die Verwaltung einer eigenen Routingtabelle in jedem Knoten. Das bedeutet, dass ein Teilnehmer für jeden Funkkanal eine eigene Pfadsuche durchführen muss, mit dem jeweils auf diesem Kanal operierenden Knoten als Ausgangspunkt.

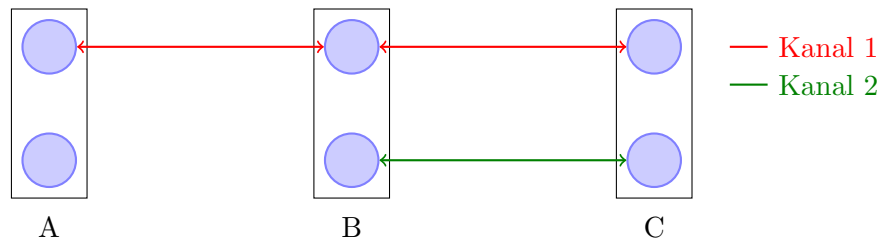


Abbildung 3.1: Sub-Topologien in einem Netzwerk mit zwei Funkkanälen.

Um den Wechsel des Funkkanals auf einer Route zu ermöglichen, ist in jedem Knoten der Wechsel in eine andere Sub-Topologie erlaubt. Kanalwechsel sollen sogar bevorzugt werden, um konkurrierende Zugriffe auf das Funkmedium zu verhindern. Um keine Änderungen an der Pfadsuche vornehmen zu müssen, sollen Kanalwechsel auf Topologie-Ebene als *virtuelle Links* repräsentiert werden. Virtuelle Links verbinden die einzelnen Sub-Topologien untereinander. Sie sind virtuell, weil sie Knoten miteinander verbinden, die sich aufgrund der Nutzung verschiedener Funkkanäle nicht erreichen können. Bei einem Datenaustausch über einen virtuellen Link muss daher entsprechend zwischen den beteiligten Funkkanälen vermittelt werden. Durch die Wahl geeigneter Kosten sollen virtuelle Links bei der Pfadwahl bevorzugt werden. Denn auf einer Route mit vielen virtuellen Links finden häufig Kanalwechsel statt, was in einer Steigerung des Durchsatzes resultiert.

Da ein Kanalwechsel nur innerhalb eines Teilnehmers stattfinden kann, ist der offensichtlichste Einsatz virtueller Links die Verbindung der einzelnen Knoten eines Teilnehmers (Abb. 3.2). Diese virtuellen Verbindungen werden in Anlehnung an die Darstellung in der Abbildung im Folgenden als *vertikale Links* bezeichnet. Da das Senden von Daten über einen vertikalen Link innerhalb des Teilnehmers erfolgt, müssen sie lediglich an den jeweiligen Ziel-Knoten übergeben werden. Pfade mit vertikalen Links sind länger als reale Pfade und zunächst entsprechend teurer. Um dennoch Kanalwechsel

zu begünstigen, müssen die Kosten eines Pfades mit vertikalen Links gegenüber realen Pfaden gesenkt werden. Dies ist nur möglich, wenn der vertikale Link negative Kosten erhält. Denn selbst bei einem Wert von 0 sind beide Pfade gleich teuer, wodurch noch kein Anreiz zum Wechseln besteht. Die Zuweisung negativer Werte erzeugt jedoch negative Zyklen, mit denen Shortest-Path-Verfahren nicht umgehen können. Die Verwendung vertikaler Links führt daher nicht zum Ziel.

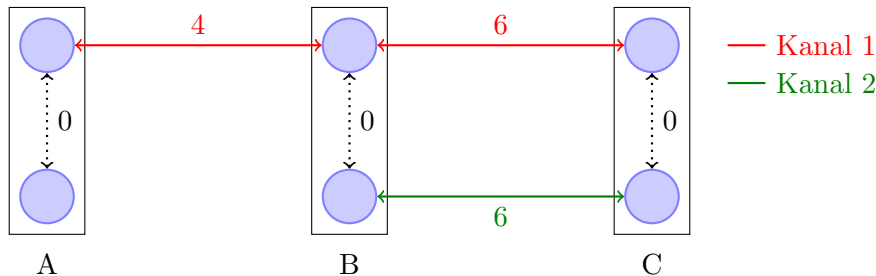


Abbildung 3.2: Topologie mit vertikalen Links (gestrichelte Pfeile)

Damit ein Pfad mit virtuellen Links nicht länger als andere Pfade ist, muss dessen Benutzung einen normalen Link ersetzen. Virtuelle Links müssen demnach einen Teilnehmer mit seinen Nachbarn verbinden (Abb. 3.3). Weil sie dabei Knoten auf unterschiedlichen Funkkanälen verbinden, werden sie als *diagonale Links* bezeichnet. Der Datenaustausch über einen diagonalen Link erfolgt durch das Wechseln des Kanals im Sender und das anschließende Versenden über die entsprechende reale Verbindung.

Um Pfade mit diagonalen Links zu bevorzugen, müssen diagonale Links geringere Kosten aufweisen als ihre entsprechenden realen Links. Dazu soll die gegenseitige Beeinflussung der Teilnehmer entlang des Pfades berücksichtigt werden, die durch das Weiterleiten von Paketen auf dem gleichen Funkkanal entsteht.

Es soll von einem Netzwerk ausgegangen werden, in dem die Interferenzreichweite der Teilnehmer nur unwesentlich größer als die Übertragungreichweite ist. Dann führt das Weiterleiten von Paketen auf dem gleichen Funkkanal zu einer Blockierung des Mediums für den Vorgänger-Knoten, so dass er für die Dauer der Übertragung nicht selbst senden kann. Das Weiterleiten auf einem anderen Funkkanal hingegen blockiert das Medium für den Vorgänger nicht, so dass dieser ununterbrochen senden kann. Durch den Wechsel des Funkkanals können daher in gleicher Zeit doppelt so viele Pakete verschickt werden. Für die Modellierung der diagonalen Links bedeutet dies, dass sie nur die halben Kosten ihres entsprechenden realen Links aufweisen. Dadurch werden sie während der Pfadsuche gegenüber realen Verbindungen bevorzugt, ihre Kosten sind trotzdem nicht negativ. Diagonale Links zwischen zwei Knoten sind stets unidirektional, weil sie unterschiedlichen realen Links zugeordnet sind und dadurch unterschiedliche Kosten in die verschiedenen Richtungen entstehen.

Die Erhöhung der Interferenzreichweite auf ein Vielfaches der Übertragungreichweite ist prinzipiell möglich, allerdings mit einem stark erhöhten Mehraufwand verbunden. Beträgt die Interferenzreichweite das doppelte der Übertragungreichweite, blo-

3 Konzept

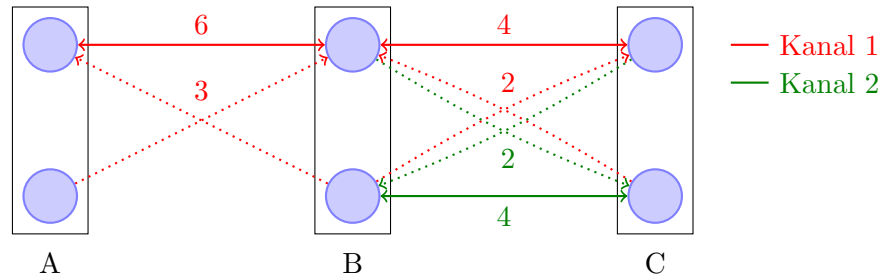


Abbildung 3.3: Topologie mit diagonalen Links (gestrichelte Pfeile)

ektiert das Senden eines Pakets auf dem gleichen Kanal zusätzlich die Übertragungen des vorletzten Knotens. Die Modellierung der Wartekosten auf ein freies Medium auf Topologie-Ebene würde erfordern, dass virtuelle Links nicht nur zwischen Nachbarn aufgebaut werden müssten, sondern auch zwischen Teilnehmern, die zwei Hops entfernt sind. Dazu müsste für jede Kombination aus diagonalen und realen 1-Hop-Links ein eigener virtueller 2-Hop-Link eingeführt werden. Da ein Teilnehmer gewöhnlich sehr viele 2-Hop-Nachbarn hat, bedeutet dies einen starken Anstieg der Anzahl der Kanten in der Topologie, und dadurch einen erhöhten Mehraufwand bei der Pfadsuche. Ein weiteres Kriterium ist die Kenntnis des bisherigen Pfades. Ein Teilnehmer muss den bisher durchlaufenen Pfad eines Paketes innerhalb seines Interferenzbereichs kennen, um den optimalen Sendekanal auszuwählen. Bei einer Interferenzreichweite von einem Hop kann der Funkkanal der letzten Übertragung über denjenigen eigenen Knoten hergeleitet werden, der das Paket empfangen hat. Bei einer Interferenzreichweite von zwei Hops muss hingegen zusätzlich der Funkkanal der vorletzten Übertragung bekannt sein. Dieser kann nicht hergeleitet werden und bedarf daher einer Speicherung im Paket. Um den Mehraufwand bei der Modellierung der Topologie und die Erzeugung zusätzlichen Overheads beim Paketaufbau zu vermeiden, sollen Interferenzreichweiten von einem Vielfachen der Übertragungreichweite im Folgenden nicht weiter betrachtet werden. Diagonale Links verbinden damit lediglich direkte Nachbarn.

LIBRA modelliert Kanalwechsel ebenfalls auf Topologie-Ebene, es sieht ebenfalls die Auftrennung eines Teilnehmers in mehrere Knoten und die Verbindung der Knoten verschiedener Funkkanäle durch virtuelle Verbindungen vor. Virtuelle Links in LIBRA verlaufen jedoch innerhalb des Teilnehmers und erhalten fixe Kosten zugewiesen. Dadurch entsteht der Nachteil, dass bei hohen Kosten der realen Links der Unterschied zwischen einem Kanalwechsel und der Beibehaltung des Kanals nicht mehr ins Gewicht fällt und demzufolge eine Beibehaltung des Kanals wahrscheinlich ist. Das Verfahren der diagonalen Links sieht hingegen die direkte Abhängigkeit der Kosten diagonalen und realer Links vor, so dass selbst bei hohen realen Kosten Kanalwechsel bevorzugt werden.

3.2.2 Aufwandsabschätzung

Das Konzept der diagonalen Links erfordert gegenüber einer Modellierung der Topologie in Einkanal-Netzwerken die Hinzunahme neuer Knoten und Kanten. Für die nachfolgende Schätzung des erforderlichen Mehraufwandes gelten folgende Parameter:

k die Anzahl der genutzten Funkkanäle

i die Anzahl der Knoten bzw. genutzten Netzwerkkarten eines Teilnehmers

j die Anzahl der realen Verbindungen eines Teilnehmers zu einem Nachbarn

Offensichtlich gilt $j \leq i \leq k$. Der Aufwand ist maximal, wenn die Konnektivität der Teilnehmer maximal ist, d.h. jeder Teilnehmer ausreichend Netzwerkkarten besitzt und mit seinen Nachbarn auf allen Funkkanälen verbunden ist. Dann gilt $j = i = k$.

Anzahl der Knoten

Durch die Auftrennung der Topologie in mehrere Sub-Topologien wird die Anzahl der Knoten vervielfacht. Jeder Teilnehmer wird nicht mehr als ein Knoten, sondern als eine Menge von i Knoten modelliert. Da ein Teilnehmer für jeden Knoten eine separate Pfadsuche durchführen muss, bestimmt die Anzahl der Knoten außerdem die Anzahl der nötigen Pfadberechnungen. Besitzen alle Teilnehmer ausreichend Netzwerkkarten, wächst die Anzahl der Knoten linear mit der Anzahl der Funkkanäle, wodurch keine größeren Einschränkungen in der Realisierbarkeit der diagonalen Links entstehen.

Anzahl der Kanten

Für jede reale Verbindung werden von allen unbeteiligten Startknoten diagonale Links zum Zielknoten erzeugt. Deren Anzahl pro realer Verbindung $z_{diagonal}$ entspricht demnach der Formel

$$z_{diagonal} = j(i - 1)$$

Die Gesamtzahl der Verbindungen zu diesem Nachbarn z_{gesamt} setzt sich aus der Anzahl der realen und diagonalen Verbindungen zusammen:

$$\begin{aligned} z_{gesamt} &= j + z_{diagonal} \\ &= j + j(i - 1) \\ &= i \cdot j \end{aligned} \tag{3.1}$$

3 Konzept

Bei maximaler Konnektivität gilt die Gleichheit $j = i = k$, so dass Gleichung (3.1) zu

$$z_{gesamt} = k^2$$

vereinfacht werden kann.

Während in Einkanal-Topologien stets nur eine einzige Verbindung zwischen zwei Teilnehmern besteht, wächst deren Anzahl durch die Nutzung diagonalen Links quadratisch mit der Anzahl der eingesetzten Funkkanäle. Da die Anzahl der Netzwerkkarten pro Teilnehmer, und damit die Anzahl der nutzbaren Funkkanäle, relativ klein ist, sollte auch hier der durch die diagonalen Links erzeugte Mehraufwand noch im Rahmen der praktikablen Durchführbarkeit liegen.

3.2.3 Gruppierung von Knoten

Jeder Teilnehmer besteht aus einer Menge von Knoten. Beim Versand von Unicast-Paketen gibt es daher mehrere potentielle Empfänger, nämlich genau die zum Ziel-Teilnehmer gehörenden Knoten.

Sollen Daten an einen bestimmten Teilnehmer gesendet werden, so ist es prinzipiell irrelevant, welcher der Zielknoten als tatsächliches Ziel angegeben wird. In jedem Fall treffen die Daten im Ziel-Teilnehmer ein und können dort weiter verarbeitet werden. In der Regel sind die Pfadkosten zu den einzelnen Zielknoten jedoch verschieden. Das Routingverfahren ermittelt daher den besten Pfad, indem es die Pfadkosten zu allen Zielknoten miteinander vergleicht und den günstigsten auswählt.

Diese Zuordnung zwischen Knoten und Teilnehmern muss im gesamten Netzwerk bekannt sein, weil jeder Teilnehmer prinzipiell Sender und Empfänger sein kann.

Jeder Teilnehmer erhält dazu eine im Netzwerk eindeutige Identifikations-Nummer (ID), welche all seinen Knoten zugewiesen wird. Dies ist die MAC-Adresse der Netzwerkkarte seines ersten Knotens. Durch die Propagierung der Teilnehmer-ID macht jeder Knoten seine Zugehörigkeit dem gesamten Netzwerk bekannt. Alle Teilnehmer verwalten eine Tabelle von IDs und den zugehörigen Knoten. Auf diese Weise wissen sie, aus welchen Knoten ein anderer Teilnehmer besteht und zu welchem Teilnehmer ein bestimmter Knoten gehört.

3.2.4 Optimalität des Pfades

Das Konzept der diagonalen Links ist ein Kompromiss aus Dauer der Padsuche und Optimalität des Pfades. Besonders die Annahme, dass die Interferenzreichweite lediglich einen Hop beträgt, führt dazu, dass stets nur zwei Teilnehmer um das Medium konkurrieren. Dadurch wird die maximale Durchsatzsteigerung bereits bei der Nutzung von zwei Funkkanälen erreicht, die Hinzunahme weiterer Funkkanäle bringt theoretisch keine Vorteile.

Messungen haben jedoch gezeigt, dass die Interferenzreichweite mindestens das Doppelte der Übertragungreichweite beträgt [Wew06], weshalb der Einsatz weiterer Funkkanäle sinnvoll sein kann. Durch die fehlende Betrachtung der bisherigen Pfade arbeitet das Verfahren der diagonalen Links dann allerdings unzuverlässig. So besteht die Möglichkeit, dass ein Pfad lediglich zwischen zwei verschiedenen Funkkanälen alterniert und alle weiteren Kanäle nicht nutzt.

Das Verfahren der diagonalen Links betrachtet nicht das Vorhandensein weiterer Datenströme in der Nachbarschaft. Es ist somit möglich, dass Pakete in stark ausgelastete Bereiche des Netzwerks geleitet werden und dort Verzögerungen hervorrufen.

3.3 Heuristik

Sollen stets optimale Pfade gefunden werden, so müssen alle Kombinationen der Einzelverbindungen bis zum Ziel untersucht und miteinander verglichen werden. Da der Aufwand eines solchen Brute-Force-Ansatzes exponentiell mit der Anzahl der Links steigt, ist die effiziente Nutzung insbesondere bei größeren Netzwerken fraglich. Das im Folgenden vorgestellte heuristische Verfahren besteht daher aus zwei Schritten. Im ersten Schritt wird der Suchraum reduziert, indem zunächst ein nach Latenz optimierter Pfad bis zum Empfänger gesucht wird. Die darin enthaltenen Teilnehmer und ihre Verbindungen untereinander bilden eine neue Topologie, die an eine Brute-Force-Pfadsuche übergeben wird.

3.3.1 Vorauswahl

Das Ziel der Vorauswahl ist die Reduzierung der Knoten- und Kantenmenge für die anschließende Brute-Force-Suche. Sie fungiert damit als Filter, indem sie schlechte Alternativen bereits im Vorfeld aussortiert. An die Vorauswahl werden zwei Anforderungen gestellt. Einerseits muss sie performant sein, um die in der eingeschränkten Brute-Force-Suche erzielte Zeitersparnis nicht wieder aufzuheben. Andererseits sollte sie zuverlässig sein, d.h. gute Pfade zurückliefern und schlechte Pfade verwerfen. Damit im Brute-Force-Schritt der optimale Pfad gefunden wird, muss dieser bereits Teil der Vorauswahl sein.

Das Ergebnis der Vorauswahl soll für die weitere Verwendung in der Brute-Force-Suche geeignet sein. Sie beschränkt daher die Topologie derart, dass lediglich ein einzelner Pfad von Teilnehmern übrig bleibt, wobei das erste Element der eigene Teilnehmer und das letzte Element das Ziel ist. Ein Durchlaufen des Teilnehmer-Pfades gibt damit die Reihenfolge der Teilnehmer an, die das Paket passieren muss. Alle übrigen Teilnehmer und irrelevanten Links werden entfernt (Abb. 3.4).

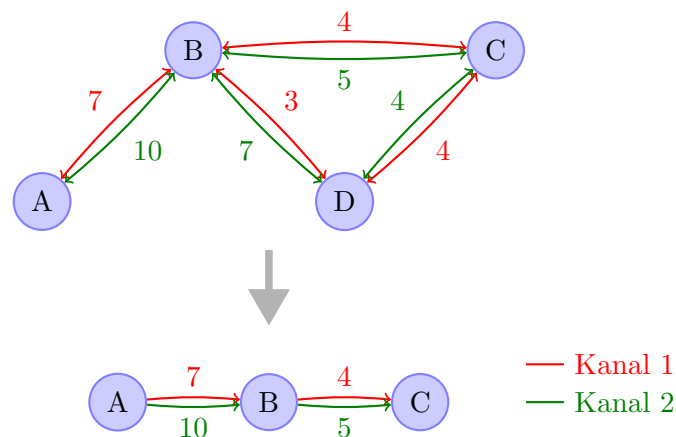


Abbildung 3.4: Möchte Teilnehmer A Daten zu C senden, reduziert er zunächst das Netzwerk auf einen einzelnen Teilnehmer-Pfad zum Ziel. Dieser Schritt dient als Vorstufe für die eigentliche Pfadsuche.

Für die konkrete Umsetzung der Vorauswahl sind viele Verfahren denkbar. Beispielsweise besteht die Möglichkeit, ein Shortest-Path-Verfahren auf dem ursprünglichen Topologie-Graphen mit parallelen Kanten einzusetzen. Dies ist zwar sehr effizient, aber nicht sonderlich effektiv, da auf diese Weise die gegenseitige Beeinflussung der Teilnehmern durch die Nutzung gleicher Funkkanäle nicht betrachtet wird. Vielversprechender ist die Nutzung einer Pfadsuche, die Kanalwechsel entlang des Pfades favorisiert (z.B. das Verfahren der diagonalen Links oder LIBRA [YWK]). Sie ermittelt insgesamt bessere Pfade als Verfahren, die Kanalwechsel nicht berücksichtigen. Somit steigt auch die Wahrscheinlichkeit, dass sich darunter der optimale Pfad befindet.

3.3.2 Brute-Force-Schritt

Auf der reduzierten Menge von Teilnehmern werden nun alle Kanalkombinationen bis zum Ziel untersucht, von denen die beste ausgewählt wird. Dazu wird ein Entscheidungsbaum aufgebaut, dessen Wurzel der eigene Teilnehmer bildet (Abb. 3.5). Jeder weitere Teilnehmer entlang des Pfades bildet eine neue Ebene, die Blätter des Baumes stellen somit die Knoten des Ziel-Teilnehmers dar. Jeder Ast entspricht einer Verbindung von einem Teilnehmer zu seinem Nachfolger auf einem einzelnen Funkkanal. Auf diese Weise repräsentiert der Weg von der Wurzel zu einem Blatt einen einzelnen Pfad vom Sender zum Empfänger.

Der Entscheidungsbaum besitzt die folgenden Eigenschaften:

1. Alle Knoten einer Ebene repräsentieren denselben Teilnehmer.

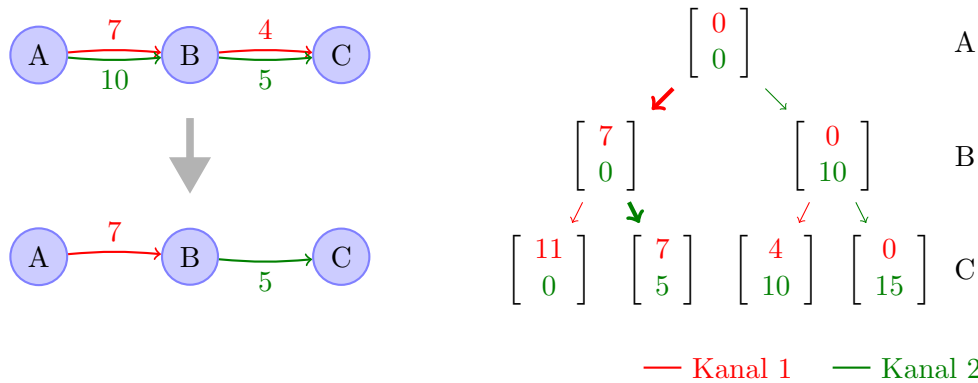


Abbildung 3.5: Im Brute-Force-Schritt wird auf dem vorselektierten Graphen der optimale Pfad bestimmt (links). Dazu werden alle Kanalkombinationen in einen Entscheidungsbaum eingetragen und miteinander verglichen (rechts).

2. Die n -te Ebene enthält alle Kanalkombinationen bis zum n -ten Teilnehmer des vorausgewählten Teilnehmer-Pfades.
3. Jeder Knoten einer Ebene hat die identische Anzahl an Kindern.

Eigenschaft 1 wird durch die Konstruktion des Baumes erfüllt. Eigenschaft 2 kann von Eigenschaft 1 hergeleitet werden. Es bleibt lediglich zu zeigen, dass tatsächlich alle Kanalkombinationen erfasst werden. Diese Forderung wird durch das einmalige Durchlaufen aller Verbindungen zwischen dem aktuellen Teilnehmer und seinem Nachbarn erfüllt. Der Entscheidungsbaum besitzt Eigenschaft 3, weil die Kinder jedes Knotens im Baum die Kanalkombinationen zum jeweils nächsten Hop enthalten. Durch Eigenschaft 1 gilt dies für alle Knoten einer Ebene gleichermaßen.

Weil sich Signale auf unterschiedlichen Funkkanälen nicht beeinflussen, sind die Kosten unterschiedlicher Kanäle orthogonale Größen. Deswegen werden Pfadkosten vektoriell gespeichert und nach dem zugrunde liegenden Funkkanal differenziert. Jeder Ast wird mit dem vektoriellen Gewicht des entsprechenden Links versehen. Die Gewichte werden beim Durchlaufen des Baumes aufaddiert, so dass an den Blättern die Linkkosten des gesamten Pfades getrennt nach Kanal feststehen. Um die Blätter schließlich miteinander vergleichen zu können, werden die summierten Gewichte mittels des WCETT-Verfahrens [DPZ04] zu einem skalaren Kosten-Wert zusammengefasst. Demnach werden aus den vektoriellen Kosten-Werten zwei Kennwerte berechnet. Man betrachte einen Pfad über n Hops, d_i bezeichne die Kosten des i -ten Hops. Dann entspricht die Latenz L einer Datenübertragung der Summe aller Einzelkosten:

$$L = \sum_{i=1}^n d_i$$

3 Konzept

Die Latenz wächst mit jedem weiteren Hop, der dem Pfad hinzugefügt wird. Sie allein ist allerdings nicht ausreichend, um den Wunsch nach der Nutzung verschiedener Kanäle zu entsprechen. Deswegen wird ein weiterer Kennwert eingeführt, der den Durchsatz repräsentiert. Dazu werden die Kosten einer Multi-Hop-Übertragung getrennt nach ihrem Kanal betrachtet. Die Summe aller Kosten pro Kanal beträgt:

$$X_j = \sum_{\text{Hop } i \text{ nutzt Kanal } k} d_i \quad 1 \leq j \leq k$$

Datenübertragungen auf dem gleichen Kanal stören sich gegenseitig, so dass der Kanal mit den höchsten Gesamtkosten den Flaschenhals der Übertragung bildet. Der Durchsatz D einer Multi-Hop-Übertragung lässt sich demnach mittels

$$D = \max_{1 \leq j \leq k} X_j$$

abschätzen. Unter Zuhilfenahme eines Wichtungsparameters β werden Latenz und Durchsatz nun zu einem skalaren Wert zusammengefasst:

$$K = (1 - \beta) \times \sum_{i=1}^n d_i + \beta \times \max_{1 \leq j \leq k} X_j$$

Es gilt $0 \leq \beta \leq 1$. Mittels β kann die Präferenz zwischen hohem Durchsatz oder geringer Latenz festgelegt werden. K bezeichnet die gewichteten Kosten des Pfades. Dasjenige Blatt mit minimalem K bestimmt schließlich den optimalen Pfad durch die vorausgewählte Topologie. Die Verschmelzung der Kostenwerte kann erst in den Blättern durchgeführt werden, weil erst dann der endgültige Pfad feststeht. Würde β bereits auf jeden Einzelpfad angewandt, gingen die Informationen über den konkreten Funkkanal der Verbindung verloren. Dann wäre keine effektive Lastverteilung mehr möglich.

Die Heuristik übernimmt von der WCETT-Metrik lediglich das Verfahren zur Verschmelzung der Funkkanal-Kosten. Dabei ist es unerheblich, wie die einzelnen Kosten zustande kommen. So ist das heuristische Verfahren nicht auf die Nutzung der ETT-Metrik beschränkt, sondern kann prinzipiell mit jeder anderen Metrik betrieben werden.

Da ein Hop-by-Hop-Verfahren eingesetzt werden soll, muss jeder Teilnehmer den optimalen Pfad zum Ziel selbst berechnen. Dabei besteht die Gefahr, dass die Pfadwahl einer Zwischenstation vom ursprünglich im Sender bestimmten optimalen Pfad abweicht. Der Grund hierfür ist, dass eine Zwischenstation den bisherigen Verlauf des Pakets nicht kennt. Sie kann lediglich die Kosten des letzten Hops anhand der Netzwerkkarte herleiten, die das Paket empfangen hat. Dies ist jedoch nicht ausreichend, um den kompletten bisherigen Pfad zu rekonstruieren. Deswegen wird der bisherige Paketverlauf, inklusive den Kosten des nächsten Hops, im Paket selbst abgespeichert.

Dies geschieht durch eine Zuordnung der jeweiligen Kanalnummer nach dem WLAN-Standard auf die entsprechenden Kosten. Der Empfänger liest den Verlauf aus dem Paket aus und initialisiert damit seinen Entscheidungsbaum. So arbeiten alle Teilnehmer auf dem identischen Entscheidungsbaum, vom Sender ausgehend beginnt jeder Teilnehmer lediglich jeweils eine Ebene tiefer (Abb. 3.6). Demzufolge ist der Aufwand für die Pfadsuche umso geringer, je näher sich der Teilnehmer am Ende des Pfades befindet.

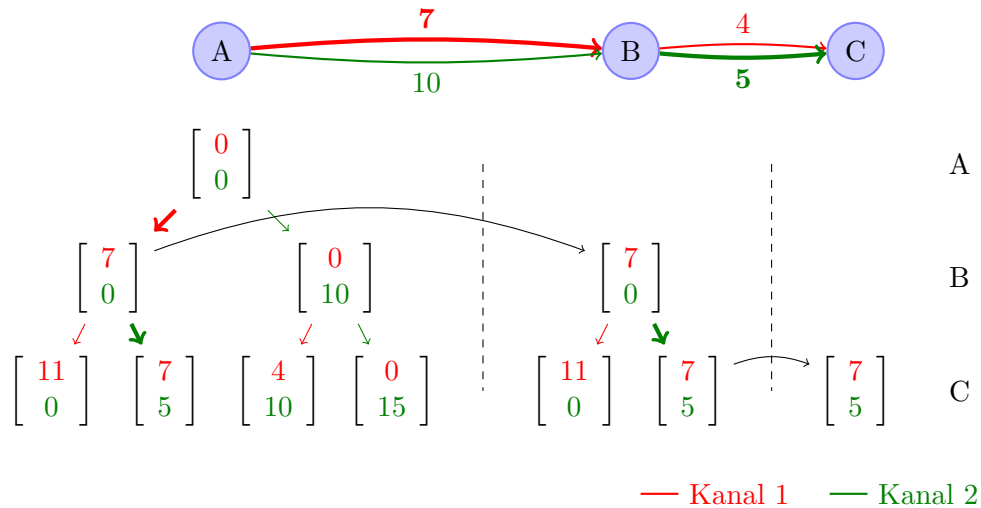


Abbildung 3.6: Die Kosten des bereits durchlaufenen Pfades werden im Paket-Header gespeichert und vom Empfänger ausgewertet (schwarze Pfeile). Sie dienen zur Initialisierung des Entscheidungsbaumes.

Weil die Routenberechnung nun vom bisherigen Paketverlauf abhängig ist, muss dieser in der Routingtabelle vermerkt werden. Die in klassischem Hop-by-Hop-Routing eingesetzte Abbildung

$$\text{Ziel} \mapsto \text{NächsterHop}$$

muss daher zu

$$(\text{Ziel}, \text{Verlauf}) \mapsto \text{NächsterHop}$$

erweitert werden. Aufgrund der prinzipiell unendlichen Anzahl verschiedener Paketverläufe ist die regelmäßige Berechnung aller Pfade nicht mehr möglich. Die Ermittlung einer neuen, speziellen Route soll deswegen nur bei Bedarf erfolgen, nämlich genau dann, wenn ein neues Paket weitergeleitet werden soll und das Tupel aus Ziel und bisherigen Kosten noch nicht in der Routingtabelle steht.

3.3.3 Aufwandsabschätzung

Im heuristischen Verfahren ist der Pfadsuche die Einschränkung des Suchraums vorgeschaltet. Der Gesamtaufwand besteht dadurch aus der Summe des Aufwands der

3 Konzept

Vorauswahl und des Brute-Force-Schrittes. Da das Verfahren für die Vorauswahl frei wählbar ist, kann hierfür keine generelle Aufwandsabschätzung angegeben werden. Wird beispielsweise das Dijkstra-Verfahren eingesetzt, erzeugt die Vorauswahl gegenüber einem Einkanal-Routingverfahren keinen Mehraufwand. Der Mehraufwand des Verfahrens der diagonalen Links wird in Abschnitt 3.2.2 untersucht.

Im Brute-Force-Schritt werden alle verbliebenen Kanalkombinationen der Vorauswahl untersucht. Dazu wird ein Entscheidungsbaum aufgebaut, dessen Knotenzahl von der Pfadlänge und der Anzahl der verfügbaren Funkkanäle abhängt. Beide Parameter sollen daher zur Schätzung des Aufwandes herangezogen werden:

k Anzahl der Funkkanäle des Netzwerks

l Länge des vorausgewählten Teilnehmer-Pfads

Um den optimalen Pfad auf dem vorausgewählten Graphen zu finden, müssen alle Knoten des Baumes durchlaufen werden. Alle Eltern-Knoten müssen besucht werden, um deren Kinder zu erzeugen. Nachdem der Baum fertig aufgebaut ist, befinden sich die einzelnen Kanalkombinationen als Blätter in der untersten Ebene des Baumes. Sie müssen durchlaufen werden, um das Blatt mit den geringsten Kosten zu finden. Die Anzahl aller Knoten des Baumes kann daher als grobe Schätzung für den Aufwand des Brute-Force-Schrittes herangezogen werden. Sie ist maximal, wenn alle Teilnehmer entlang des vorausgewählten Pfades über alle Funkkanäle mit ihren Nachbarn verbunden sind.

Die Anzahl der Knoten des Baumes z_{gesamt} entspricht der Summe der Knoten aller Ebenen:

$$z_{gesamt} = \sum_{i=1}^l z_i$$

Die Größe z_i gibt die Knotenanzahl von Ebene i an, mit $i = 0, 1, \dots, l$.

Sie vervielfacht sich beim Übergang auf die $(i + 1)$ -te Ebene um das k -fache, weil jeder Elternknoten k Kinder besitzt. Demnach gilt:

$$z_i = k^i$$

Dies führt zu

$$z_{gesamt} = \sum_{i=1}^l k^i$$

bzw. nach Anwendung der geometrischen Summenformel zu

$$z_{gesamt} = \frac{k^{l+1} - 1}{k - 1}$$

Der für den Brute-Force-Schritt erforderliche Aufwand wächst demnach exponentiell mit der Pfadlänge. Solange das Netzwerk jedoch räumlich nicht weit ausgedehnt ist,

ist die Pfadlänge l eher gering. Da die Anzahl der Funkkanäle von der Anzahl der Netzwerkkarten pro Teilnehmer abhängt, ist k ebenfalls gering, so dass der Einsatz des heuristischen Verfahrens noch praktikabel ist.

3.3.4 Optimalität des Pfades

Die Vorauswahl hat den größten Einfluss auf die Optimalität des Pfades. Sie soll schneller als die Brute-Force-Suche ablaufen und kann daher nicht das gleiche Verfahren einsetzen. Demzufolge kann die Vorauswahl keine Optimalität garantieren. Sie kann lediglich so gestaltet werden, dass sie eine Menge möglichst guter Pfade findet. Dies muss jedoch nicht bedeuten, dass darin stets der optimale Pfad enthalten ist.

Der Brute-Force-Schritt findet immer einen optimalen Pfad auf dem vorselektierten Graphen, sofern sich alle beteiligten Stationen in Interferenzreichweite befinden. Das liegt daran, dass stets der komplette Pfad bis zum Ziel betrachtet wird. Dies ist jedoch nur dann sinnvoll, wenn sich alle Teilnehmer entlang des Pfades tatsächlich gegenseitig stören können, was nur bei relativ kurzen Pfaden zutrifft. In großen Netzwerken sind einzelne Teilnehmer dagegen mitunter zu weit entfernt, um miteinander um den Medienzugriff zu konkurrieren.

Ähnlich dem Konzept der diagonalen Links optimiert das heuristische Verfahren lediglich die gegenseitige Beeinflussung der Teilnehmer entlang des zu betrachtenden Datenstroms. Es beachtet keine parallelen Datenübertragungen in der Nachbarschaft, die ebenfalls Einfluss auf den Durchsatz haben.

3 Konzept

4 Implementierung

Dieses Kapitel beschreibt die prototypische Implementierung der entwickelten Routing-Verfahren für die Nutzung in AWDS. Nach einer Einführung in AWDS wird beschrieben, wie dieses erweitert werden musste, um prinzipiell mehrere Funkkanäle zu unterstützen. Die beiden folgenden Abschnitte beschäftigen sich mit der konkreten Umsetzung des Verfahrens der diagonalen Links und der Heuristik.

4.1 Einführung in relevante Teile von AWDS

Zum Verständnis der durchgeführten Anpassungen ist es wichtig, sich mit dem Aufbau und der Funktionsweise von AWDS vertraut zu machen. Dazu dient dieser Abschnitt. AWDS wird als C++-Projekt an der Universität Magdeburg in einer Linux-Umgebung entwickelt. Die Implementierung dieser Masterarbeit basiert auf dem Quellcode von AWDS7.

4.1.1 Laufzeitumgebung

Die Evaluierung von Routingprotokollen in realen Netzwerken ist sehr aufwändig, weil die Software auf alle Teilnehmer aufgespielt werden muss und nach Abschluss alle Messergebnisse wieder zusammengeführt werden müssen. Darüber hinaus wird der Vergleich der Messergebnisse dadurch erschwert, dass sich aufgrund schwankender Signalstärken kaum identische Ausgangsbedingungen schaffen lassen. Indem das Netzwerk lediglich auf einem einzelnen PC simuliert wird, lassen sich diese Probleme umgehen. AWDS wurde daher darauf konzipiert, sowohl in realen Betriebssystemen als auch im Netzwerksimulator ns-2 eingesetzt werden zu können. Allerdings besitzt ns-2 eine andere Programmierschnittstelle als übliche Betriebssysteme, was eine Migration schwierig macht.

Zu diesem Zweck nutzt AWDS als Laufzeitumgebung die Generic Event-based API (GEA) [HM05]. GEA kapselt den Zugriff auf die zugrunde liegenden Systemschnittstellen, indem es dem Protokoll eine einheitliche Schnittstelle anbietet (Abb. 4.1). Dadurch wird die Ausführung von AWDS auf unterschiedlichen Systemen ermöglicht, ohne jeweils spezielle Anpassungen vornehmen zu müssen. Momentan werden Betriebssysteme, die auf dem POSIX-Standard (Portable Operating System Interface) basieren, sowie der Netzwerksimulator ns-2 unterstützt.

Anwendung	
Netzwerk-Protokoll (AWDS)	
GEA	
POSIX (Linux)	ns-2

Abbildung 4.1: Systemarchitektur mit GEA.

Das Netzwerk-Protokoll muss als Modul vorliegen, das von GEA zur Laufzeit dynamisch geladen wird. GEA unterstützt die Nutzung beliebig vieler Module, so dass die Funktionalität von AWDS in verschiedene Module ausgegliedert ist, die nacheinander geladen werden. Indem Module ausgetauscht oder ausgelassen werden, lässt sich AWDS sehr flexibel im Funktionsumfang anpassen.

Die Kommunikation der Module untereinander erfolgt durch den Austausch von Objekt-Referenzen. Dazu bietet GEA mit dem *ObjectRepository* eine zentrale Ablage, die den Zugriff auf beliebige Objekte durch einen eindeutigen Bezeichner ermöglicht. Möchte ein Modul Zugriff auf bestimmte Abschnitte gewähren, kapselt es diese durch eine Klasse, legt ein Objekt der Klasse an und speichert es im *ObjectRepository*. Andere Module können das Objekt laden und erhalten damit Zugriff auf dessen Funktionalität.

4.1.2 Module

Im Folgenden werden einige der wichtigsten Module kurz vorgestellt:

shell.so Erlaubt die Manipulation anderer Module über eine Kommandozeile auf Port 8444.

rawbasic.so Dieses Modul kapselt grundlegende Ein- und Ausgabeoperationen auf der WLAN-Schnittstelle. Diese wird als Parameter an das Modul übergeben.

awdsrouting.so Das awdsrouting-Modul verwaltet die Topologie, führt die Routenberechnung durch und übernimmt den Versand von Daten- und Kontrollpaketen.

tapiface2.so Das Modul “tapiface2.so” erzeugt eine virtuelle Netzwerkschnittstelle (Tap-Device). Über diese Schnittstelle werden die Daten zwischen der Applikation und dem AWDS-Netz ausgetauscht.

topowatch.so Verpackt die Topologie-Daten in einen XML-Strom und bietet sie über Port 8333 externen Anwendungen an.

topolock.so Ermöglicht, die aktuelle Topologie zu fixieren, so dass diese durch nachfolgende Topologie-Pakete nicht verändert werden kann. Routing-Informationen bleiben für die gesamte Dauer der Fixierung gültig.

srcfilter.so Ermöglicht die Deaktivierung bestehender Links.

Die zwei letztgenannten Module sind hilfreich, um für Experimente die gleichen Ausgangsbedingungen zu schaffen. Das awdsrouting-Modul enthält das eigentliche Routing-Protokoll. Bis auf rawbasic.so dienen alle weiteren Module dazu, das awdsrouting-Modul im Funktionsumfang zu erweitern. Abbildung 4.2 zeigt in vereinfachter Form die Modulstruktur von AWDS. Die Module oberhalb von awdsrouting stellen lediglich eine Auswahl dar.

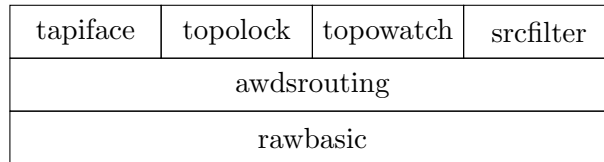


Abbildung 4.2: Vereinfachte Darstellung der Modulstruktur von AWDS.

4.1.3 Das awdsrouting-Modul

Das awdsrouting-Modul besteht im Wesentlichen aus der Klasse AwdsRouting, welche die Kernfunktionen des Routingprotokolls implementiert. Obwohl der Großteil der Funktionalität direkt in die Klasse integriert ist, lässt diese sich in mehrere logische Komponenten gliedern, deren Zusammenwirken in Abb. 4.3 dargestellt wird. Die Funktionsweise der einzelnen Komponenten wird im Folgenden detailliert erläutert.

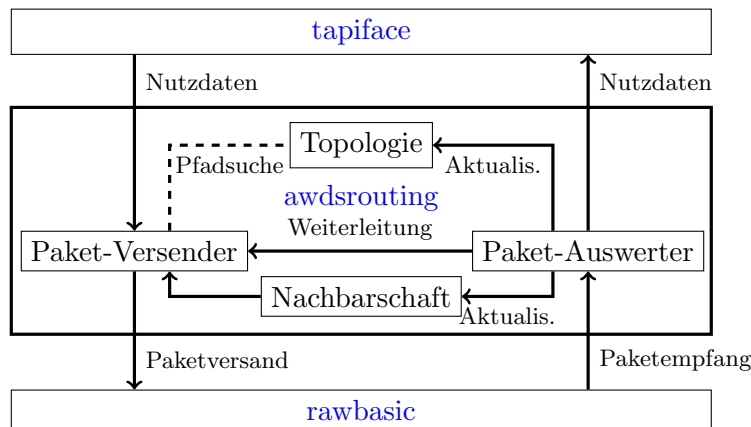


Abbildung 4.3: Logische Komponenten des awdsrouting-Moduls im Zusammenspiel mit dem tapiface- und rawbasic-Modul.

Paket-Auswerter

Beim Starten des Moduls lädt der Paket-Auswerter das rawbasic-Modul aus dem ObjectRepository und registriert in GEA eine Rückruffunktion für den Empfang von Pa-

4 Implementierung

keten auf der im rawbasic-Modul definierten Schnittstelle. Signalisiert GEA den Empfang eines Paketes, wird dieses über das rawbasic-Modul ausgelesen und je nach Typ weiter behandelt. Beim Empfang von Flutpaketen wird anhand einer Sequenznummer und des Absenders überprüft, ob das Paket nicht schon einmal empfangen wurde. Ist dies nicht der Fall, wird sein Inhalt ausgewertet und als neues Flutpaket an den Paket-Versender übergeben. Gewöhnliche Broadcast-Pakete sind die Hello-Nachrichten der Nachbarn. Sie werden lokal ausgewertet und nicht wieder neu verschickt. Bei Unicast-Paketen wird anhand der Zieladresse geprüft, ob sie lokal zugestellt oder zur Weiterleitung an dem Paket-Versender übergeben werden müssen.

Der Paket-Auswerter bietet anderen Modulen die Möglichkeit, sich für spezielle Unicast- oder Flut-Pakettypen zu registrieren. Dazu verwaltet er eine Tabelle, die dem entsprechenden Pakettyp eine Rückruffunktion zuordnet. Wird ein neues Paket mit einem selbst definierten Typ empfangen, so wird die jeweilige Rückruffunktion herausgesucht und aufgerufen. Das tapiface-Modul nutzt diesen Mechanismus, um über den Empfang von Unicast-Paketen benachrichtigt zu werden.

Topologie

Die Verwaltung der lokalen Sicht auf die Topologie ist in AWDS derart umfangreich, dass sie in die separate Klasse `Topology` ausgegliedert wurde.

Die Speicherung der Topologie erfolgt in Form einer *Adjazenzliste*, d.h. einer Datenstruktur aus verketteten Listen (Abb. 4.4). Die Basisstruktur bildet eine Liste aller Knoten. Sie ist sortiert, um den effizienten Zugriff auf die Knoten zu ermöglichen. Dies ist insbesondere bei der Pfadsuche wichtig. Jedem Knoten wird ein Objekt der Klasse `NodeDescr` zugeordnet, das alle Informationen über ihn enthält, beispielsweise eine Bezeichnung und eine Auflistung aller ausgehenden Verbindungen. Jede dieser Verbindungen ist ein Objekt der Klasse `LinkQuality`. Es speichert im Wesentlichen den gegenüberliegenden Knoten und das Gewicht der Verbindung. Die Routing-Informationen werden nicht in einer separaten Tabelle gespeichert, sondern liegen innerhalb der Topologie verstreut vor. So enthält die Klasse `NodeDescr` zusätzliche Informationen wie den nächsten Hop, über den der jeweilige Knoten erreicht werden kann.

Als Hop-by-Hop-Routingverfahren speichert AWDS für jedes Ziel nur den nächsten Hop. Das ist derjenige Knoten, an den das Paket weitergeschickt werden muss. Diese Information kann mittels `Topology::getNextHop()` abgefragt werden. Ist die Topologie gültig, wird der nächste Hop ausgelesen und zurückgegeben. Hat sich die Topologie jedoch seit der letzten Pfadsuche verändert, so ist sie als ungültig markiert und alle nächsten Hops müssen neu bestimmt werden. Dazu werden zunächst mittels des Dijkstra-Algorithmus die kürzesten Pfade zu allen Teilnehmern im Netzwerk gesucht. Die nächsten Hops entsprechen jeweils dem ersten Knoten dieser Pfade.

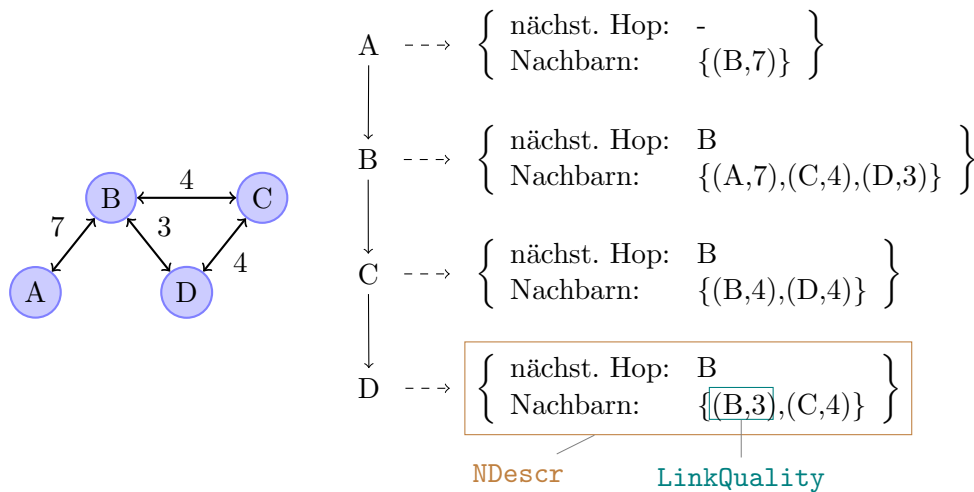


Abbildung 4.4: Vereinfachte Darstellung der lokalen Sicht von Teilnehmer A auf die Topologie.

Nachbarschaft

Die Nachbarschafts-Komponente nimmt vom Paket-Auswerter alle empfangenen Hello-Pakete entgegen und nutzt sie zur Aktualisierung der Nachbarschaft des Teilnehmers. Die Nachbarschaft umfasst die Adressen der Nachbarn sowie Informationen über die Qualität der Verbindungen. Damit die Nachbarn ebenfalls Kenntnis von der Existenz des Teilnehmers haben, werden periodisch selbst Hello-Pakete verschickt. Wie in klassischem Link-State-Routing üblich, wird die lokale Nachbarschaft periodisch als Topologie-Paket durch das Netzwerk verschickt. Alle Teilnehmer nutzen diese Information, um ihre eigene Sicht auf die Topologie zu aktualisieren.

Paket-Versender

Der Paket-Versender ist für den Versand aller Pakettypen verantwortlich. Dazu wird als Absender des Paketes die eigene Adresse eingetragen. Während Flut- und Hello-Pakete ansonsten unverändert bleiben, muss für das Versenden von Unicast-Paketen zusätzlich die Zieladresse bestimmt werden, indem die Topologie nach dem nächsten Hop für den Empfänger befragt wird. Anschließend wird das Paket an das rawbasic-Modul übergeben.

4.2 Allgemeine Anpassungen von AWDS

Als bisheriges Einkanal-Routingverfahren ist AWDS nicht für die gleichzeitige Nutzung verschiedener Funkkanäle ausgelegt. Aufgrund seiner modularen Struktur ist es aber ohne Weiteres möglich, Unterstützung für weitere Kanäle hinzuzufügen.

4.2.1 Modulstruktur

Um mehrere Funkkanäle zu nutzen, muss AWDS in der Lage sein, mehrere Netzwerkkarten anzusprechen. Da diese Funktion vom *rawbasic*-Modul abstrahiert wird, bietet es sich an, das Modul so oft zu laden, wie Funkkanäle zur Verfügung stehen (Abb. 4.5). Damit jedes der Module eine andere Netzwerkkarte anspricht, bekommt es diese durch einen neuen Parameter mitgeteilt. Des Weiteren muss sichergestellt werden, dass jedes der *rawbasic*-Module unter einem eindeutigen Bezeichner im ObjectRepository aufgefunden werden kann. Hierzu speichert es sich selbst inklusive des Namens der zugewiesenen Netzwerkkarte ab, beispielsweise als “*rawbasic_ath0*”.

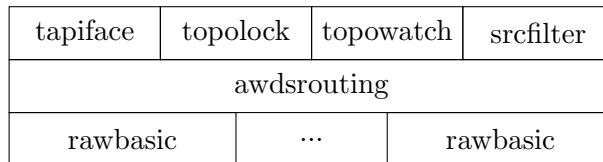


Abbildung 4.5: Nach der Umstrukturierung der AWDS-Module kann das *awdsrouting*-Modul beliebig viele *rawbasic*-Module verwalten.

Das *awdsrouting*-Modul besitzt nun den neuen Parameter “*--ifaces*”, über den ihm alle verwendeten Netzwerkkarten bekannt gemacht werden. Aus deren Namen leitet es sich die Bezeichnung der *rawbasic*-Module her und lädt diese aus dem ObjectRepository.

Innerhalb des gesamten AWDS-Quellcodes sind alle Datentypen auf die Verarbeitung eines einzelnen Funkkanals ausgelegt. Zur Verwaltung mehrerer Funkkanäle müssen diese Daten mehrfach gespeichert werden. Sie wurden daher jeweils in einen Vektor der C++ Standard Template Library (STL) [SLL95] übernommen. Diese Datenstruktur garantiert den Zugriff auf ihre Elemente in konstanter Zeit, wenn deren Index bekannt ist. Intern wird ein spezifischer Funkkanal daher stets über einen Index repräsentiert. Dieser Index wird aus der Reihenfolge der *rawbasic*-Module beim Start von AWDS abgeleitet. Beispielsweise wird dem ersten *rawbasic*-Modul eine Netzwerkkarte zugewiesen, die auf Kanal 6 funkt. Alle für Kanal 6 spezifischen Daten werden dann an erster Stelle in den jeweiligen Vektoren gespeichert. Des Weiteren erwarten alle Methoden, die funkanal-spezifische Aktionen ausführen, den Index des jeweiligen *rawbasic*-Moduls als weiteren Parameter. Dazu gehören beispielsweise alle Methoden zum Paketempfang und zur Routenberechnung.

4.2.2 Nutzung der ETX-Metrik

Die von AWDS7 übernommene Code-Basis setzt die Hop-Count-Metrik zur Bestimmung der Linkkosten ein. Für die Anwendung in Mehrkanal-Verfahren ist die Hop-Count-Metrik allerdings ungeeignet, weil die Pfadkosten nur von der Länge der Pfade abhängen. So erhält eine zuverlässige, schnelle Verbindung die gleiche Gewichtung wie eine unzuverlässige, langsame Verbindung. Die verschiedenen Ausbreitungseigenschaften der Funkkanäle führen vor allem innerhalb von Gebäuden jedoch zu deutlichen Unterschieden in Zuverlässigkeit und Geschwindigkeit der Verbindungen [LC].

Um eine bessere Bewertung der Verbindungen zu erreichen, wird stattdessen die ETX-Metrik eingesetzt. Sie schätzt die Anzahl der notwendigen Paketwiederholungen, bis ein Paket erfolgreich abgeschickt werden kann. Dazu wird in AWDS die Zustellungsrate der Hello-Pakete herangezogen. Je häufiger Hello-Pakete auf der Verbindung zu einem Nachbarn verloren gehen, desto schlechter wird diese bewertet. Das inzwischen veröffentlichte AWDS8 nutzt ebenfalls standardmäßig die ETX-Metrik. Zur Nutzung der Metrik muss lediglich das Modul `etxmetric.so` geladen werden.

Die ETX-Metrik bewertet ausfallfreie Pfade stets gleich gut, selbst wenn diese aufgrund verschiedener Senderaten unterschiedlich schnell sind. Eine bessere Bewertung verspricht die Nutzung der ETT-Metrik, welche allerdings noch nicht in AWDS implementiert ist.

Damit die ETT-Metrik Linkkosten ermitteln kann, muss sie die Senderate der Verbindung kennen. Die Bestimmung der Senderate ist bei Teilnehmern mit automatischer Ratenanpassung aber nicht ohne Weiteres möglich, weil sie den Austausch von Paketen erfordert. Falls der Netzwerktreiber nicht die Möglichkeit bietet, die Senderate aus den aktuellen Datenübertragungen zu bestimmen, müssen zusätzliche Pakete verschickt werden, welche zusätzliche Bandbreite beanspruchen. In jedem Fall müssen Pakete an diejenigen Nachbarn geschickt werden, mit denen momentan keine Daten ausgetauscht werden. Außerdem sind Verbindungen problematisch, die sich in einem Ratenübergang befinden. Dann sind regelmäßige Übertragungsfehler zu erwarten [Wie07]. Darüber hinaus führt die häufige Änderung von Verbindungskosten zu instabilen Pfaden, deren ständige Aktualisierung das Netzwerk stark belasten kann. Aus diesem Grund wird die ETT-Metrik nicht eingesetzt.

4.2.3 Paketversand

Der Versand von Hello- und Flut- Paketen erfolgt durch die Weitergabe des Pakets an alle `rawbasic`-Module, was dem gleichzeitigen Versand über alle verfügbaren Funkkanäle entspricht.

Im effizienten Modus werden Unicast-Pakete dagegen nur auf einem einzigen Funkkanal versendet, um Kollisionen auf dem Medium zu verringern. Nachdem der nächste Hop von der Topologie-Komponente abgefragt wurde, muss das Paket daher an genau

eines der rawbasic-Module übergeben werden. Zur Bestimmung des richtigen rawbasic-Moduls befragt der Paket-Versender die Nachbarschafts-Komponente nach derjenigen Nachbarschaft, die den nächsten Hop enthält. Hierzu ruft er die neue Methode `AwdsRouting::getBasicIndex()` auf, welche die Knoten aller Nachbarschaften mit dem übergebenen Knoten vergleicht und bei Gleichheit den Index der entsprechenden Nachbarschaft zurück gibt. Der Index definiert den Funkkanal der Übertragung und spezifiziert gleichzeitig das zu benutzende rawbasic-Modul.

4.3 Diagonale Links

4.3.1 Erweiterung der Topologie

Der grundlegende Gedanke hinter der Idee der diagonalen Links ist die Anpassung der Topologie, um weiterhin einen Shortest-Path-Algorithmus einsetzen zu können. Jeder Teilnehmer wird als eine Vielzahl von Knoten modelliert, so dass die Adjazenzliste nun wesentlich mehr Elemente umfasst. Die Hinzunahme virtueller Verbindungen besteht im Wesentlichen in der Erweiterung der Nachbarschafts-Listen um neue Knoten (Abb. 4.6). Weitere Änderungen an der Verwaltung der Topologie werden nachfolgend erläutert.

Die Gruppierung der Knoten (siehe Abschnitt 3.2.3) erfolgt im `GroupCache`, dessen Struktur in Abbildung 4.7 dargestellt wird. Er besteht aus zwei Tabellen. Die erste Tabelle speichert für jeden Teilnehmer die Menge seiner zugehörigen Knoten und kann mittels `getNodeIds()` abgefragt werden. Die zweite Tabelle speichert für jeden Knoten den entsprechenden Teilnehmer. Dieser kann mittels `getGrpId()` bestimmt werden. Die `NodeId` ist die bisher übliche MAC-Adresse jedes Knotens, während die `GroupId` den Teilnehmer als Ganzes repräsentiert. Sie entspricht jeweils der `NodeId` des ersten Knotens.

Die Topologie-Pakete enthalten nun nicht mehr genau eine Nachbarschaft, sondern je eine pro Funkkanal (Tabelle 4.1). Eine einzelne Nachbarschaft besteht aus der `NodeId` des Knotens sowie einer Liste der Nachbarn, inklusive der Verbindungsqualität zu ihnen (Tabelle 4.2). Diese Änderungen am Paketformat führen zur Inkompatibilität mit dem bisherigen AWDS-Protokoll, weswegen die Protokoll-Version von 8334 auf 8340 angehoben wurde. Die Protokoll-Version stellt sicher, dass AWDS nur Pakete mit gültigem Format auswertet und bearbeitet.

Die Auswertung von Topologie-Paketen ist Aufgabe der Topologie-Komponente. Die Methode `Topology::feed()` liest die Zuordnung zwischen Knoten und Teilnehmern in den `GroupCache` ein. Sie übernimmt weiterhin die Nachbarschafts-Daten aus dem Topologie-Paket direkt in die Adjazenzliste der lokalen Topologie. Dabei ist problematisch, dass im Topologie-Paket selbst keine virtuellen Verbindungen enthalten sind. Mit der Übernahme in die Adjazenzliste würden damit alle dort gespeicherten virtuellen Verbindungen gelöscht. Deswegen müssen die diagonalen Links bei jeder Auswertung

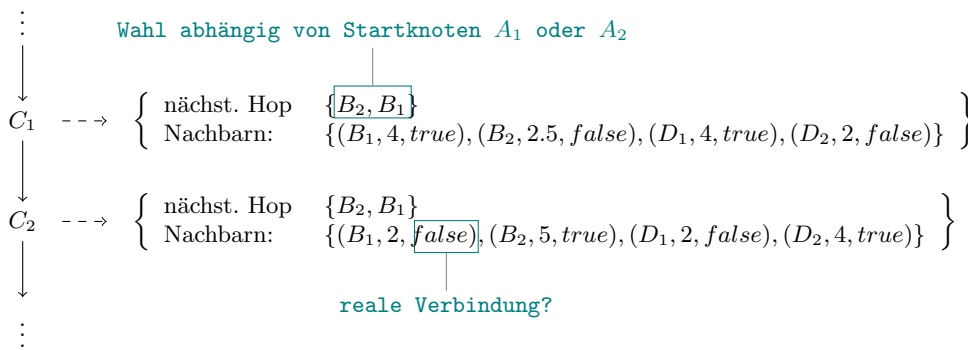
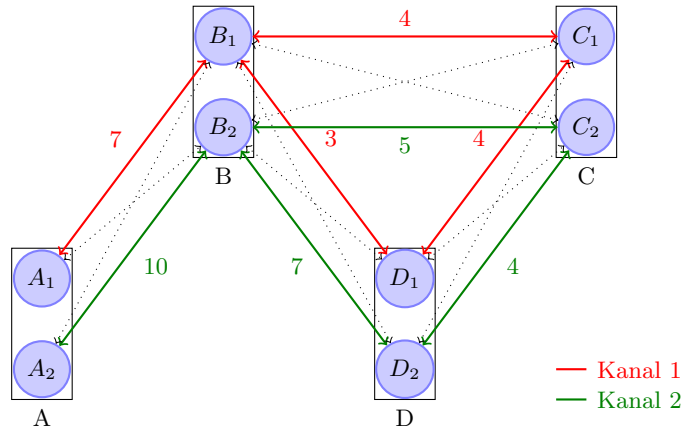


Abbildung 4.6: Ausschnitt der lokalen Sicht von Teilnehmer A auf eine Mehrkanal-Topologie mit diagonalen Links.

eines Topologie-Pakets neu generiert werden, auch wenn sich an der Topologie selbst überhaupt nichts geändert hat. Dazu werden für jede Verbindung zu einem Nachbarn mittels des `GroupCache` die unbeteiligten Knoten des Teilnehmers bestimmt. Von diesen ausgehend wird ein diagonaler Link zum tatsächlichen Zielknoten des Nachbarn erzeugt (Abb. 4.8).

Unglücklicherweise gibt es in AWDS keine strikte Trennung zwischen Routing und Topologie. So enthält die Topologie zusätzliche Daten zur Verwaltung von Routen, wie zum Beispiel eine Angabe über den nächsten Hop. Da die Routinginformationen auf einen einzelnen lokalen Knoten zugeschnitten sind, mussten diese Daten nun für jeden Funkkanal bereitgestellt werden. Dazu wurden sie in einen Vektor der STL verpackt. Die Reihenfolge der Daten entspricht dabei der Reihenfolge der rawbasic-Module. So können die für weiterzuleitende Pakete erforderlichen Topologie-Daten direkt über den Index desjenigen Knotens angefordert werden, der das Paket empfangen hat. Schließlich muss dafür gesorgt werden, dass diagonale Links nur die halben Kosten ihrer realen Entsprechungen aufweisen. Dies geschieht in `Metric::calculate()`. Die-

4 Implementierung

GroupCache
+groups: std::map<GroupId, std::set<NodeId>> +nodes: std::map<NodeId, GroupId>
+update(const NodeId&, const GroupId&) +getGroupId(GroupId&, const NodeId&): bool +getNodeIds(std::set<NodeId>&, const GroupId&): bool +removeGroupId(const GroupId&) +removeNodeId(const NodeId&)

Abbildung 4.7: Die Klasse GroupCache.

Name	Typ	Bedeutung
Validity	size_t	Gültigkeit
ChannelCount	size_t	Anzahl der Nachbarschaften
Nachbarschaften	Nachbarschaft []	Feld von Nachbarschaften
NameLen	unsigned char	Länge des Teilnehmer-Namens
Name	char*	Teilnehmer-Name

Tabelle 4.1: Genereller Aufbau eines Topologie-Pakets. Anpassungen zur Speicherung mehrerer Nachbarschaften sind hervorgehoben.

se Methode errechnet aus den Distanzwerten eines Links einen Gewichtswert. Nachdem das Gewicht ermittelt wurde, wird der Link darauf geprüft, ob er diagonal ist. Für diesen Zweck besitzt die Klasse `LinkQuality` die neue boolesche Membervariable `horizontal`, welche angibt, ob die Verbindung real ist. Diagonale Links behalten ihre Kosten bei, während die Kosten horizontaler Links verdoppelt werden. Auf diese Weise werden Rechenungenauigkeiten beim Umgang mit den ganzzahligen Gewichten vermieden.

Name	Typ	Bedeutung
Self	NodeId	lokale Knoten-Adresse
NeighCount	char	Anzahl der Nachbarn
Neigh	NodeId	Adresse des 1. Nachbarn
LinkQuality	uint_16t	Verbindungsqualität zum 1. Nachbarn
...

Tabelle 4.2: Struktur einer Nachbarschaft innerhalb eines Topologie-Pakets.

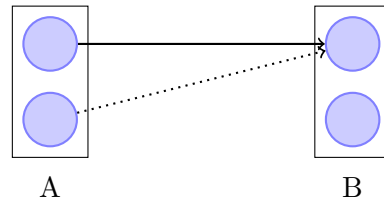


Abbildung 4.8: Bei der Erzeugung diagonalen Links bildet der Zielknoten des realen Links den Endpunkt, die unbeteiligten Knoten des Senders bilden die Startpunkte.

4.3.2 Pfadsuche

Die Angabe des nächsten Hops ist abhängig vom jeweiligen lokalen Knoten, der das Paket empfangen hat. Folglich muss die Pfadsuche für jeden lokalen Knoten separat durchgeführt werden. Hierzu muss der Methode `Topology::getNextHop()` nun zusätzlich der Index des Startknotens übergeben werden. Alle anderen lokalen Knoten werden vor dem Dijkstra-Verfahren versteckt, sie werden weder als zusätzliche Startknoten, noch als normale Knoten im Netzwerk behandelt. Dadurch soll vermieden werden, dass sie Teil des Pfades werden. Weil bereits der Startknoten Teil des Pfades ist, bedeutete dies das Vorhandensein einer Routingschleife. Während der Bestimmung der kürzesten Routen speichert Dijkstra in jedem Knoten seinen Vorgänger ab. Diese Information wird dazu genutzt, um im Anschluss den Pfad vom Zielknoten zurückzuverfolgen, der letzte Knoten ist dann der nächste Hop. In Abschnitt 3.2.3 wurde jedoch beschrieben, dass von den ermittelten Pfaden zu allen Knoten des Ziel-Teilnehmers nur der kürzeste interessant ist. Es ist also gar nicht nötig, für alle Knoten der Teilnehmer im Netzwerk den nächsten Hop zu bestimmen. Stattdessen wird der nächste Hop jeweils nur für den am nächsten gelegenen Knoten aller Teilnehmer ermittelt, die restlichen Knoten des gleichen Teilnehmers erhalten diesen ebenfalls zugewiesen (Tabelle 4.3). Im Prinzip würde es ausreichen, als Zielknoten keine Knoten-Ids, sondern Gruppen-Ids zu speichern. Damit würde die Tabelle der nächsten Hops deutlich weniger Einträge enthalten. Eine solche Tabelle gibt es jedoch nicht, weil AWDS die nächsten Hops innerhalb der Adjazenzliste unter dem jeweiligen Zielknoten speichert. An dieser Stelle ist eine Zusammenlegung der Knoten nicht möglich.

Die Unterscheidung des Startknotens ist dann wichtig, wenn es tatsächlich einen solchen gibt. Dies ist immer derjenige lokale Knoten, der das Paket empfangen hat. Wurde das Paket allerdings auf dem Teilnehmer selbst erzeugt, so sind alle seine Knoten potentielle Startknoten. Für diesen Fall wurde die Methode `getNextHop()` überladen, sie lässt sich auch ohne die Angabe eines Indexes aufrufen. Dann werden die Entfernungen von allen Startknoten zum Ziel-Teilnehmer miteinander verglichen und der nächste Hop auf der kürzesten Route zurückgegeben.

Startknoten	Zielknoten	nächster Hop
A_1	A_1	-
A_1	A_2	-
A_1	B_1	B_2
A_1	B_2	B_2
A_1	C_1	B_2
A_1	C_2	B_2
A_1	D_1	B_2
A_1	D_2	B_2
A_2	A_1	-
A_2	A_2	-
A_2	B_1	B_1
A_2	B_2	B_1
A_2	C_1	B_1
A_2	C_2	B_1
A_2	D_1	B_1
A_2	D_2	B_1

Tabelle 4.3: Nächste Hops von Teilnehmer A.

4.3.3 Topologie-Export

Der bereits in das awdsrouting-Modul integrierte Shell-Befehl `topo` dient zum Abrufen der Topologie-Daten auf der Kommandozeile. Er beachtet nun die Zuordnung mehrerer Knoten zu einem Teilnehmer. Bei größeren Netzwerken wird die Überwachung der Topologie auf der Kommandozeile jedoch schnell unübersichtlich. Zu diesem Zweck bietet das `topowatch`-Modul den Export der Topologie-Daten auf Port 8333 im XML-Format an. Das externe Java-Programm “awds-topoviewer” liest diese Daten aus und stellt sie grafisch dar.

Eine Darstellung aller Knoten ist in einer Mehrkanal-Topologie nicht sinnvoll, weil dann jeder Teilnehmer mehrfach in der Topologie auftaucht. Sinnvoller ist die Anzeige aller Teilnehmer, deren parallele Verbindungen untereinander als eine Vielzahl von Kanten visualisiert werden. Damit der TopoViewer überhaupt Knoten zu Teilnehmern zuordnen kann, wurde das Format des XML-Datenstroms angepasst. Dieser enthält zu jedem Knoten ein zusätzliches Attribut mit dessen Gruppen-Id. Beispielsweise besteht das Element `add.edge` neben den bisherigen Attributen `from` und `to` zusätzlich aus `fromgrp` und `togrp`. Beim Auswerten der Daten verarbeitet der TopoViewer ausschließlich die Gruppen-Ids, sofern vorhanden. Enthält der XML-Strom keine Gruppen-Ids, werden wie bisher die Knoten-Ids ausgewertet. Damit bleibt der TopoViewer kompatibel zum Topologie-Export des originalen AWDS.

Der TopoViewer nutzt die *TouchGraph*-Bibliothek [tou], um die Topologie intern zu

verwalten. Die bisherige Topologie sah zwischen zwei Teilnehmern jeweils nur eine Kante vor. Weil es nun parallele Kanten zwischen Teilnehmern gibt, besitzt die Klasse `Edge` den neuen Parameter `index`. Dieser entspricht dem Index des zugewiesenen Kanals. Der Index wird innerhalb des `TopoViewers` aus der Beziehung zwischen Knoten-Ids und Gruppen-Ids bestimmt. Weil die Gruppen-Id stets der ersten Knoten-Id entspricht, wird der Index bei Gleichheit beider Ids auf 0 gesetzt, bei Ungleichheit auf 1. Dieses Verfahren liefert nur deswegen sinnvolle Resultate, weil die zur Verfügung stehende Hardware lediglich maximal zwei Funkkanäle unterstützt. Für Netzwerke mit mehr als zwei Funkkanälen lässt sich der Index eines Knotens auf diese Weise nicht zuverlässig bestimmen und müsste bereits von AWDS in den XML-Strom geschrieben werden.

In der bisherigen Implementierung wird eine Kante stets vom Mittelpunkt des einen Knotens zum Mittelpunkt des anderen Knotens gezeichnet. Anschließend werden die Knoten selbst gezeichnet, so dass der Eindruck entsteht, die Kante verbindet lediglich die Außenrahmen der Knoten. In einem Netzwerk mit zwei Funkkanälen benötigt jedes Teilnehmer-Paar zwei Kanten, die möglichst parallel verlaufen sollten. Weil für jeden Teilnehmer nur der Mittelpunkt bekannt ist, müssen die Endpunkte der Kanten zunächst berechnet werden. Damit die parallelen Kanten stets als solche erkennbar sind, sollen sie einen bestimmten Abstand zur Verbindungslinie der Mittelpunkte beider Knoten haben. Dazu wird die Verbindungslinie zunächst als Vektor bestimmt und deren Orthogonale berechnet. Die beiden Endpunkte der Kanten liegen auf der Orthogonalen mit einem festen Abstand zum Mittelpunkt des Teilnehmers (Abb. 4.9). Die simple waagerechte Anordnung der Endpunkte ist nicht möglich, weil die Kanten bei einer horizontalen Anordnung der Teilnehmer sonst ineinander liegen würden und nicht mehr unterschieden werden könnten.

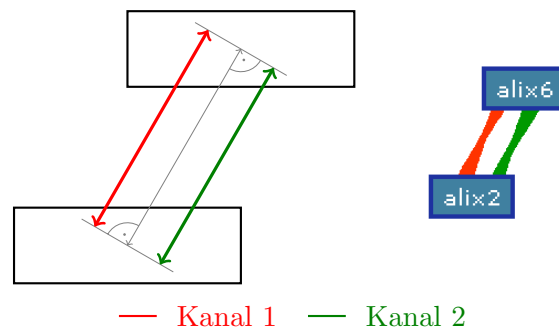


Abbildung 4.9: Konstruktion der Kanten-Endpunkte zwischen zwei Teilnehmern (links) und Anzeige im `TopoViewer` (rechts).

Problematisch ist, dass die jeweils gegenüberliegenden Endpunkte einer Kante konstruktionsbedingt zu unterschiedlichen Kanälen gehören. Dies stört besonders beim Einfärben der Kanten, weil der Beginn und das Ende jeder Kante unterschiedlich gefärbt sind. Um einfarbige Kanten zu erhalten, müssen für jedes Knoten-Paar die Endpunkte in genau einem der Teilnehmer vertauscht werden. Dazu wird der Winkel

4 Implementierung

der Mittellinie berechnet. Die Vertauschung der Endpunkte erfolgt in dem Teilnehmer, dessen Mittellinie einen Winkel von mindestens 180° aufweist, d.h. südlich zeigt. Zur Positionierung der Teilnehmer auf der Zeichenfläche nutzt der TopoViewer ein Feder-Masse-Modell. Jeder Teilnehmer wird als Massepunkt modelliert, der über Federn mit seinen Nachbarn verbunden ist. Demnach ziehen sich Teilnehmer gegenseitig an, wenn zwischen ihnen eine Verbindung besteht. Diese Anziehungskraft ist umso stärker, je geringer die Kosten der Verbindung sind. Besteht zwischen Teilnehmern keine direkte Verbindung, stoßen sie sich ab. Durch eine Veränderung der Topologie verändert sich das Kräfteverhältnis zwischen den Teilnehmern, die daraufhin automatisch neu positioniert werden. Dieser Prozess ist fließend, erreicht nach wenigen Sekunden jedoch stets einen stabilen Zustand. Durch die Einführung paralleler Kanten ziehen sich die Teilnehmer nun stärker an als zuvor, was dazu führt, dass sie selbst auf der größten Zoom-Stufe noch übereinander angezeigt werden. Eine Erhöhung des maximal zulässigen Vergrößerungsfaktors löste dieses Problem, die Verringerung der Anziehungskraft zwischen den Teilnehmern wäre ebenso möglich gewesen.

4.4 Heuristik

Das heuristische Verfahren nutzt zur Einschränkung des Suchraums das Verfahren der diagonalen Links. Deswegen wurde entschieden, die bisherige Implementierung der diagonalen Links um die neuen Funktionen des Brute-Force-Schritts zu erweitern. Falls nicht anders angegeben, sind daher alle in diesem Abschnitt besprochenen Anpassungen Ergänzungen der Implementierung der diagonalen Links, nicht des ursprünglichen AWDS. Ein Großteil der weiteren Änderungen fand in der Topologie-Klasse, speziell bei der Bestimmung der nächsten Hops, statt.

4.4.1 Verwaltung der Kanalkosten

Das Konzept der Heuristik sieht die Zuordnung der im WLAN-Standard spezifizierten Kanalnummer zu den jeweiligen Kanalkosten vor. Allerdings wird zur Vorauswahl die Implementierung der diagonalen Links eingesetzt, die statt der Kanalnummer lediglich den Index des Kanals speichert. Damit die Ermittlung, Propagierung und Speicherung der Kanalnummer nicht nachträglich implementiert werden musste, verwendet die prototypische Implementierung ebenfalls den Index des Kanals anstelle der Kanalnummer. Dadurch sind Teilnehmer mit weniger Netzwerkkarten allerdings an eine feste Kanalreihenfolge angewiesen und können Kanäle mit einem zu hohen Index nicht nutzen. Der Einsatz der in Abschnitt 3.1.3 beschriebenen automatischen Wahl des besten Funkkanals ist dadurch nicht mehr möglich.

Prinzipiell ist eine explizite Speicherung des Kanalindizes nicht notwendig, wenn die Kanalkosten in einer listenförmigen Datenstruktur gespeichert werden. Dann wird der

Index bereits indirekt durch die Position der Kosten in der Liste festgelegt. Es ist jedoch davon auszugehen, dass vollständige Implementierungen die Speicherung der Kanalnummer bevorzugen, um keinen Beschränkungen in der Kanalreihenfolge zu unterliegen. Um mit diesen Implementierungen im Paketformat binärkompatibel zu sein, muss im Header von Nutzdaten-Paketen die Zuordnung einer Ganzzahl x auf die jeweiligen Kanalkosten erfolgen (detaillierte Erläuterungen zur Anpassung der Pakete folgen in Abschnitt 4.4.2). Dann entspricht x in vollständigen Implementierungen der Kanalnummer und in dieser prototypischen Implementierung dem Index des Kanals. In diesem Zusammenhang ist die Verwaltung der Kosten in einer tabellarischen Struktur ebenfalls sinnvoll.

Für die Speicherung der Kanalkosten wurde die neue Klasse `VectorWeight` eingeführt. Ihre Struktur wird in Abbildung 4.10 dargestellt. Sie erhielt diese Bezeichnung, weil die Kanalkosten aus Sicht der Topologie eine vektorielle Größe darstellen. Tatsächlich implementiert sie aus oben genannten Gründen jedoch eine Hashtabelle, welche vom Kanalindex auf die Kanalkosten abbildet.

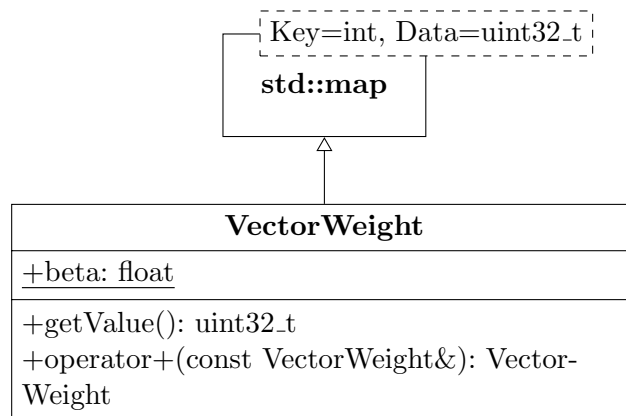


Abbildung 4.10: Die Klasse `VectorWeight`.

Die Addition von Kanalkosten erfolgt durch simples Addieren von `VectorWeight`-Objekten. Dazu wurde der Additions-Operator überladen. Es werden jedoch lediglich die Kosten mit gleichem Kanal-Index addiert, die Kosten anderer Kanäle werden übernommen.

Die Bildung eines skalaren Werts erfolgt in der Methode `getValue()`, deren Implementierung in Listing 4.1 angegeben ist. Sie iteriert über alle gespeicherten Kanalkosten und bestimmt die Kennwerte für Durchsatz und Latenz. Der Durchsatz entspricht dabei dem maximalen Wert aller gespeicherten Gewichte (Zeilen 11-12), die Latenz ist die Summe aller Gewichte (Zeile 14). Mittels des Wichtungsparameters `beta` werden beide Werte zu einem skalaren Wert verschmolzen, der schließlich zurückgegeben wird (Zeile 17).

Weil der Entscheidungsbaum auf Objekten vom Typ `VectorWeight` operieren soll,

4 Implementierung

```
1  uint32_t getValue()
2  {
3      uint32_t throughput = 0;
4      uint32_t latency = 0;
5
6      // iterate over all stored channels
7      for (const_iterator it=begin(); it!=end(); it++)
8      {
9          uint32_t weight = it->second;
10
11         if (weight > throughput)
12             throughput = weight;
13
14         latency+= weight;
15     }
16
17     return (uint32_t) ((1-beta) * latency + beta * throughput);
18 }
```

Listing 4.1: Die Methode `VectorWeight::getValue()`.

gibt es je nach seiner Tiefe sehr viele solcher Objekte. Um Inkonsistenzen in der Pfadberechnung zu vermeiden, muss darauf geachtet werden, dass Änderungen des Wichtungsparameters `beta` in allen Objekten gleichzeitig stattfinden. Durch die Deklaration von `beta` als statische Membervariable wird erreicht, dass der Wichtungsparameter innerhalb der Klasse gespeichert wird und in allen Objekten identisch ist.

4.4.2 Anpassung der Datenpakete

Zur Vermeidung von Routingschleifen muss sichergestellt werden, dass alle Teilnehmer entlang eines Pfades identische Entscheidungsbäume aufbauen. Um den Baum mit den bereits durchlaufenen Kosten initialisieren zu können, müssen diese innerhalb des Pakets übertragen werden.

Der Header von Unicast-Paketen wurde dahingehend erweitert, dass er ein Objekt des Typs `VectorWeight` aufnehmen kann (siehe Tabelle 4.4). Die Klasse `UnicastPacket` wurde dafür um die Methoden `getVectorWeight()` und `setVectorWeight()` ergänzt. Problematisch ist hierbei, dass in einem `VectorWeight`-Objekt prinzipiell unendlich viele Kanäle gespeichert werden können und der Header des Unicast-Pakets dadurch unendlich groß werden kann. Die prototypische Implementierung beschränkt sich daher auf die Unterstützung von maximal zwei Funkkanälen. Wird genau ein Kanal im Paket gespeichert, bleibt der nachfolgende Platz für den zweiten Kanal ungenutzt. Damit bleibt die Paketstruktur weiterhin statisch und die Nutzdaten beginnen wie bisher an einem festen Punkt. Das ist praktisch, weil der Zugriff auf die Nutzdaten ohne die

Berechnung eines Startpunkts im Speicher erfolgen kann. Der Index des Funkkanals wird ebenfalls im Paket gespeichert, um Binärkompatibilität zu Implementierungen herzustellen, die an dieser Stelle die Kanalnummer speichern.

Name	Typ	Bedeutung
Control	unsigned char	Kontroll-Bit
Src	NodeId::size	Absender
Seq	size_t	Sequenznummer
UcDest	NodeId::size	Paketziel
NextHop	NodeId::size	nächster Hop
TTL	unsigned char	Time to live
UcType	unsigned char	Unicast-Typ
NumWeights	unsigned char	Anzahl der Kanäle (1 oder 2)
Weight1Index	unsigned char	Index des 1. Kanals
Weight1Value	uint32_t	Kosten des 1. Kanals
Weight2Index	unsigned char	Index des 2. Kanals
Weight2Value	uint32_t	Kosten des 2. Kanals

Tabelle 4.4: Header eines Datenpakets. Ergänzungen zur Speicherung des Paketverlaufs sind hervorgehoben.

4.4.3 Paketversand

Vor der Weiterleitung eines Daten-Pakets wird die Topologie nach dem nächsten Hop befragt. Weil die Bestimmung des nächsten Hops nun abhängig vom bisherigen Paketverlauf ist, muss dieser zunächst aus dem Paket gelesen und zusätzlich an die Topologie-Komponente übergeben werden. Bevor das Datenpaket an den nächsten Hop verschickt wird, wird sein aktuell gespeichertes `VectorWeight`-Objekt mit den Kosten des neuen Links ergänzt. Dies erfolgt in `awdsRouting::send_unicast()`. Die Methode erwartet als Parameter bereits den Index des Sende-Kanals, der zuvor mittels `awdsRouting::getBasicIndex()` ermittelt wurde. Sie liest die aktuellen Pfadkosten aus dem Paket aus und befragt die Topologie nach den Kosten des Sende-Links. Im nächsten Schritt werden diese beiden Kosten miteinander addiert und wieder im Paket gespeichert. Schließlich wird das Paket über das per Index angegebene `rawbasic`-Modul versendet.

4.4.4 Verwaltung der Routingtabelle

Damit nicht bei jedem Paketversand eine neue Pfadsuche gestartet werden muss, benötigt das heuristische Verfahren eine eigene Routingtabelle, die zusätzlich zu den für die Vorauswahl benötigten Routinginformationen verwaltet werden muss. Hierzu

4 Implementierung

wurde die Topologie um den `NextHopCache` erweitert. Diese Hashtabelle bildet von einer `NextHopCacheEntry`-Struktur auf den nächsten Hop ab. Ein `NextHopCacheEntry` besteht aus dem bisherigen Paketverlauf und der Gruppen-Id des Zielknotens, so dass jeder Eintrag in der Routingtabelle der Form

$$(\text{Verlauf, Zielteilnehmer}) \mapsto \text{nächster Hop}$$

genügt. Fragt der Paket-Versender nun den nächsten Hop zu einem bestimmten Zielknoten ab, wird zunächst in der Routingtabelle nachgeschaut, ob dort bereits ein entsprechender Eintrag existiert. Bei Erfolg wird er zurückgegeben, anderenfalls wird die Pfadsuche durchgeführt und der nächste Hop bestimmt. Dieser wird in die Routingtabelle übernommen und schließlich zurückgegeben. Damit die Routinginformationen stets aktuell bleiben, werden alle Einträge bei jeglichen Änderungen der Topologie entfernt.

4.4.5 Pfadsuche

Für die Abfrage des nächsten Hops erwartet die Methode `Topology::getNextHop()` nun zusätzlich den bisherigen Paketverlauf in Form eines `VectorWeight`-Objektes. Enthält die Routingtabelle keinen entsprechenden Eintrag, muss dieser zunächst bestimmt werden. Dazu wird mittels der Vorauswahl die Topologie reduziert und an den Brute-Force-Schritt übergeben.

Vorauswahl

In der Vorauswahl wird ein einzelner Teilnehmer-Pfad zum Empfänger konstruiert. Hierfür wird die von der Implementierung der diagonalen Links übernommene Methode `Topology::getNextHop()` aufgerufen, die mittels des Dijkstra-Verfahrens den kürzesten Pfad zu einem der Knoten des Zielteilnehmers sucht. Diese Methode vergleicht die Pfade von allen Startknoten zu allen Zielknoten, wählt davon den besten aus und liefert zusätzlich den Index des Sendeknotens. Der Index bestimmt die Wahl der Routinginformationen, mit deren Hilfe der Pfad vom Zielknoten zum eigenen Teilnehmer zurückverfolgt wird.

Die Rückverfolgung des Pfades ist bereits im Verfahren der diagonalen Links implementiert. Sie wird dort eingesetzt, um den nächsten Hop zu bestimmen. Für die Nutzung im heuristischen Verfahren sind daher keine zusätzlichen Topologie-Daten zu speichern.

Der Teilnehmer-Pfad P ist eine Liste von Gruppen-Ids. Er wird mit der Gruppen-Id des Empfängers initialisiert. Anschließend wird der Pfad vom Empfänger so lange zurückverfolgt, bis der eigene Teilnehmer erreicht ist. Die dabei durchlaufenen Teilnehmer werden jeweils an den Anfang von P geschrieben. Nachdem das Verfahren

terminiert, ist der Teilnehmer-Pfad komplett aufgebaut. Die Reihenfolge seiner Einträge entspricht der Reihenfolge der Teilnehmer, die das Paket durchlaufen muss.

Abbildung 4.11 zeigt eine Topologie, auf der Teilnehmer A als Absender eines Pakets einen Pfad zu Teilnehmer C sucht. Das Dijkstra-Verfahren hat dazu den Pfad $\{A_2, B_1, C_2\}$ bestimmt, so dass die Topologie auf die Teilnehmer A, B und C reduziert wird. Sie bilden ein Teil-Netzwerk, auf dem im Brute-Force-Schritt die optimale Kanalkombination bestimmt werden soll. Da der erste Teilnehmer im Pfad immer der eigene Teilnehmer ist, kann er entfernt werden. Somit wird der Pfad $P = \{B, C\}$ an den Brute-Force-Schritt übergeben.

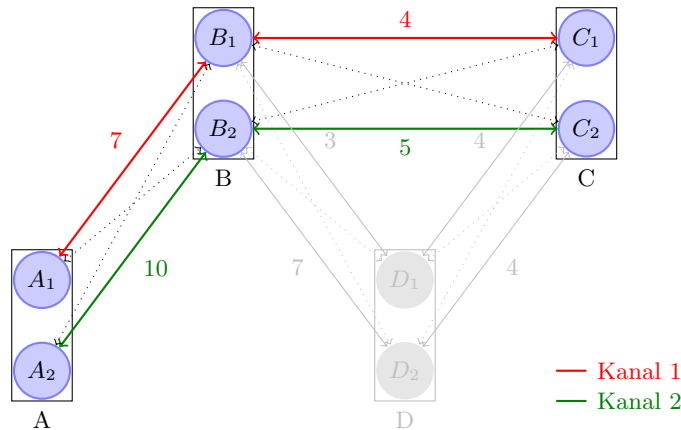


Abbildung 4.11: Reduzierter Suchraum für die Pfadsuche von Teilnehmer A zu Teilnehmer C.

Konstruktion des Entscheidungsbaumes

Der Entscheidungsbaum wird aufgebaut, um alle Kanalkombinationen zum Zielteilnehmer zu ermitteln. Diese befinden sich nach Abschluss der Konstruktion als Blätter in der untersten Ebene. Alle Ebenen darüber sind lediglich während der Konstruktion interessant und sollen nicht permanent gespeichert werden. Die Konstruktion des Baumes erfolgt daher Ebene für Ebene, d.h. sie folgt dem Schema einer Breitensuche. Es genügt, anstelle des gesamten Baumes jeweils immer nur die aktuell betrachtete Ebene zu speichern. Sie enthält eine Auflistung L aller momentanen Blätter. Ein einzelnes Blatt ist eine Struktur vom Typ `Leaf`. Es speichert den nächsten Hop, den aktuellen Knoten sowie die Kosten, um diesen zu erreichen.

Abbildung 4.12 zeigt die einzelnen Schritte, die während des Brute-Force-Schrittes ablaufen. Schritt 0 beschreibt die Ausgangssituation. Die Vorauswahl übergibt dem Brute-Force-Schritt den Teilnehmer-Pfad $P = \{B, C\}$, L ist zunächst leer.

In Schritt 1 wird der Entscheidungsbaum mit dem ersten Teilnehmer aus P initialisiert, welcher stets der direkte Nachbar des eigenen Teilnehmers ist. Dazu werden die

4 Implementierung

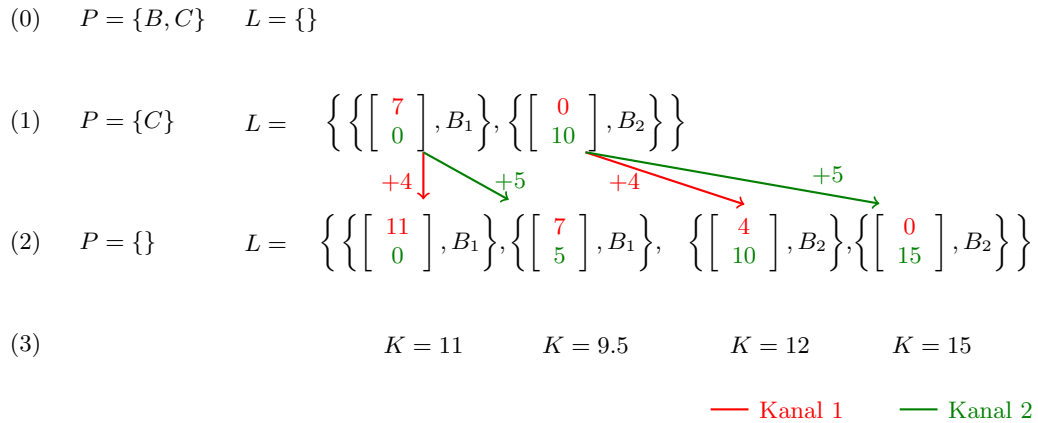


Abbildung 4.12: Suche nach der optimalen Kanalkombination.

Linkkosten zu jedem der Nachbar-Knoten mit den Initialkosten addiert, welche aus dem Paket ausgelesen wurden. Die Initialkosten entsprechen in diesem Fall $\{0, 0\}$, weil Teilnehmer A der Absender des Pakets ist. Neben den Kosten wird pro Blatt unter anderem der eigene nächste Hop mitgeführt. Dieser entspricht den Knoten B_1 bzw. B_2 . Der Teilnehmer B ist nun in den Baum aufgenommen worden, so dass er aus P entfernt werden kann.

Schritt 2 besteht aus einem iterativen Prozess, der pro Durchlauf eine weitere Ebene des Baumes konstruiert. Dazu wird jeweils der erste Teilnehmer aus dem Teilnehmer-Pfad P entfernt und die Linkkosten zu dessen Knoten mit allen momentanen Blättern des Baumes kombiniert, sofern zwischen diesen tatsächlich eine Verbindung besteht. Die Anzahl der zu speichernden Blätter vervielfacht sich demnach pro Durchlauf um die Anzahl der verfügbaren Verbindungen zum aktuell betrachteten Teilnehmer. Jedes Blatt übernimmt den nächsten Hop von seinem Vorgänger. Dieser Prozess läuft so lange, bis der Teilnehmer-Pfad komplett abgearbeitet wurde. Da im Beispiel der Teilnehmer-Pfad nur noch ein Element enthielt, besteht Schritt 2 aus lediglich einem Durchlauf.

Nun enthält L alle Kanalkombinationen vom eigenen Teilnehmer bis zum Zielteilnehmer. Für alle in dieser Liste enthaltenen Elemente wird aus den zugewiesenen Kosten mittels `VectorWeight::getValue()` ein skalarer Wert gebildet. Dies wird in Schritt 3 dargestellt, der zugrunde liegende Wichtungswert beträgt 0.5. Dasjenige Blatt mit dem kleinsten Kosten-Wert identifiziert den optimalen Pfad durch das vorausgewählte Netzwerk. Im Beispiel ist dies das Blatt mit den Kosten $K = 9.5$. Der darin enthaltene eigene nächste Hop ist der Knoten B_1 . Er wird gemeinsam mit den Initialkosten $\{0, 0\}$ in die Routingtabelle eingetragen und an den Aufrufer der Methode zurückgegeben.

4.4.6 Erweiterungen

Modifikation des Wichtungsparameters

Der Wert des Wichtungsparameters β hat signifikanten Einfluss auf das Routingverhalten des Netzwerks. Eine sinnvolle Erweiterung des heuristischen Verfahrens ist daher die Möglichkeit, den Parameter zur Laufzeit anzupassen. Problematisch daran ist, dass β von jedem Knoten separat verwaltet wird, eine Änderung in einem der Teilnehmer hätte keine Auswirkung auf die anderen Teilnehmer. Das Netzwerk würde nicht mehr mit einem einheitlichen Parameter arbeiten, was unvorhersehbare Auswirkungen auf das Routing hätte. Ideal wäre die Möglichkeit, β global für das gesamte Netzwerk ändern zu können. Für diesen Zweck stellt AWDS bereits ein Framework zur Verfügung. In der Datei `packettypes.h` lassen sich eigene Nachrichten definieren, welche mittels der Methoden `send_unicast()` bzw. `send_broadcast()` verschickt werden können. Um den Empfang einer Nachricht in einer eigenen Methode auswerten zu können, muss man diese mittels `registerUnicastProtocol()` bzw. `registerBroadcastProtocol()` im Paket-Auswerter registrieren. Beim Empfang eines entsprechenden Paketes sieht AWDS in seinem Protokoll-Register nach, ob für diesen Typ eine Rückruffunktion eingetragen ist und führt ihn aus. Weiterhin bietet das shell-Modul die Möglichkeit, interaktiv auf das Verhalten von AWDS über eine Eingabeaufforderung Einfluss zu nehmen. Über die Methode `add_command()` lassen sich eigene Kommandos hinzufügen und mittels einer Rückruffunktion verarbeiten.

Die Manipulation des Wichtungsparameters erfolgt über einen neuen Shell-Befehl namens `vWeight`. Nach Eingabe von `# vweight get` wird der aktuelle Parameter des lokalen Knotens ausgelesen und in der Konsole ausgegeben. Eine Änderung des Parameters wird durch die Eingabe von `# vweight set <Wert>` erreicht. Der dabei übergebene Wert wird in ein Flutpaket mit dem Typ `PACKET_TYPE_BC_VECTORWEIGHT` geschrieben und durch das Netz propagiert. Die Methode `vectorWeight_fn()` wurde als Rückruffunktion für den Empfang der neuen Flutpakete registriert. Sie setzt die statische Membervariable `Vectorweight::beta` auf den im Paket gespeicherten Wert. Anschließend wird die gesamte Topologie zurückgesetzt, um die Routingtabellen der Vorauswahl und des Brute-Force-Schrittes zu leeren.

Die Propagierung des neuen Wichtungsparameters benötigt je nach Größe des Netzwerks einige Zeit. Es ist darauf zu achten, dass in diesem Zeitraum keine Datenübertragungen stattfinden. Obwohl die im Paket gespeicherten Kosten gültig bleiben, werden sie von einigen Teilnehmern mit dem neuen Parameter bearbeitet und von anderen Teilnehmern mit dem alten Parameter. Weil sich dann die Entscheidungsbäume in den Teilnehmern unterscheiden, können Routingschleifen auftreten.

Vollständiges Brute-Force

Die Beschränkung der vorgestellten Heuristik auf einen einzelnen Teilnehmer-Pfad soll die Rechenzeit verringern, die zur Suche der besten Kanalkombination aufgewendet wird, so dass mehr Rechenzeit zum Übertragen der Daten zur Verfügung steht. Zur Bewertung der Optimalität des durch die Heuristik ermittelten Pfades müssen jedoch zusätzlich die Messwerte eines vollständigen Brute-Force-Durchlaufs zum Vergleich vorliegen, d.h. einer Pfadsuche, die tatsächlich alle im Netzwerk vorhandenen Paket-Wege analysiert (siehe Abb. 4.13) und somit stets den optimalen Pfad findet.

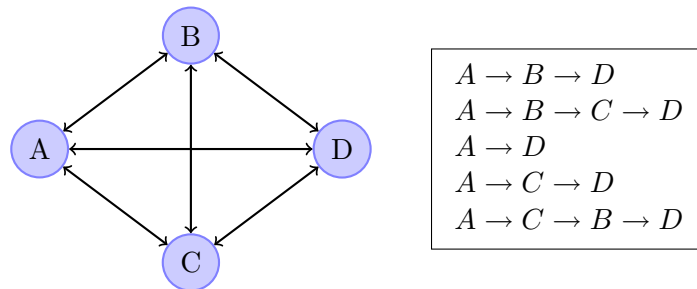


Abbildung 4.13: Liste der Teilnehmer-Pfade von Teilnehmer A zu Teilnehmer D.

In der bestehenden Bestimmung des nächsten Hops wurde die Konstruktion des Entscheidungsbaumes in die neue Methode `Topology::getBestChannelCombination()` ausgelagert. Sie erwartet eine Liste von Teilnehmer-Pfaden, auf denen jeweils die beste Kanalkombination gesucht werden soll. Bei der Heuristik besteht sie lediglich aus einem einzigen Element, beim vollständigen Brute-Force-Verfahren aus allen möglichen Teilnehmer-Pfaden. Für jeden der übergebenen Pfade wird die beste Kanalkombination ermittelt, der beste aller Teilnehmer-Pfade bestimmt schließlich den nächsten Hop. Dieser wird wie bisher in der Routingtabelle abgelegt und zurückgegeben.

Die Bestimmung der Liste aller schleifenfreien Teilnehmer-Pfade erfolgt innerhalb der rekursiven Methode `getPath()` nach dem Schema einer Tiefensuche. Ihre Implementierung ist aufgrund der Unterscheidung von `NodeIds` und `GroupIds` etwas aufgebläht, Listing 4.2 zeigt die Vorgehensweise daher als Pseudo-Quelltext.

Vor dem ersten Aufruf der Methode enthält die Liste aller schleifenfreien Pfade `paths` sowie der betrachtete Pfad `curpath` keine Elemente. In Zeile 3 wird der aktuelle Pfad `path` um den momentan betrachteten Teilnehmer `src` erweitert. Zur weiteren Suche werden alle Nachbarn von `src` bestimmt, die sich noch nicht in `path` befinden (Zeile 11-12). Dadurch soll zum Einen eine Endlosrekursion vermieden werden, zum Anderen entspricht das mehrmalige Aufsuchen eines Teilnehmers einer Routingschleife und ist daher nicht zweckmäßig. Für jeden verbleibenden Nachbarn wird `getPath()` erneut aufgerufen (Zeile 13). Die Zeilen 5 bis 9 behandeln den Abbruch der Rekursion. Er wird ausgelöst, wenn der momentan betrachtete Teilnehmer dem Zielteilnehmer entspricht. Dann wird `path` zur Liste aller Pfade hinzugefügt. Weil der Aufbau des Entscheidungs-

baumes ab dem Nachbar-Teilnehmer beginnt, muss zuvor noch die eigene Gruppen-Id aus dem Pfad entfernt werden.

```

1  function getPath(s, curPath, src, dest, paths)
2  {
3      curPath.add(src)
4
5      if src equals dest
6          curPath.removeFront()
7          paths.add(curPath)
8      return
9  endif
10
11 foreach neighbor n of src
12     if n not in curPath
13         call getPath(s, curPath, n, dest, paths)
14     endif
15 endforeach
16 }
```

Listing 4.2: Pseudo-Quelltext der rekursiven Pfadsuche.

Eine andere Möglichkeit der Umsetzung bestünde darin, statt einem Entscheidungsbaum pro Teilnehmer-Pfad einen einzigen Baum aufzubauen, der alle Pfade enthält. Damit wäre die Suche nach der besten Kanalkombination deutlich schneller, weil die teilweise erforderliche mehrfache Untersuchung gleicher Pfadabschnitte entfällt. Allerdings besitzen verschiedene Teilnehmer-Pfade unterschiedliche Längen. Aufgrund dessen wäre es nicht mehr möglich, lediglich die aktuelle Ebene des Entscheidungsbaumes abzuspeichern und alle oberen Ebenen zu verwerfen. Diese müssten zumindest herangezogen werden, um herauszufinden, ob ein zu untersuchender Teilnehmer bereits im aktuellen Pfad vorkommt oder nicht. Die Implementierung mit je einem Entscheidungsbaum pro Teilnehmer-Pfad bedarf hingegen keiner Änderungen an der Implementierung des Baumes selbst und wurde daher für praktikabler befunden.

Der Typ der Pfadsuche ist vom Schalter `bftype` abhängig. Dieser kann, analog zum Wichtungsparemeter β , global im Netzwerk mittels eigens registrierter Flutpakete vom Typ `PACKET_TYPE_BC_BFTYPE` verändert werden. Der neue Shell-Befehl `bftype` dient zum Auslesen (`# bftype get`) und Ändern (`# bftype set <Wert>`) der aktuellen Pfadsuche. Ein Wert von 0 entspricht einer Anwendung der Heuristik, 1 steht für die Brute-Force-Suche auf der vollständigen Topologie. Während der Propagierung des Typs sollte sichergestellt werden, dass keine Datenübertragungen stattfinden. In dieser Zeit ist `bftype` nicht in allen Teilnehmern identisch, wodurch Routingschleifen entstehen können.

4.5 Zusammenfassung

In diesem Kapitel wurden die prototypischen Implementierungen der in dieser Arbeit entwickelten Routingverfahren als Erweiterung des Routingverfahrens AWDS7 vorgestellt. Sie beschränken sich auf die Umsetzung der wesentlichen Grundfunktionen und besitzen keine der im Konzept erläuterten Erweiterungen. So wurde beispielsweise an einigen Stellen die Implementierung durch die Annahme vereinfacht, dass lediglich zwei Funkkanäle eingesetzt werden. Teilnehmer mit zu wenig Netzwerkkarten besitzen außerdem keine Freiheiten bei der Kanaluweisung, sondern sind an die Nutzung des ersten Funkkanals gebunden.

Die Prototypen sind jedoch ausreichend, um die Funktionsfähigkeit der Verfahren nachzuweisen und deren Eigenschaften experimentell zu untersuchen.

5 Evaluierung

Die Evaluierung der entwickelten Routingprotokolle soll einerseits deren korrekte Funktionsweise überprüfen und andererseits aufzeigen, in welchem Maß die Ziele der Aufgabenstellung erreicht wurden. Sie wurde in einer realen Testumgebung durchgeführt, um die Praxistauglichkeit der Verfahren unter Beweis zu stellen.

5.1 Hardware

Als Mesh-Teilnehmer wurden PCs und eingebettete Geräte eingesetzt. Die eingebetteten Geräte stehen in zwei Ausführungen zur Verfügung. Geräte mit WRAP-Architektur [pceb] enthalten einen vergleichsweise schwachen Prozessor mit 266 MHz, während die Geräte mit der neueren ALIX-Architektur [pcea] mit 433 MHz und größerem Cache wesentlich mehr Rechenkraft besitzen.

Jeder der eingesetzten Teilnehmer ist mit zwei WLAN-Karten von Atheros ausgestattet. Während die erste Netzwerkkarte auf Kanal 8 im 2.4GHz-Band funkt, ist die zweite Netzwerkkarte auf Kanal 36 im 5GHz-Band eingestellt. Als Medienzugriffsverfahren werden IEEE 802.11g bzw. IEEE 802.11a eingesetzt, so dass die Brutto-Datenrate maximal 54 MBit/s beträgt. Die Teilnehmer verwenden jedoch eine automatische Anpassung der Senderate, um auf Schwankungen der Verbindungsqualität zu reagieren.

In [Pfe08] wurde gezeigt, dass die begrenzte Rechenleistung der WRAP-basierten Geräte den Flaschenhals von Datenübertragungen bilden kann. Daher wurde darauf geachtet, dass WRAP-Geräte kein Bestandteil aktiver Pfade sind. Ebenso wurde in [Sch08] festgestellt, dass auch die Rechenleistung der ALIX-basierten Geräte im Mehrkanalbetrieb auf kurzen Pfaden zum Flaschenhals der Übertragung werden kann. Für Messungen kurzer Pfade wurden daher ausschließlich PCs eingesetzt.

5.2 Steigerung des Durchsatzes

Die Steigerung des Durchsatzes ist eines der Kernziele dieser Arbeit. Sie ist maximal, wenn das Routingverfahren den optimalen Pfad nutzt. Die Ermittlung des optimalen Pfades kann von den entwickelten Verfahren jedoch nicht garantiert werden. Es wurden daher Messungen durchgeführt, die anhand des ermittelten Durchsatzes die Optimalität der Pfadsuche bewerten.

5.2.1 Versuchsaufbau

Die Steigerung des Durchsatzes lässt sich nur ermitteln, wenn zusätzlich Messungen eines Einkanal-Routingverfahrens vorliegen. Es war naheliegend, hierzu das originale AWDS7 einzusetzen. Da AWDS standardmäßig im 2.4 GHz-Band eingesetzt wird, wurde es auf die Nutzung von Kanal 8 eingestellt. Außerdem wurde der Durchsatz bestimmt, der sich durch die Nutzung des optimalen Pfades erreichen lässt. Die Ermittlung des optimalen Pfades erforderte den Einsatz einer Brute-Force-Suche auf der vollständigen Topologie, welche in Kapitel 4.4.6 kurz vorgestellt wurde.

Damit der Pfadsuche möglichst viele Alternativen bereitstehen, fanden die Messungen in einer größeren Topologie statt. Abbildung 5.1 zeigt die eingesetzte Topologie mit 13 Teilnehmern und 106 realen, unidirektionalen Kanten. Die Vernetzung auf den beiden Funkkanälen ist durch rote (Kanal 8) und grüne Linien (Kanal 36) gekennzeichnet. Da auch im späteren Anwendungsfeld die Kommunikation durch Hindernisse beeinträchtigt wird, sind die Teilnehmer auf verschiedene Räume verteilt, so dass die Funksignale mehrere Wände durchdringen müssen.

Der Austausch des Routingverfahrens erforderte einen manuellen Eingriff auf allen Teilnehmern, so dass die Messung insgesamt mehrere Stunden andauerte. Es wurde beobachtet, dass sich nach dem Neustart der Teilnehmer die natürlichen Topologien teilweise voneinander unterschieden. Durch das srcfitler-Modul musste daher sichergestellt werden, dass die Topologie bei allen Versuchen konstant bleibt. Vor jeder Messreihe wurde die Topologie unter Zuhilfenahme des topolock-Moduls fixiert, um keine weiteren Veränderungen zu erlauben.

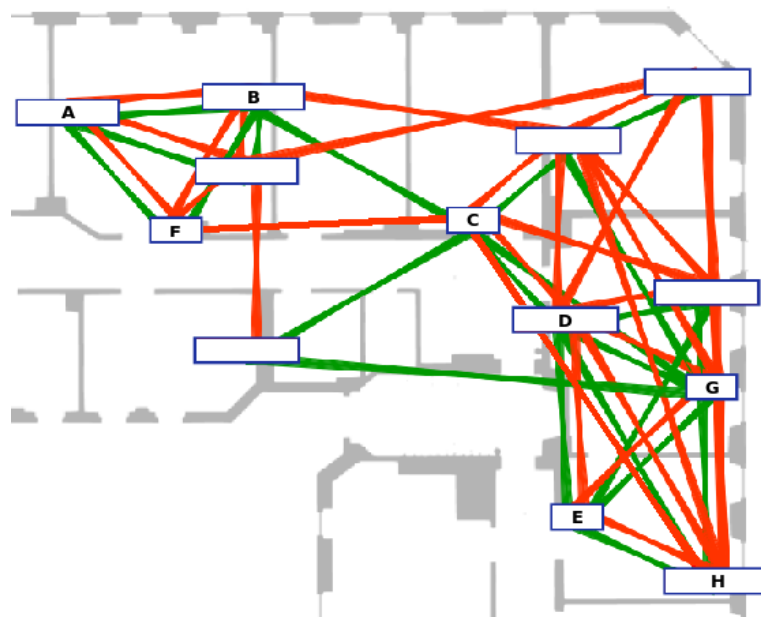


Abbildung 5.1: Topologie während der Durchsatzmessungen.

Die Ermittlung des erzielten Durchsatzes erfolgte als Messung eines einzelnen TCP-Datenstromes über mehrere Pfadlängen zwischen einem und vier Hops. Sie wurde mit dem UNIX-Programm `iperf` für eine Dauer von jeweils fünf Minuten durchgeführt. Tabelle 5.1 gibt die Sender und Empfänger für die verschiedenen Pfadlängen an.

Pfadlänge (Hops)	Sender	Empfänger
1	A	B
2	A	C
3	A	D
4	A	E

Tabelle 5.1: Start- und Ziel-Teilnehmer für verschiedene Pfadlängen.

Damit die ermittelten Durchsätze unterschiedlicher Pfadlängen besser verglichen werden können, wurden sie auf die Kapazität der Verbindung skaliert. Dies entspricht einer Transformation in die Nutzung der verfügbaren Kapazität.

Die Kapazität einer Verbindung bezeichnet den maximal möglichen Durchsatz, der auf ihr erzielt werden kann. Sie hängt von den Interferenzreichweiten der einzelnen Teilnehmer sowie der genutzten Senderaten ab. Weil die Teilnehmer eine automatische Ratenanpassung durchführen und die Interferenzbereiche durch Hindernisse beeinflusst werden, ist die Bestimmung der Kapazität allerdings sehr schwierig. Sie wurde deshalb über die Summe der Durchsätze abgeschätzt, die mit dem Einkanal-Verfahren auf beiden Kanälen einzeln erreicht wurden.

Diese Näherung über die Summe der geschätzten Durchsätze ist aus mehreren Gründen nicht optimal. Zum Einen ist die zugrunde liegende ETX-Metrik für eine Schätzung des Durchsatzes nur bedingt geeignet. Sie bewertet lediglich die Anzahl der Paketwiederholungen und kann gute Pfade nicht von sehr guten Pfaden unterscheiden. Zum anderen wird die Kapazität für zwei einzelne Pfade geschätzt, die den Funkkanal beibehalten. Der von den Mehrkanal-Verfahren ermittelte Pfad wird voraussichtlich die Funkkanäle wechseln und kann dadurch über andere Teilnehmer verlaufen.

5.2.2 Erwartungen

Bei Übertragungen über einen Hop ist zu erwarten, dass alle Verfahren etwa 50% der Kapazität nutzen, weil sie nur einen Funkkanal verwenden können. Das Einkanal-Verfahren sollte auch bei allen anderen Pfadlängen maximal 50% der Kapazität nutzen. Ab einer Pfadlänge von zwei Hops sind die Mehrkanal-Verfahren in der Lage, beide Funkkanäle zu verwenden. Sofern stets der optimale Pfad gefunden wird, werden sie die verfügbare Kapazität voll ausnutzen. Besonders bei größeren Pfadlängen ist jedoch davon auszugehen, dass das Verfahren der diagonalen Links keine optimalen Pfade findet. Da es zur Vorauswahl der Heuristik eingesetzt wird, wird die Heuristik stets

gleichwertige oder bessere Ergebnisse liefern. Das Brute-Force-Verfahren sollte bei allen Pfaden mit mindestens zwei Hops die volle Kapazität erreichen, weil es stets den optimalen Pfad findet.

5.2.3 Auswertung

Die Ergebnisse der Messungen werden in Abbildung 5.2 dargestellt. Die erreichte Nutzung der Kapazität liegt bei einer Pfadlänge von einem Hop zwischen 45% und 57%. Das schlechtere Abschneiden des Einkanal-Verfahrens ist durch die Nutzung von Kanal 8 begründet (siehe Tabelle 5.2). Denn durch die automatische Ratenanpassung besaßen die Verbindungen im 2.4GHz-Band eine niedrigere Datenrate, als solche im 5GHz-Band. Die drei Mehrkanal-Verfahren verwendeten hingegen Kanal 36 und sollten theoretisch eine identische Nutzung der Kapazität erreichen, die gemessenen Abweichungen entstehen durch natürliche Schwankungen der Verbindungsqualitäten. Die Anwendung des Verfahrens der diagonalen Links erreicht eine 18-prozentige Steigerung, die Heuristik eine 20-prozentige Steigerung gegenüber dem Einkanal-Verfahren.

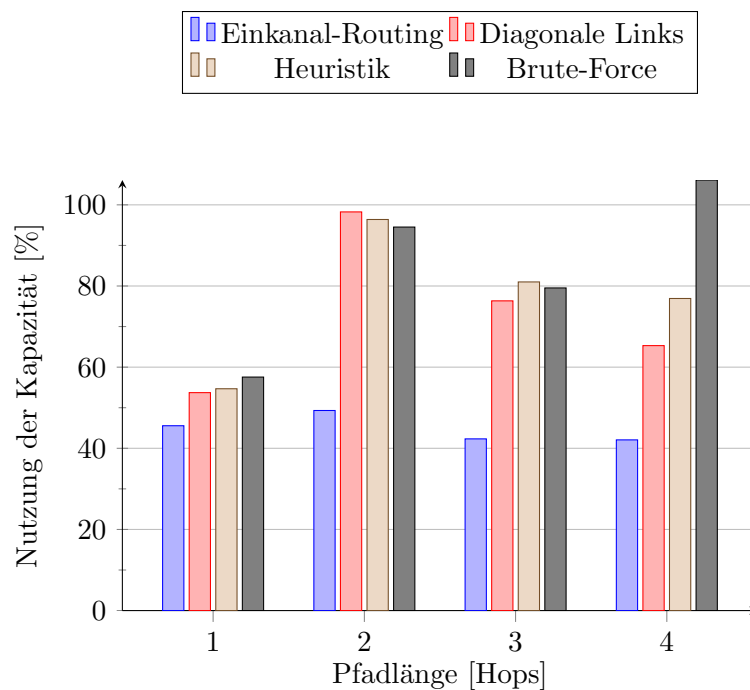


Abbildung 5.2: Vergleich der Nutzung der Verbindungskapazität durch verschiedene Routingverfahren.

Bei einer Pfadlänge von zwei Hops erreicht das Einkanal-Verfahren 49% der maximalen Kapazität. Die Mehrkanal-Verfahren erreichen hingegen weitaus höhere Werte bis zu

Verfahren	Pfad
Einkanal-Routing	A → B
Diagonale Links	A → B
Heuristik	A → B
Brute-Force	A → B

Tabelle 5.2: Ermittelte optimale Pfade mit einer Länge von einem Hop.

98%. Sie nutzen jedoch einen anderen Pfad als das Einkanal-Verfahren (Tabelle 5.3). Offensichtlich ist der Pfad A-B-C besser als A-F-C. Da die Teilstrecke B-C jedoch nur auf Kanal 36 verfügbar ist, kann sie vom Einkanal-Verfahren nicht genutzt werden. Alle drei Mehrkanal-Verfahren nutzen den gleichen Pfad, so dass die gemessenen Unterschiede erneut auf Schwankungen der Verbindungsqualitäten zurückzuführen sind. Die ermittelte Durchsatzsteigerung der diagonalen Links beträgt 98%, die der Heuristik 95%.

Verfahren	Pfad
Einkanal-Routing	A → F → C
Diagonale Links	A → B → C
Heuristik	A → B → C
Brute-Force	A → B → C

Tabelle 5.3: Ermittelte optimale Pfade mit einer Länge von zwei Hops.

Bei einer Pfadlänge von drei Hops nutzt das Einkanal-Verfahren die verfügbare Kapazität lediglich zu 42%. Wie Tabelle 5.4 zu entnehmen ist, weichen die Pfade der Mehrkanal-Verfahren voneinander ab. Sie enthalten zwar die gleichen Teilnehmer, nutzen auf dem letzten Hop aber unterschiedliche Funkkanäle. Das Verfahren der diagonalen Links und die Heuristik nutzen dazu Kanal 8, während der von der Brute-Force-Verfahren ermittelte Pfad Kanal 36 benutzt. Diese Tatsache ist sehr überraschend, denn das Brute-Force-Verfahren findet stets die optimale Kanalkombination durch das Netzwerk. Diese Kombination hätte auch von der Heuristik ermittelt werden müssen, weil sie im vorausgewählten Teilnehmer-Pfad A-B-C-D enthalten ist. Wahrscheinlich unterscheiden sich die beiden ermittelten Pfade nur minimal, so dass während des Wechsels des Routingverfahrens bereits kleine Schwankungen der Verbindungsqualität zu einer Änderung des optimalen Pfades führten. Diese Annahme wird durch die Beobachtung bestärkt, dass die gemessene Nutzung der Kapazität bei allen drei Mehrkanal-Verfahren ungefähr identisch ist.

Dass die ermittelten Anteile der genutzten Kapazität nur bei 80% liegen, ist auf die grobe Schätzung der Kapazität zurückzuführen. Denn zumindest das Brute-Force-Verfahren sollte auf drei Hops die Kapazität zu 100% nutzen. Festzuhalten ist jedoch,

5 Evaluierung

dass der durch die beiden anderen Mehrkanal-Verfahren ermittelte Pfad in etwa gleichwertig mit dem optimalen Pfad ist. Die erzielte Durchsatzsteigerung der diagonalen Links beträgt 80%, die Heuristik erreicht eine Steigerung von 91%.

Verfahren	Pfad
Einkanal-Routing	A → F → C → D
Diagonale Links	A → B → C → D
Heuristik	A → B → C → D
Brute-Force	A → B → C → D

Tabelle 5.4: Ermittelte optimale Pfade mit einer Länge von drei Hops.

Tabelle 5.5 zeigt die ermittelten Pfade für vier Hops. Interessant ist, dass die Heuristik eine andere Kanalkombination wählt, als das Verfahren der diagonalen Links, das zur Vorauswahl eingesetzt wird. Sie kann dadurch die Kapazität um 11% besser nutzen. Das Brute-Force-Verfahren zeigt jedoch, dass der optimale Pfad statt über Teilnehmer D über Teilnehmer G verläuft. Das Verfahren der diagonalen Links und die Heuristik können dadurch nur 65% bzw. 76% der vollen Kapazität nutzen. Dies entspricht einer Durchsatzsteigerung von 55% bzw. 83%.

Der Anteil der genutzten Kapazität durch das Brute-Force-Verfahren von 106% verdeutlicht erneut die Ungenauigkeit der Kapazitätsschätzung.

Verfahren	Pfad
Einkanal-Routing	A → F → C → H → E
Diagonale Links	A → B → C → D → E
Heuristik	A → B → C → D → E
Brute-Force	A → B → C → G → E

Tabelle 5.5: Ermittelte optimale Pfade mit einer Länge von vier Hops.

Die Messungen bestätigen, dass sich durch die Nutzung mehrerer Funkkanäle tatsächlich eine bessere Nutzung der verfügbaren Kapazität erreichen lässt. Bei Pfadlängen von zwei und drei Hops wurde die maximale Nutzung der verfügbaren Kapazität erreicht, was einer Durchsatzsteigerung von bis zu 98% gegenüber der Nutzung eines einzigen Funkkanals entsprach.

Die Analyse der optimalen Pfade zeigt, dass das Verfahren der diagonalen Links tatsächlich auf jedem Hop den Funkkanal wechselt. Der dadurch entstandene Pfad ist jedoch nicht immer optimal. So wurde beobachtet, dass das heuristische Verfahren den diagonalen Links überlegen ist, wenn der ermittelte Pfad durch die Analyse aller Kanalkombinationen noch weiter optimiert werden kann. Weiterhin wurde ersichtlich, dass die Optimalität der Heuristik entscheidend von der Vorauswahl abhängt. Befin-

det sich der optimale Pfad nicht im eingeschränkten Suchraum, kann die verfügbare Kapazität nicht ausgereizt werden.

5.3 Dauer der Pfadsuche

Während der Durchsatzmessungen wurde die Topologie fixiert, um identische Bedingungen für alle Messreihen zu schaffen. Tatsächlich kann die Topologie aufgrund schwankender Signalstärken im realen Einsatz jedoch ständigen Veränderungen unterworfen sein. Dies kann Auswirkungen auf aktive Pfade haben, deren Verlauf im Anschluss neu ermittelt werden muss. Eine wichtige Bedingung an die Pfadsuche ist eine kurze Berechnungsdauer, damit mehr Zeit für die eigentlichen Datenübertragungen zur Verfügung steht. Aus diesem Grund wurde die Dauer der Pfadsuche der bereits eingesetzten Verfahren untersucht.

5.3.1 Versuchsaufbau

Es wurde die gleiche Topologie wie zur Durchsatzmessung eingesetzt (siehe Abbildung 5.1). Sie wird vom Einkanal-Verfahren mit 13 Knoten und 58 Verbindungen modelliert. Die Mehrkanal-Verfahren müssen 26 Knoten und 106 reale Verbindungen verwalten, beim Verfahren der diagonalen Links kommen zusätzlich 56 diagonale Verbindungen hinzu. Die Messungen wurden auf einem eingebetteten Gerät mit ALIX-Architektur durchgeführt, da diese Geräte die typischen Teilnehmer späterer Anwendungen sind. Für jedes der untersuchten Routingverfahren wurde die Dauer ermittelt, die Teilnehmer F benötigt, um einen Pfad zu Teilnehmer E zu bestimmen. Die Ermittlung der Dauer erfolgte durch die eingesetzten Routingverfahren selbst, indem die Differenz der Zeitstempel zwischen Beginn und Ende der Pfadsuche über das shell-Modul ausgegeben wurde.

5.3.2 Erwartungen

Das Einkanal-Verfahren verwendet lediglich ein Shortest-Path-Verfahren zur Pfadsuche. Zusätzlich ist die zu speichernde Topologie relativ klein, so dass die Pfadsuche hier am kürzesten dauern wird.

Das Verfahren der diagonalen Links setzt ebenfalls ein Shortest-Path-Verfahren ein, allerdings separat für jeden der zwei eigenen Knoten. Zusätzlich enthält die Topologie etwa doppelt so viele Knoten und dreimal so viele Kanten, so dass der Aufwand gegenüber dem Einkanal-Verfahren deutlich steigen wird. Eine Schätzung des Mehraufwandes kann über die Berechnung der Komplexität des eingesetzten Dijkstra-Verfahrens erfolgen, die im Falle der in AWDS genutzten Prioritätswarteschlange für zu bearbeitende Knoten $O(n + m \log n)$ beträgt [Ney]. Dabei bezeichnet n die Anzahl der Knoten und m die Anzahl der Kanten. In der zugrundeliegenden Topologie

entsteht durch die höhere Knoten- und Kantenzahl ein Mehraufwand von etwa dem 7-fachen. Da das Dijkstra-Verfahren nur kürzeste Entfernungen bestimmt, müssen im Anschluss die nächsten Hops ermittelt werden, so dass der Aufwand sogar noch größer sein wird.

Die Heuristik nutzt zur Vorauswahl das Verfahren der diagonalen Links und sucht zusätzlich die optimale Kanalkombination auf dem Teilgraphen heraus. Bei einer Pfadlänge von vier Hops muss sie dazu einen Entscheidungsbaum mit 31 Knoten aufbauen. Der Aufwand der Pfadsuche der Heuristik wird daher minimal über dem Aufwand der diagonalen Links liegen.

Das Brute-Force-Verfahren skaliert exponentiell mit der Größe der Topologie. Die Topologie mit 26 Knoten und 106 Verbindungen wird bereits groß genug sein, um den Aufwand der anderen Verfahren um Größenordnungen zu übersteigen.

5.3.3 Auswertung

Die Messergebnisse werden in Abbildung 5.3 dargestellt. Das Einkanal-Verfahren benötigte mit 0.122 Millisekunden mit Abstand die geringste Zeit für die Pfadsuche. Die Dauer der Pfadsuche im Verfahren der diagonalen Links setzt sich aus der Zeit für jeweils eine Shortest-Path-Suche pro eigenem Knoten zusammen. Da beide Shortest-Path-Durchläufe auf der gleichen Topologie arbeiten, dauern sie etwa gleich lang. Insgesamt dauert die Pfadsuche der diagonalen Links 5.354 Millisekunden, was etwa einer 44-fachen Steigerung gegenüber der Pfadsuche des Einkanal-Verfahrens entspricht. Diese Steigerung fällt stärker als zunächst erwartet aus, dies ist jedoch durch die konkrete Implementierung bedingt. Aufgrund der speziellen Modellierung des Topologie-Graphen kann AWDS das Dijkstra-Verfahren nicht optimal implementieren. So müssen beispielsweise alle Knoten vor Beginn des Dijkstra-Verfahrens in eine andere Datenstruktur kopiert werden, auf der die eigentliche Pfadsuche stattfindet. Dies kostet zusätzliche Zeit. Die nachträgliche Ermittlung der nächsten Hops benötigt ebenfalls eine gewisse Rechenzeit bei und fällt bei den diagonalen Links aufgrund der Unterscheidung von Teilnehmern und Knoten sogar etwas komplexer aus.

Die Heuristik übernimmt von den diagonalen Links die Dauer der beiden Shortest-Path-Durchläufe, die für die Vorauswahl benötigt werden. Die Ermittlung der optimalen Kanalkombination auf dem vorausgewählten Teilnehmer-Pfad dauerte lediglich 0.692 Millisekunden.

Das Brute-Force-Verfahren vergleicht die Kanalkombinationen auf allen möglichen Teilnehmer-Pfaden zum Ziel. In der gewählten Topologie existierten 1484 Teilnehmer-Pfade von F nach E, deren Ermittlung und Analyse 10.0370 Sekunden benötigte.

Die Messergebnisse bestätigen alle getroffenen Erwartungen. Sie zeigen, dass die Einkanal-Pfadsuche mit dem geringsten Aufwand umgesetzt wird. Der Aufwand der diagonalen Links und der Heuristik ist zwar bereits weitaus größer, die Dauer der Pfadsuche bewegt sich mit wenigen Millisekunden allerdings noch im Rahmen der praktikablen

5.4 Einfluss der Wichtung zwischen Latenz und Durchsatz

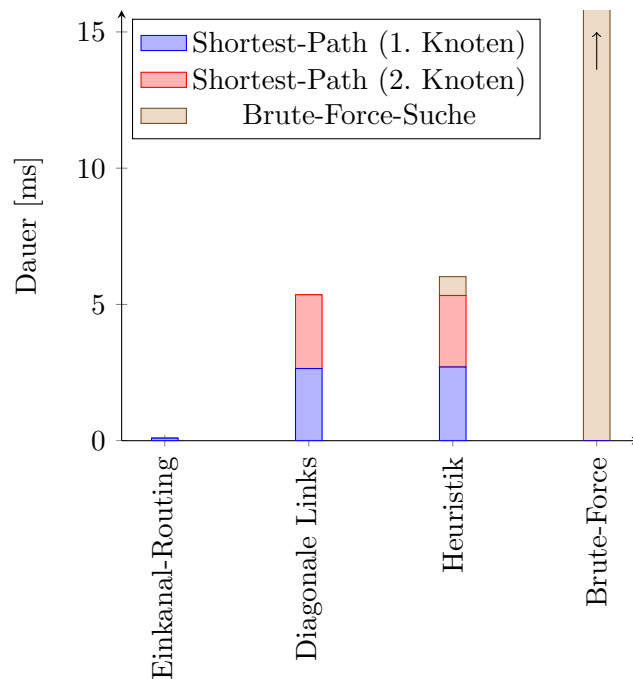


Abbildung 5.3: Dauer der Pfadsuche.

Durchführbarkeit. Dies ist bei der Suche nach dem optimalen Pfad, d.h. einer Brute-Force-Suche auf dem kompletten Topologie-Graphen, nicht mehr der Fall.

5.4 Einfluss der Wichtung zwischen Latenz und Durchsatz

Die Heuristik und das Brute-Force-Verfahren sehen die Manipulation der Pfadsuche durch den Wichtungsparemeter β vor. Er kann dazu genutzt werden, um die Präferenz zwischen hohem Durchsatz oder geringer Latenz anzugeben. An dieser Stelle soll überprüft werden, ob eine Manipulation des Wichtungsparemters tatsächlich Auswirkungen auf die Pfadsuche hat und ob die Eigenschaften des optimalen Pfades den theoretischen Betrachtungen entsprechen.

5.4.1 Versuchsaufbau

Der Einfluss von β verändert die Präferenzen der Pfadsuche. Es ist daher sinnvoll, dem Routingverfahren eine möglichst große Anzahl an Pfaden bereitzustellen. Dazu wurde die in Abschnitt 5.2 beschriebene Topologie erneut eingesetzt. Die Messungen erfolgten zwischen Teilnehmer A und Teilnehmer E, das heißt auf einer Pfadlänge von

5 Evaluierung

mindestens vier Hops. Um keine weiteren Einschränkungen der Pfade vorzunehmen, wurde das Brute-Force-Verfahren eingesetzt.

Es wurden Pfade für $\beta = \{0, 0.3, 0.5, 0.7, 1.0\}$ hinsichtlich Durchsatz und Latenz untersucht. Dazu wurde jeweils der erreichte TCP-Durchsatz mit `iperf` gemessen. Abbildung 5.4 zeigt den schematischen Aufbau der Latenzmessung. Die Teilnehmer A und E sind über Gigabit-Ethernet an eine separate Instanz, den *TimestampServer*, angeschlossen. Sobald Teilnehmer A ein Paket an E sendet, benachrichtigt er den TimestampServer. Ebenso benachrichtigt Teilnehmer E den TimestampServer beim Empfang des Paketes. Der TimestampServer wertet die Empfangszeitpunkte der Nachrichten beider Teilnehmer aus und bildet deren Differenz, um die Latenz des Mesh-Pfades zu erhalten. Es wurde festgestellt, dass der Latenzunterschied der beiden Ethernetverbindungen unter 0.1 Millisekunden liegt. Er beeinflusst die Berechnung der Latenz des Mesh-Netzwerks somit nur minimal und wurde nicht weiter berücksichtigt.

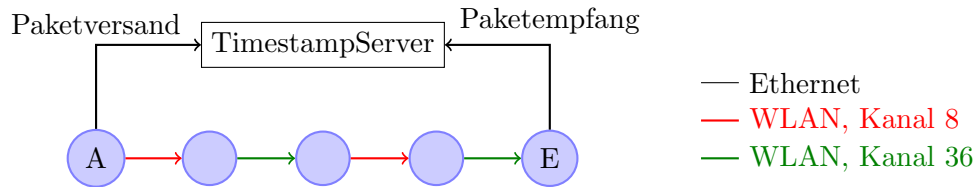


Abbildung 5.4: Schematische Darstellung der Latenzmessung.

Jede Messung erfolgte über einen Zeitraum von fünf Minuten. Die Ergebnisse wurden über den Durchschnitt der Messwerte bestimmt.

5.4.2 Erwartungen

Bei $\beta = 0$ wird der Pfad vollständig auf die Latenz optimiert. Daher ist zu erwarten, dass die Latenz in diesem Fall am geringsten ist. Mit steigendem β sinkt der Einfluss der Latenz auf die Optimalität des Pfades, dafür wird der Einfluss des Durchsatzes immer stärker. Deswegen werden Latenz und Durchsatz mit wachsendem β ebenfalls steigen. Bei $\beta = 1$ ist der maximale Durchsatz zu erwarten.

5.4.3 Auswertung

Abbildung 5.5 zeigt die Ergebnisse der Messungen. Die Wahl des optimalen Pfades kann Tabelle 5.6 entnommen werden. Bei $\beta = 0$ ist die Optimierung nach minimaler Latenz sehr gut zu erkennen, denn der optimale Pfad nutzt lediglich Kanal 36. Offensichtlich sind Verbindungskosten auf Kanal 36 durchweg geringer, als die Kosten auf Kanal 8. Mit steigendem β enthält der Pfad mehr und mehr Kanalwechsel, so

5.4 Einfluss der Wichtung zwischen Latenz und Durchsatz

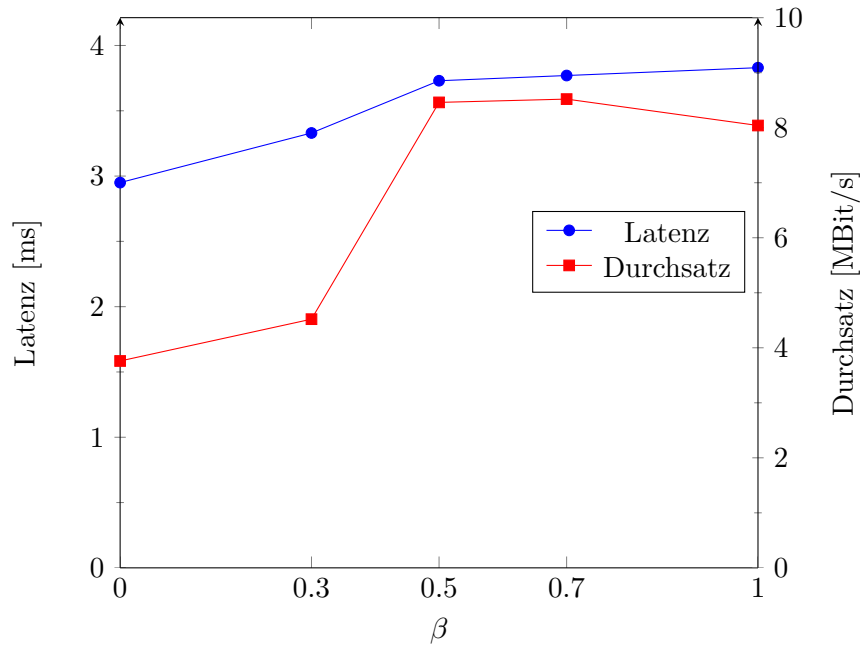


Abbildung 5.5: Einfluss von β .

dass Durchsatz und Latenz ebenfalls ansteigen. Ab $\beta = 0.5$ ändert sich der Pfad nicht mehr, folglich bleiben Latenz und Durchsatz etwa konstant.

β	Pfad
0.0	A → B → C → D → E
0.3	A → B → C → D → E
0.5	A → B → C → D → E
0.7	A → B → C → D → E
1.0	A → B → C → D → E

Tabelle 5.6: Ermittelte optimale Pfade für verschiedene Werte von β .

Die Messergebnisse zeigen, dass der Wichtungsparameter β einen Einfluss auf die Wahl des optimalen Pfades hat. Es wurde festgestellt, dass sowohl die Latenz als auch der erreichbare Durchsatz des Pfades von β in der gewünschten Form abhängt.

Interessant ist jedoch die Beobachtung, dass bei einem Wichtungsparameter von 0.5 bereits der Pfad mit dem maximalen Durchsatz ermittelt wurde. Dies ist auf die Verwendung der ETX-Metrik zurückzuführen, welche die Pfade lediglich nach ihren Verlustraten bewertet und zur Schätzung des Durchsatzes nur bedingt geeignet ist.

5.5 Verhalten im Fehlerfall

Die entwickelten Routingverfahren haben zum Ziel, eine Steigerung des Durchsatzes zu ermöglichen, ohne dass darunter die Zuverlässigkeit leiden muss. Deswegen soll das Verhalten des Netzwerks im Fehlerfall analysiert werden. Die in Abschnitt 3.1.1 getroffenen Entscheidungen zur Erhöhung der Redundanz sind allen entwickelten Mehrkanal-Verfahren gemein. Es genügt daher, lediglich eines der Verfahren zu testen.

5.5.1 Versuchsaufbau

Es wurden Messungen des effizienten und des zuverlässigen Modus mit dem Verfahren der diagonalen Links durchgeführt. Dazu wurde die Rate der gesendeten und empfangenen Unicast-Pakete eines Datenstroms über zwei Hops ermittelt und analysiert. Da im effizienten Modus die Senderate mit der Empfangsrate nahezu übereinstimmt, genügte die Analyse der Empfangsrate. Die Ermittlung der gesendeten bzw. empfangenen Pakete erfolgte direkt innerhalb des Routingverfahrens. Die Empfangsrate im zuverlässigen Modus zählt nur diejenigen Pakete, die zum ersten Mal empfangen und an die Anwendung weitergegeben werden.

Weil in dieser Messung die Optimalität des Pfades nicht im Vordergrund steht, genügt es, nur wenige Alternativen bereitzuhalten. Daher wurde unter Zuhilfenahme des `srfilter`-Moduls eine kettenförmige Topologie aus drei PCs aufgebaut, wobei auf allen Verbindungen Sichtkontakt besteht.

Es wurde eine Störung von Kanal 36 durch einen externen Teilnehmer simuliert, die bis zum Ende der jeweiligen Messung bestehen blieb. Der `madwifi`-Treiber bietet die Möglichkeit, durch die Missachtung des Medienzugriffsverfahrens permanent fehlerhafte Pakete zu verschicken. Dieses Verhalten wird durch die Eingabe von

```
# iwpriv <Schnittstelle> txcont 1
```

aktiviert. Andere Teilnehmer ermitteln dann ständig ein Medium in Benutzung und können ihre eigenen Pakete auf dem entsprechenden Kanal nicht versenden.

5.5.2 Erwartungen

Im effizienten Modus ist davon auszugehen, dass der Pfad unterschiedliche Funkkanäle auf beiden Hops nutzt. Eine Störung von Kanal 36 bedeutet daher eine Störung des Pfades, so dass bis zur Erkennung und Neuberechnung keine Pakete verschickt werden können.

Der zuverlässige Modus sendet Pakete redundant über beide Funkkanäle. Da eine Störung von Kanal 36 die Datenübertragung auf Kanal 8 nicht beeinflusst, ist kein Abfall der Paket-Empfangsrate zu erwarten.

5.5.3 Auswertung

Abbildung 5.6 zeigt für beide Modi die eingesetzten Topologien und die ermittelten Paketraten, das Blitzsymbol markiert den Beginn der Kanalstörung.

Auf der linken Seite wird die Messung des effizienten Modus dargestellt. Dort schwankt die Empfangsrate im fehlerfreien Betrieb etwa um 1500 Pakete pro Sekunde. Ab dem Zeitpunkt der Kanalstörung fällt die Empfangsrate für 15 Sekunden auf 0. Nachdem ein neuer Pfad bestimmt wurde, steigt die Empfangsrate wieder an, erreicht mit etwa 750 Paketen pro Sekunde aber nur die Hälfte des Ausgangswertes.

Die Zeitspanne zwischen dem Auftreten der Störung und der Ermittlung eines neuen Pfades ist auffällig groß. Sie setzt sich aus der Dauer der Fehlererkennung und der Dauer der Pfadsuche zusammen. Da bereits gezeigt wurde, dass die Pfadsuche nur wenige Millisekunden benötigt, ist die Dauer der Fehlererkennung die kritische Komponente. Diese lässt sich auf mehrere Arten optimieren. So flutet das neue AWDS8 inzwischen Topologie-Pakete nicht nur periodisch durch das Netzwerk, sondern auch direkt nach einer Veränderung der Topologie. Eine weitere Verbesserung ist derzeit in der Erprobung. Sie sieht die Signalisierung von Paket-Zustellungsfehlern durch die MAC-Schicht vor, so dass bereits bei Übertragungsfehlern einiger weniger Datenpakete ein Verbindungsabbruch angenommen werden kann. Eine derartige Fehlererkennung benötigt nur noch etwa 200 Millisekunden.

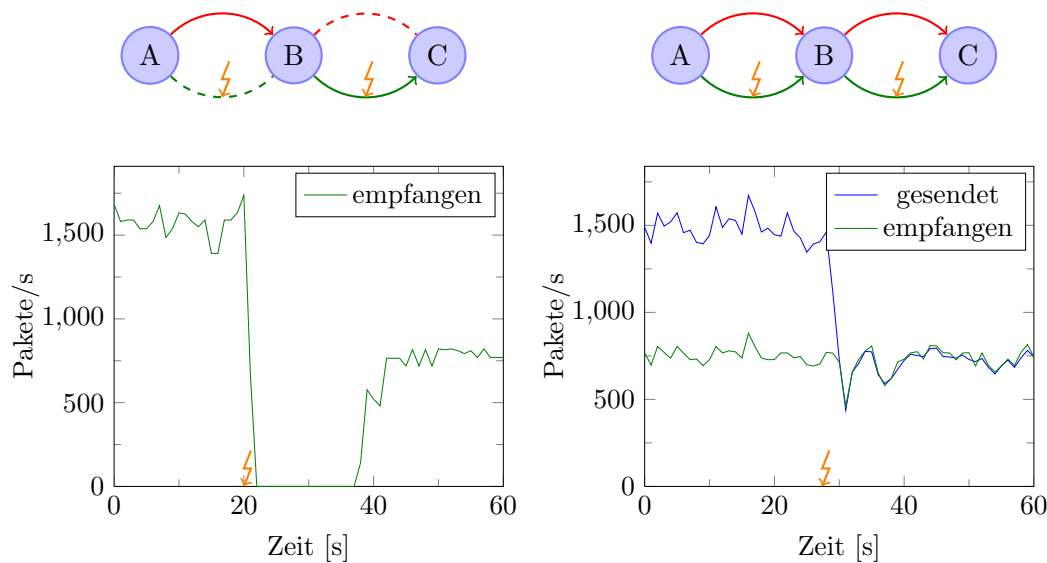


Abbildung 5.6: Auswirkung einer Kanalstörung auf eine Datenübertragung im effizienten Modus (links) und im zuverlässigen Modus (rechts).

Die rechte Seite zeigt die Messungen des zuverlässigen Modus. Die Senderate der Pakete beträgt im fehlerfreien Fall etwa 1500 Pakete/s, da jedes Paket doppelt verschickt wird ist die Netto-Paketrate aber nur halb so groß. Die Empfangsrate der Pakete beträgt etwa 750 Pakete/s, weil alle Kopien des Paketes zuvor verworfen werden. Im

fehlerfreien Fall entspricht die Empfangsrate des zuverlässigen Modus etwa der halben Empfangsrate des effizienten Modus. Dies war zu erwarten, da im zuverlässigen Modus die Teilnehmer A und B um das Medium konkurrieren.

Die Rate gesendeter Pakete halbiert sich ab dem Zeitpunkt der Kanalstörung, da auf Kanal 36 keine Pakete mehr verschickt werden können. Dagegen ist die Empfangsrate über die gesamte Messdauer weitgehend konstant, weil die Pakete auf Kanal 8 auch während der Kanalstörung im Empfänger eintreffen. Dadurch wird die laufende Datenübertragung nicht unterbrochen. Nach 31 Sekunden ist ein kurzer Einbruch der Empfangsrate zu verzeichnen. Er entsteht durch die überfüllte Paketwarteschlange von Kanal 8, welche die interne Paketverarbeitung des Routingverfahrens kurz ausbremst. Bei 37 Sekunden ist ein weiterer kleinerer Einbruch der Empfangsrate zu erkennen. Zu diesem Zeitpunkt wurde eine Neuberechnung der Pfade durchgeführt, die aufgrund der Veränderung der Topologie erforderlich wurde.

5.6 Diskussion der Ergebnisse

Die Untersuchung des Verhaltens der Datenübertragungen im Fehlerfall bestätigte, dass durch die Nutzung von Mehrkanal-Verfahren das Netzwerk selbst bei einem Kanalausfall nicht zum Erliegen kommt. Je nach Wahl des Übertragungsmodus besteht sogar die Möglichkeit, dass Datenströme zwischen den Anwendungen durch Störungen nicht unterbrochen werden. Von diesem Gesichtspunkt aus eignen sich beide entwickelten Verfahren für den industriellen Einsatz.

Werden die diagonalen Links als Vorauswahl-Verfahren der Heuristik eingesetzt, findet diese stets gleichwertige oder bessere Pfade. Der dazu erforderliche Mehraufwand wächst zwar exponentiell mit der Länge des vorausgewählten Pfades, ist bei kurzen Pfaden aber noch sehr gering. Die Heuristik bietet außerdem die Möglichkeit, über den Wichtungspareter β die gewünschten Eigenschaften des Pfades zu bestimmen.

Das Fazit der Untersuchungen lautet daher, dass die Heuristik aufgrund der besseren Pfadsuche und höheren Flexibilität gegenüber dem Verfahren der diagonalen Links zu bevorzugen ist. Anders als beim Verfahren der diagonalen Links ist bei der Heuristik zusätzlich davon auszugehen, dass die Optimalität des Pfades mit steigender Anzahl der Funkkanäle nicht immer weiter sinkt. Diese Vermutung ist jedoch rein theoretisch, sie konnte mangels entsprechender Hardware nicht experimentell bestätigt werden.

6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden zwei Routingverfahren für drahtlose Mehrkanal-Mesh-Netzwerke entwickelt, welche die zusätzlich vorhandene Kapazität zur Durchsatzsteigerung einsetzen, ohne dabei die Zuverlässigkeit des Netzwerks einzuschränken. Dazu werden die Verbindungen des Netzwerks redundant ausgelegt, indem jeder Teilnehmer die gleichen Funkkanäle nutzt.

Mehrkanal-Netzwerke erlauben die Erhöhung des Durchsatzes durch die Verringerung konkurrierender Medienzugriffe. Dazu müssen die Teilnehmer auf unterschiedlichen Funkkanälen senden. Bei der Pfadwahl für einen Multi-Hop-Datenstrom muss daher darauf geachtet werden, dass dessen einzelne Abschnitte möglichst gleichmäßig auf die verfügbaren Funkkanäle verteilt sind. Die Suche eines derart optimalen Pfades ist jedoch zu aufwändig, so dass ein Kompromiss aus Laufzeit und Optimalität gefunden werden muss. Dieser Kompromiss wird von den beiden vorgestellten Verfahren auf unterschiedliche Weise eingegangen.

Das Verfahren der diagonalen Links sieht die Pfadsuche mittels eines Shortest-Path-Verfahrens vor. Damit Kanalwechsel entlang eines Pfades bevorzugt werden, werden der Topologie zusätzliche Verbindungen mit geringen Kosten hinzugefügt. Sollen Pakete über diese Verbindung geschickt werden, muss im Sender zuvor der Kanal gewechselt werden. Das heuristische Verfahren nutzt dagegen eine angepasste Pfadsuche. Es sieht die Minimierung des Suchraums vor, indem ein Großteil der möglichen Pfade bereits im Vorfeld ausgeschlossen wird. Die eigentliche Pfadsuche erfolgt durch die Ermittlung und Analyse aller verbleibenden Möglichkeiten.

Das abschnittsweise Versenden von Daten über alle Funkkanäle hat zur Folge, dass die Störung eines beliebigen Kanals die laufende Datenübertragung so lange unterbricht, bis ein alternativer Pfad ermittelt wird. Für besonders sensible Anwendungen ist daher auch die Möglichkeit vorgesehen, die Nutzdaten auf allen Funkkanälen gleichzeitig zu verschicken. Dabei kann zwar keine Durchsatzsteigerung erreicht werden, die Übertragung ist jedoch wesentlich robuster gegen Störungen.

In einer realen Testumgebung wurde die korrekte Funktionsweise der prototypischen Implementierungen beider Verfahren nachgewiesen. Es wurde festgestellt, dass sich durch die Nutzung zweier Funkkanäle der Durchsatz gegenüber Einkanal-Routing auf etwa das Doppelte steigern ließ. Dies erfordert die Nutzung des optimalen Pfades, dessen Ermittlung jedoch nicht garantiert werden kann. Die Messungen zeigten weiterhin, dass die Heuristik keine schlechteren Pfade als das Verfahren der diagonalen Links findet und gleichzeitig flexibler konfigurierbar ist.

Weiterhin wurde bestätigt, dass mit beiden Verfahren die Kommunikation auch bei einer Störung eines kompletten Funkkanals fortgeführt werden kann.

Ausblick

Während der Ausarbeitung des Konzepts und der Evaluierung der Verfahren wurden einige Möglichkeiten zur Verbesserung und Weiterentwicklung identifiziert, deren Verwirklichung über den Rahmen dieser Arbeit hinausgeht.

Die zur Verfügung gestellte Redundanz des Netzwerks dient lediglich als Basis für übergeordnete Verfahren, welche die Echtzeitanforderungen der Anwendungen tatsächlich zusichern können. Eine konsequente Weiterführung dieser Arbeit ist daher die Entwicklung eines Dienstegüte-Moduls, das die geforderten Anforderungen durch die Reservierung von Pfaden garantiert. Durch eine geeignete Pfadwahl für verschiedene Datenströme ließe sich sogar das Problem beheben, dass die verfügbare Kapazität auf einem Hop nur zur Hälfte genutzt werden konnte.

Während der Durchsatzmessungen über eine Pfadlänge von vier Hops trat der Fall auf, dass durch die diagonalen Links der optimale Pfad aus der Vorauswahl der Heuristik entfernt wurde. Eine interessante Fragestellung ist, ob andere Mehrkanal-Verfahren mit statischer Kanaluweisung für den Einsatz in der Vorauswahl besser geeignet sind. Zur Vergleichbarkeit der Ergebnisse müsste dazu eine weitere Evaluierung durchgeführt werden.

Ebenso wurde festgestellt, dass der Aufwand des Brute-Force-Schrittes auf kurzen Pfaden noch sehr gering ist. Es wäre denkbar, dass der Suchraum nicht ganz so stark eingeschränkt wird, um die Möglichkeit zu erhöhen, den optimalen Pfad nicht auszuschließen. Eine mögliche Umsetzung wäre die Beschränkung der Topologie auf mehrere Teilnehmer-Pfade, aus deren Kanalkombinationen die beste ausgewählt wird. Die Größe des Suchraums könnte sogar abhängig von der Pfadlänge gewählt werden, so dass bei kurzen Pfadlängen viele Pfade und bei sehr großen Pfadlängen nur wenige Pfade zu untersuchen sind.

Literaturverzeichnis

- [awd] *AWDS - Wirelss Multi-Hop Routing*. <http://awds.berlios.de/>. – letzte Begutachtung: 18.12.2008
- [BC56] BELLMAN, R. ; CALIF, RAND CORP SANTA M.: *ON A ROUTING PROBLEM*. Defense Technical Information Center, 1956
- [BCD04] BAHL, P. ; CHANDRA, R. ; DUNAGAN, J.: SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In: *Proceedings of the 10th annual international conference on Mobile computing and networking* ACM New York, NY, USA, 2004, S. 216–230
- [Ben] BENVENISTE, Mathilde: *Description of the CCC MMAC protocol*. <https://mentor.ieee.org/802.11/file/05/11-05-0666-00-000s-description-ccc-mmacc-protocol.doc>. – letzte Begutachtung: 18.12.2008
- [Bic05] BICKET, John C.: *Bit-rate Selection in Wireless Networks*. February 2005
- [CKV06] CHEREDDI, C. ; KYASANUR, P. ; VAIDYA, N.H.: Design and implementation of a multi-channel multi-interface network. In: *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality* ACM New York, NY, USA, 2006, S. 23–30
- [DABM03] DOUGLAS, SJ ; AGUAYO, D. ; BICKET, J. ; MORRIS, R.: A High-Throughput Path Metric for Multi-Hop Wireless Routing. In: *New York: Mobile Computing and Networking 23* (2003)
- [Dij59] DIJKSTRA, E.W.: A note on two problems in connexion with graphs. In: *Numerische Mathematik* 1 (1959), Nr. 1, S. 269–271
- [DPZ04] DRAVES, Richard ; PADHYE, Jitendra ; ZILL, Brian: Routing in multi-radio, multi-hop wireless mesh networks. In: *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA : ACM, 2004. – ISBN 1-58113-868-7, S. 114–128
- [FJ56] FORD JR, LR: NETWORK FLOW THEORY. (1956)
- [Gä01] GÄRTNER, Felix C.: *Formale Grundlagen der Fehlertoleranz in verteilten Systemen*, Technische Universität Darmstadt, Diss., March 2001

- [HM05] HERMS, André ; MAHRENHOLZ, Daniel: *Unified Development and Deployment of Network Protocols*. <http://www-ivs.cs.uni-magdeburg.de/EuK/forschung/publikationen/pdf/2005/gea.pdf>. Version: July 2005. – letzte Begutachtung: 18.12.2008
- [J⁺06] JOSHI, A. u. a.: HWMP Specification. In: *IEEE P802. 11 Wireless LANs, document IEEE 802.11-06/1778r1* (2006)
- [JM96] JOHNSON, D.B. ; MALTZ, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE* (1996), S. 153–179
- [Kya06] KYASANUR, Pradeep: *Multichannel Wireless Networks: Capacity and Protocols*, Diss., August 2006
- [LBDC⁺01] LI, J. ; BLAKE, C. ; DE COUTO, D.S.J. ; LEE, H.I. ; MORRIS, R.: Capacity of Ad Hoc wireless networks. In: *Proceedings of the 7th annual international conference on Mobile computing and networking* ACM New York, NY, USA, 2001, S. 61–69
- [LC] LOSTANLEN, Y. ; CONE, Y.: *Studies on indoor propagation at various frequencies for radio local networks*
- [Ney] NEY, I.H.: Algorithmen und Datenstrukturen.
- [pcea] *PC Engines ALIX System Boards*. <http://www.pcengines.ch/alix.htm>. – letzte Begutachtung: 18.12.2008
- [pceb] *PC Engines WRAP System Boards*. <http://www.pcengines.ch/wrap.htm>. – letzte Begutachtung: 18.12.2008
- [Pfe08] PFEIFFER, Johannes: *Bottleneck analysis of AWDS*. Teamprojekt an der Otto-von-Guericke-Universität Magdeburg, 2008
- [PR99] PERKINS, C.E. ; ROYER, E.M.: Ad-hoc on-demand distance vector routing. In: *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications* Bd. 2 Feb, 1999, S. 90–100
- [RC05] RANIWALA, A. ; CHIUEH, T.: Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In: *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* Bd. 3, 2005
- [RGC04] RANIWALA, Ashish ; GOPALAN, Kartik ; CHIUEH, Tzi cker: Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 8 (2004), Nr. 2, S. 50–65. – ISSN 1559–1662
- [Sch08] SCHOLZ, Ronny: Erweiterung der AWDS Software und des AWDS-Protokolls zur parallelen Nutzung von zwei Funkkanälen. (2008)
- [SLL95] STEPANOV, A.A. ; LEE, M. ; LABORATORIES, Hewlett-Packard: *The Standard Template Library*. (1995)

- [tou] *Touchgraph library*. <http://sourceforge.net/projects/touchgraph>. – letzte Begutachtung: 18.12.2008
- [Wew06] WEWIOR, Martin: *Bandbreitenkontrolle in drahtlosen Multihop-Netzwerken*, Universität Magdeburg, Diplomarbeit, 2006. http://www-ivs.cs.uni-magdeburg.de/EuK/lehre/diplomarbeiten/dipl_wewior.pdf. – Online-Ressource. – letzte Begutachtung: 18.12.2008
- [Wie07] WIENÖBST, Jens: *Vergleich von Routingmetriken in drahtlosen Maschennetzwerken*, Universität Magdeburg, Diplomarbeit, 2007. <http://ivs.cs.uni-magdeburg.de/EuK/lehre/diplomarbeiten/wienoeb.pdf>. – Online-Ressource. – letzte Begutachtung: 18.12.2008
- [YWK] YANG, Y. ; WANG, J. ; KRAVETS, R.: Interference-aware Load Balancing for Multihop Wireless Networks / Technical Report UIUCDCS. – Forschungsbericht
- [YWK05] YANG, Y. ; WANG, J. ; KRAVETS, R.: Designing routing metrics for mesh networks. In: *IEEE Wireless Mesh Networks (WiMesh)* (2005)

Literaturverzeichnis

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Abschlussarbeit "Effizientes Mehrkanal-Routing" selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Ronny Scholz
Magdeburg, den 19. Dezember 2008