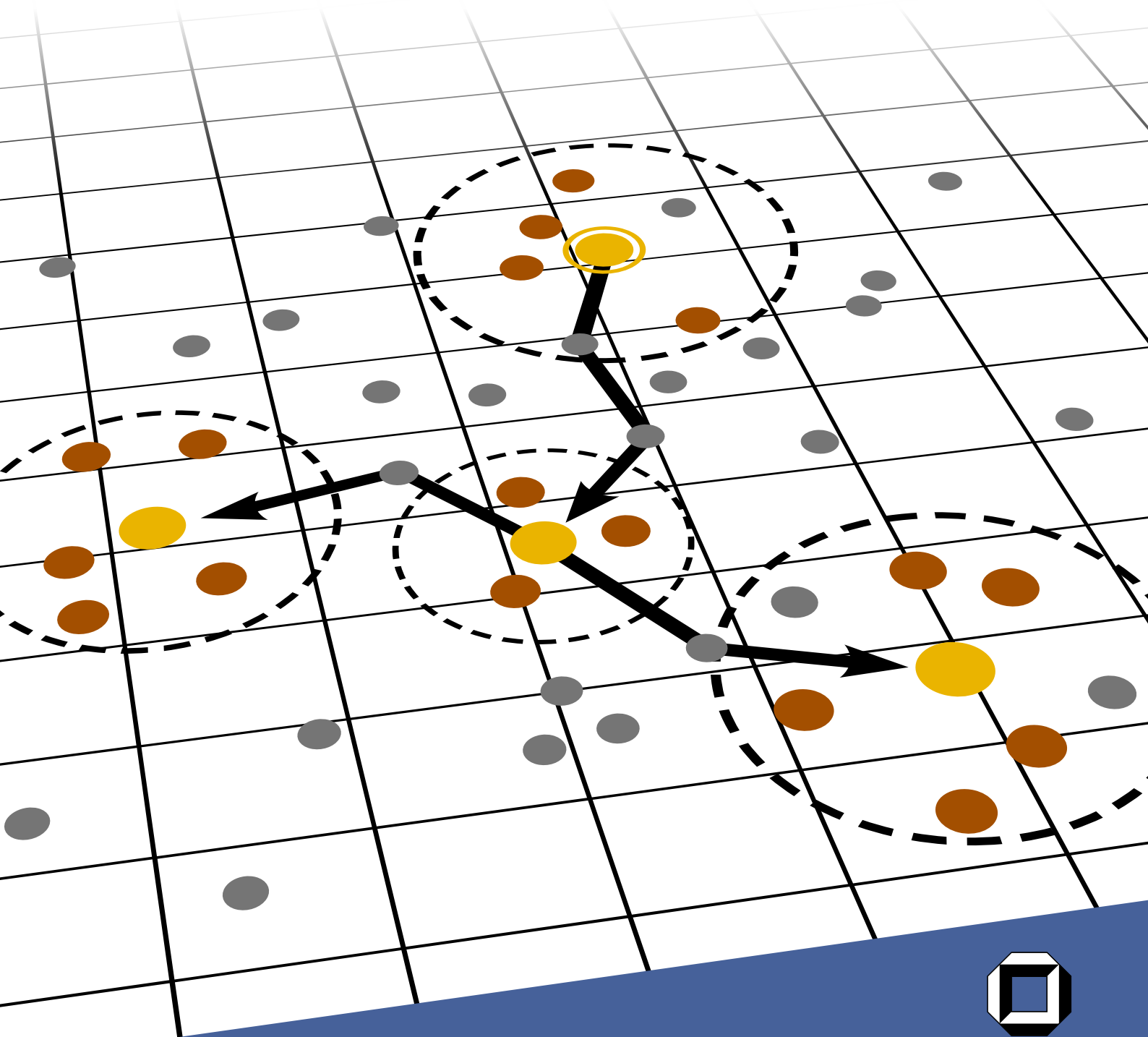


Peter Baumung

P2P-basierte Gruppenkommunikation in drahtlosen Ad-hoc-Netzen



Peter Baumung

P2P-basierte Gruppenkommunikation in drahtlosen Ad-hoc-Netzen

P2P-basierte Gruppenkommunikation in drahtlosen Ad-hoc-Netzen

von
Peter Baumung



universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH)
Fakultät für Informatik, 2008

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2008
Print on Demand

ISBN: 978-3-86644-268-9

P2P-basierte Gruppenkommunikation in drahtlosen Ad-hoc-Netzen

zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN

der Fakultät für Informatik
der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

Dissertation

von

Dipl.-Inform. Peter Baumung

aus Bruchsal

Tag der mündlichen Prüfung: 14. April 2008

Erster Gutachter: Prof. Dr. Martina Zitterbart
Universität Karlsruhe (TH)

Zweiter Gutachter: Prof. Dr. Hannes Hartenstein
Universität Karlsruhe (TH)

Karlsruhe, im Juni 2008

*F*rau Prof. Dr. Martina Zitterbart legte den Grundstein dieser Arbeit, indem sie nicht nur das Referat übernahm sondern mich auch an ihrem Lehrstuhl beschäftigte: Für die Betreuung und die wirklich einmalige Zeit am ITM bedanke ich mich herzlichst!

Herr Prof. Dr. Hannes Hartenstein erklärte sich trotz außerordentlicher beruflicher Anspannungen bereit, das Korreferat dieser Arbeit zu übernehmen: Meinen ausdrücklichen Dank hierfür!

Kollegen, Freunde und Familie schufen wiederum ein einzigartiges Umfeld, indem sie Abwechslung in Zeiten geistiger Anstrengungen boten. Sie waren es, die letztendlich den entscheidenden Ausschlag für den Erfolg der Arbeit gegeben haben: Meinen tiefsten Dank! Euch allen!

Peter Baumung



Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung und Annahmen	4
1.2	Gliederung	6
2	Grundlagen	7
2.1	Drahtlose Ad-hoc-Netze	7
2.1.1	Das Medienzugriffsprotokoll IEEE 802.11	12
2.1.1.1	Senden und Empfangen von Funksignalen	12
2.1.1.2	Die Koordination der Medienzugriffe	13
2.1.1.3	Die Sicherung von Unicast-Dateneinheiten	14
2.1.1.4	Die RTS/CTS-Erweiterung	17
2.1.2	Wegwahl in drahtlosen Ad-hoc-Netzen	19
2.1.2.1	Ad-hoc On Demand Distance Vector Routing	20
2.1.2.2	Optimized Link-State Routing	23
2.1.3	Diskussion drahtloser Ad-hoc-Netze	24
2.2	P2P-basierte Gruppenkommunikation	26
2.2.1	Allgemeine Prinzipien	28
2.2.2	Das Overlay-Netz	28
2.2.2.1	Gruppenverwaltung	30
2.2.2.2	Paketduplizierung und -Weiterleitung	30
2.2.3	Zuverlässigkeit	31
2.2.4	Diskussion von P2P-Multicast	33
2.3	Zusammenfassung	34

3	P2P-Multicast-Protokolle in drahtlosen Ad-hoc-Netzen	37
3.1	Anforderungen und Bewertungskriterien	39
3.1.1	Die Suche nach vorhandenen Gruppenmitgliedern	39
3.1.2	Die Eingliederung in die Overlay-Topologie	40
3.1.3	Die Anpassung der Overlay-Topologie	41
3.1.4	Die Flexibilität der Overlay-Topologie	42
3.1.5	Die Konsistenz des Overlay-Routing	43
3.1.6	Die Anzahl unterhaltener Overlay-Verbindungen	46
3.1.7	Die induzierte Netzbelastung	46
3.1.8	Die Nutzung des Semi-Broadcast-Mediums	47
3.1.9	Implizites und explizites Routing	47
3.1.10	Die Stabilität der Multicast-Bäume	48
3.1.11	Die Ausgliederung aus der Overlay-Topologie	49
3.1.12	Die Reaktivität gegenüber Ausfällen	50
3.2	Metriken zur Verbindungsbewertung	52
3.2.1	Die eindeutige Wiedergabe des Netzzustands	52
3.2.2	Die implizierte Belastung des Netzes	53
3.3	Zuverlässigkeit und Staukontrolle	53
3.3.1	TCP in drahtlosen Ad-hoc-Netzen	54
3.3.2	Zuverlässigkeitsmechanismen	55
3.3.3	Staukontrolle	56
3.4	Eigenschaften und Bewertung existierender P2P-Multicast-Protokolle . .	56
3.4.1	Aufbau und Anpassung der Overlay-Topologie	58
3.4.2	Der Gruppenbeitritt	67
3.4.3	Die Vermittlung von Datenpaketen	71
3.4.4	Die Ausgliederung von Gruppenmitgliedern	77
3.4.5	Die Nutzung der Semi-Broadcast-Eigenschaft	80
3.4.6	Zuverlässigkeit und Staukontrolle	80
3.5	Zusammenfassung	81

4	TrAM: Ein leichtgewichtiges Overlay für drahtlose Ad-hoc-Netze	83
4.1	Aufbau und Erhalt der Overlay-Topologie	84
4.1.1	Das Entstehen des TrAM-Baums	84
4.1.2	Die Bewertung umliegender Gruppenmitglieder	85
4.1.3	Die Suche und Auswahl des Elternknotens	89
4.1.4	Die Anmeldung bei einem Elternknoten	94
4.1.5	Die Abmeldung bei einem Elternknoten	95
4.1.6	Der Wechsel des Elternknotens	95
4.1.7	Das Verschieben der Wurzel des TrAM-Baums	97
4.1.8	Ausfälle von Gruppenmitgliedern	98
4.1.9	Gruppenaustritt	100
4.2	Das Routing	101
4.2.1	Sonderfälle	101
4.3	Bewertung und Diskussion	102
4.3.1	Aufbau und Erhalt der Overlay-Topologie	102
4.3.2	Der Gruppenbeitritt	105
4.3.3	Die Vermittlung von Datenpaketen	105
4.3.4	Die Ausgliederung von Gruppenmitgliedern	107
4.3.5	Die Nutzung der Semi-Broadcast-Eigenschaft	107
5	Verbesserungen von P2P-Multicast-Protokollen	109
5.1	Generische Verbesserungsmechanismen	109
5.1.1	Das lokale Broadcast-Clustering	109
5.1.1.1	Die lokale Auflösung der Overlay-Topologie	110
5.1.1.2	Das Bootstrapping	110
5.1.1.3	Der Versand und die Vermittlung von Daten	111
5.1.1.4	Die Unterstützung von Mobilität	111
5.1.1.5	Zustandsübergänge und Zustandsautomat	114
5.1.1.6	Eigenschaften lokaler Gruppenmitglieder	116

5.1.1.7	Diskussion und Bewertung	116
5.1.2	Die Stabilisierung des Routing	120
5.1.3	Die initiale Nachbarsuche	122
5.2	Protokollspezifische Modifikationen	123
5.2.1	NICE	123
5.2.1.1	Der Gruppenbeitritt	123
5.2.1.2	Das Aufspalten eines Cluster	124
5.2.1.3	Optimierungsmetrik	124
5.2.2	Narada	124
5.2.2.1	Die initiale Nachbarsuche	124
5.2.2.2	Aggregation von Topologie- und Routing-Information . .	124
5.2.2.3	Intelligent Probing	125
5.2.3	PAST-DM	125
5.2.3.1	Source-Routen-Konstruktion	125
5.2.3.2	Globale Optimierung der Overlay-Topologie	126
5.3	Zusammenfassung	126
6	Die MAAM-Architektur – Konzepte und Umsetzungen	127
6.1	Ziele und Anforderungen	128
6.2	Der Aufbau der MAAM-Architektur	129
6.2.1	Allgemeines zur Funktionsweise	130
6.2.1.1	Agenten	130
6.2.1.2	Paketverwaltung	131
6.2.1.3	Der Prozess zur Behandlung von Datenpaketen	132
6.2.2	Die Integration von Overlay-Topologien	133
6.2.3	Die Integration von Zuverlässigkeitsmechanismen	134
6.2.4	Die Integration lokaler Broadcast-Cluster	135
6.2.4.1	Die Erweiterung der MulticastBase	136
6.2.4.2	Die Erweiterung der TransportBase	138

6.3	Die Dienstkomposition	140
6.3.1	Ohne Verwendung des LBC-Konzepts	140
6.3.2	Mit Verwendung des LBC-Konzepts	140
6.4	Versand und Weiterleitung von Daten	141
6.4.1	Die Verarbeitung eines versendeten Datenpakets	141
6.4.2	Die Verarbeitung eines empfangenen Datenpakets	143
6.5	Die Kapselung des Basissystems	145
6.5.1	Die Wrapper-Schnittstelle	146
6.5.1.1	Umsetzungen für Linux und Windows	146
6.5.1.2	Die Netzwerksimulationsumgebung GloMoSim	150
6.5.2	Die Timeout-Schnittstelle	151
6.5.3	Das ApplicationCore-Modul	151
6.6	Zusammenfassung	152
7	Protokollauswertung	153
7.1	Konfiguration der Simulationsumgebung	153
7.1.1	Allgemeine Simulationsparameter	154
7.1.2	Betrachtete Szenarien	154
7.1.2.1	Mobilitätsmodellierung	154
7.1.2.2	Bewegungsszenarien	156
7.1.2.3	Anwendungsszenarien	157
7.2	Definitionen gemessener Werte	158
7.3	Auswertung der Overlay-Topologien	159
7.3.1	Standard P2P-Multicast	160
7.3.1.1	Kontrollfluss und Medienzugriffszeit	160
7.3.1.2	Topologieverhalten	164
7.3.2	Auswirkungen der Kombination mit LBCs	167
7.3.2.1	Kontrollfluss und Medienzugriffszeit	167
7.3.2.2	Topologieverhalten	169

7.4	Auswertung einer CBR-Anwendung	172
7.4.1	Standard P2P-Multicast	173
7.4.1.1	Zustellraten und Latenzen	173
7.4.1.2	Stabilität der Overlay-Topologien	175
7.4.1.3	Medienzugriffszeit und verworfene Pakete	176
7.4.2	Auswirkungen der Kombination mit LBCs	178
7.4.2.1	Zustellraten und Latenzen	178
7.4.2.2	Stabilität der Overlay-Topologien	179
7.4.2.3	Medienbelastung und verworfene Pakete	179
7.4.3	Hohe Datenraten mit TrAM und Narada	181
7.5	Auswertung einer Chat-Anwendung	183
7.5.1	Standard P2P-Multicast	185
7.5.1.1	Zustellraten und Latenzen	185
7.5.1.2	Belastung des Netzes und Paketverwürfe	186
7.5.2	Auswirkungen der Kombination mit LBCs	187
7.5.2.1	Zustellraten und Latenzen	188
7.5.2.2	Belastung des Netzes und Paketverwürfe	189
7.6	Auswertung von Mehrbenutzerspielen	189
7.6.1	Eine Single-Hop-Umgebung	191
7.6.2	Eine Multi-Hop-Umgebung	194
8	Zusammenfassung und Ausblick	201
8.1	Ergebnisse dieser Arbeit	202
8.2	Ausblick und weiterführende Arbeiten	204
A	Zusätzliche Protokolle und Mechanismen	207
A.1	Das n*Unicast-Protokoll	207
A.1.1	Gruppenverwaltung	208
A.1.2	Diskussion	208
A.2	Das MPGT-Protokoll	209

A.2.1	Die Aggregation von Datenpaketen	210
A.2.2	Einbettung von Verzögerungsinformation	210
A.2.3	Zuverlässigkeit	211
A.3	Bewertung von LBC-Repräsentanten	211
Literaturverzeichnis		213

Abbildungsverzeichnis

1.1	Drahtlose Kommunikation mit und ohne Infrastrukturunterstützung	2
1.2	Drahtlose Ad-hoc-Netze zur Unterstützung verteilter Anwendungen	3
1.3	Anforderungen an den Gruppenkommunikationsdienst	5
2.1	Einbezug von Endgeräten zur Weiterleitung von Datenpaketen	9
2.2	Einfluss von Mobilität auf Kommunikationswege	11
2.3	Übertragungs- und Erkennungsbereiche von WLAN-Funksignalen	14
2.4	Das CSMA/CA-Medienzugriffsverfahren auf Basis der DCF	17
2.5	Die Problematik versteckter und ausgelieferter Endgeräte	18
2.6	Routen-Aufbau bei dem AODV-Protokoll	22
2.7	Einfluss von Datenweiterleitungen auf entfernte Endgeräte	26
2.8	Verzögerungen während Wettbewerbssituationen bei IEEE 802.11	27
2.9	Aufbau eines Overlay-Netzes über einem zugrunde liegenden Netz	29
2.10	Datenweiterleitung über ein Overlay-Netz	32
3.1	Auswirkungen der Topologieanpassung bei Overlay-Netzen	42
3.2	Routing-Inkonsistenzen im Laufe einer Topologieanpassung	44
3.3	Fehlende Datenweiterleitung als Folge einer Topologieanpassung.	45
3.4	Medienbelegung bei einer Unicast-basierten Datenweiterleitung	48
3.5	Overlay-Partitionierung als Folge eines Gruppenaustritts	51
3.6	Das von Narada aufgebaute Mesh-förmige Overlay	59
3.7	Das NICE-Overlay auf Basis hierarchischer Cluster	62
3.8	Beispiel einer von PAST-DM aufgebauten Overlay-Topologie	64

3.9	Aufbau und Anpassung des ALMA-Overlay	68
3.10	Die Datenweiterleitung bei Narada	73
3.11	Die implizite Wegewahl über das NICE-Overlay	75
3.12	Subgruppenbildung im Laufe des Source-Routing bei PAST-DM	77
4.1	Kontrollpakete zum Aufbau und Erhalt des TrAM-Overlay.	85
4.2	Bewertung potenzieller Elternknoten bei TrAM	87
4.3	Zustandsautomat für die Suche nach einem Elternknoten.	90
4.4	Verarbeitung eines childHello-Pakets	91
4.5	Wechsel des Elternknotens bei TrAM	97
4.6	Möglichkeiten der Topologieanpassung bei TrAM	104
5.1	Nutzung des geteilten Mediums bei der Datenweiterleitung	112
5.2	Topologie eines LBC-erweiterten Overlay-Netzes	114
5.3	Der LBC-Zustandsautomat	115
5.4	Anzahl ausgebildeter LBCs bei bestimmter Gruppengröße	120
5.5	Wahrscheinlichkeit eines lokalen Gruppenbeitritts	121
6.1	Aufbau und Struktur der MAAM-Architektur	130
6.2	Vererbungsdiagramm für MAAM-Module	131
6.3	Versenden eines Datenpakets über die MAAM-Architektur	142
6.4	Empfangen eines Datenpakets über die MAAM-Architektur	144
6.5	Kapselung System-naher Funktionen in der MAAM-Architektur	147
6.6	Auf Basis der MAAM-Architektur entwickelte Anwendungen	149
7.1	Kontrollfluss und Medienzugriffszeit	161
7.2	Übertragungswiederholungen des Medienzugriffsprotokolls	163
7.3	Kontrollfluss und Medienzugriffszeit von Gruppengrößen	164
7.4	Verworfen Pakete	165
7.5	Kumulierte Strukturänderungen und Routen-Aufbauten	166
7.6	Strukturänderungen und Routen-Aufbauten	167

7.7	Kontrollfluss und Medienzugriffszeit (LBCs)	168
7.8	Kontrollfluss und Medienzugriffszeit von Gruppengrößen (LBCs)	170
7.9	Verworfen Pakete (LBCs)	170
7.10	Kumulierte Strukturänderungen und Routen-Aufbauten (LBCs)	171
7.11	Strukturänderungen und Routen-Aufbauten (LBCs)	173
7.12	Zustellraten und Latenzen (CBR)	174
7.13	Strukturänderungen (CBR)	176
7.14	Medienzugriffszeit (CBR)	177
7.15	Verworfen Pakete (CBR)	178
7.16	Zustellraten und Latenzen (CBR + LBCs)	180
7.17	Strukturänderungen (CBR + LBCs)	181
7.18	Medienzugriffszeit (CBR + LBCs)	182
7.19	Verworfen Pakete (CBR + LBCs)	183
7.20	Hohe Datenraten mit TrAM und Narada	184
7.21	Erzeugung künstlichen Chat-Verkehrs	186
7.22	Zustellraten und Latenzen (Chat)	187
7.23	Medienzugriffszeit und Routen-Aufbauten (Chat)	188
7.24	Verworfen Pakete (Chat)	189
7.25	Zustellraten und Latenzen (Chat + LBCs)	190
7.26	Medienzugriffszeit und Routen-Aufbauten (Chat + LBCs)	191
7.27	Verworfen Pakete (Chat + LBCs)	192
7.28	Single-Hop-Latenzen für Positionsinformation und Ereignisse	193
7.29	Single-Hop-Szenario bei Mehrbenutzerspielen	195
7.30	Szenario zur Modellierung eines verteilten Mehrbenutzerspiels	196
7.31	Multi-Hop-Szenario bei Mehrbenutzerspielen	197
7.32	Multi-Hop-Szenario bei Mehrbenutzerspielen (RTS/CTS)	198
7.33	Multi-Hop-Szenario bei Mehrbenutzerspielen (Tx-Anpassung)	199
A.1	Funktionen zur Bewertung von LBC-Repräsentanten.	212

Tabellenverzeichnis

2.1	Kostenvergleich von P2P-Multicast und Multicast-Routing	34
3.1	Bewertung von P2P-Multicast-Protokollen	58
3.2	Bei PAST-DM gehaltene Topologieinformation	65
3.3	Von Narada gehaltene Routing-Information	74
4.1	Bewertung von TrAM im Vergleich zu anderen Protokollen	103
5.1	Kombinierbarkeit von Overlay-Topologien mit LBCs	119
7.1	Konfigurationsparameter der Simulationsumgebung GloMoSim	155

Liste der Algorithmen

1	Auffindung eines Cluster-Repräsentanten bei NICE	69
2	Vergleich zweier Elternknoten bei TrAM	87
3	Entscheidung über den Wechsel des Elternknotens bei TrAM	88

1. Einleitung

Die Nutzung des Internets als Kommunikations- und Unterhaltungsmedium ist in den vergangenen Jahren zu einem wesentlichen Bestandteil unseres Alltags geworden: Das Kommunizieren via eMail, Instant-Messaging- oder VoIP-Anwendungen sowie das Surfen im WWW und die Teilnahme an Online-Mehrbenutzerspielen sind kaum noch aus unserer Gesellschaft wegzudenken. Ein entscheidender Trend, welcher seit einigen Jahren beobachtet werden kann, ist, dass die Nutzung der genannten Anwendungen nicht länger nur auf heimischen, fest installierten Arbeitsplatzrechnern stattfindet, sondern zunehmend auch auf kleinen, tragbaren Endgeräten, wie PDAs oder moderne Mobiltelefone. In der Tat ist die Entwicklung dieser Endgeräte in letzter Zeit rapide vorangeschritten, so dass sie nunmehr nicht nur über benutzerfreundliche Ein- und Ausgabemöglichkeiten verfügen: Vielmehr sind sie auch mit unterschiedlichen drahtlosen Kommunikationsschnittstellen ausgestattet, wie Bluetooth, IrDA, GSM oder WLAN. Diese Kommunikationsschnittstellen werden heutzutage genutzt, um einen drahtlosen Zugang zu einem Infrastruktur-basierten Netzwerk, etwa zu einer GSM-Basisstation oder einem WLAN-Hotspot, herzustellen und über dieses unterschiedliche Dienste (Telefonie, eMail, WWW) zu nutzen.

Die vorliegende Arbeit ist in einem noch sehr jungen Bereich der Mobilkommunikation, der so genannten *mobilen, drahtlosen Ad-hoc-Kommunikation*, angesiedelt. Diese Kommunikationsform beruht auf der Nutzung drahtloser Kommunikationsschnittstellen tragbarer Endgeräte, wobei die Kommunikation der Endgeräte jedoch *direkt* miteinander anstatt über fest installierte Basisstationen erfolgt. Wie in Abbildung 1.1 verdeutlicht ist, ermöglicht die Ad-hoc-Kommunikation durch Aufbau eines *mobilen, drahtlosen Ad-hoc-Netztes* die Kommunikation zwischen Benutzern selbst in Umfeldern, in

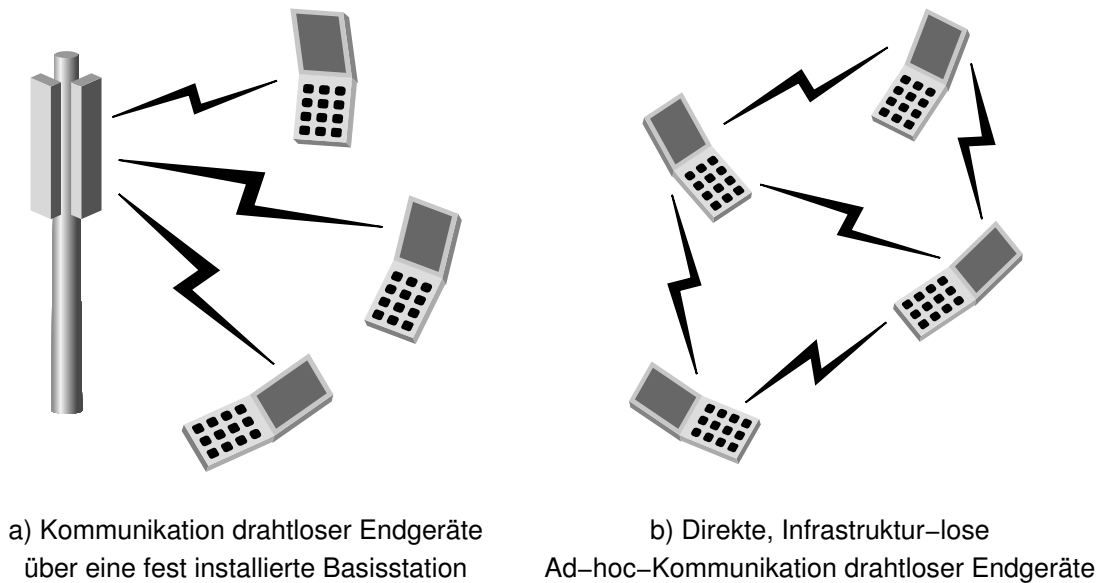


Abbildung 1.1 Drahtlose Endgeräte können über ein Infrastruktur-basiertes Netzwerk oder auch direkt miteinander kommunizieren.

welchen keinerlei Unterstützung durch zuvor installierte Infrastruktur stattfindet: Sie kann demnach an jedem beliebigen Ort erfolgen, an welchem sich drahtlose Endgeräte in gegenseitiger Übertragungreichweite befinden. Dies ist insbesondere aufgrund der Tatsache von Bedeutung, dass die Anzahl der sich in Umlauf befindenden drahtlosen Endgeräte seit Jahren einem kontinuierlichen Wachstum unterliegt und sich somit ein immer deutlicher werdendes Potenzial zur Ausbildung drahtloser Ad-hoc-Netze bietet.

Da sich die einstigen Annahmen, auf denen die im Internet heutzutage weitläufig eingesetzten Kommunikationsprotokolle basieren, nicht auf von Spontaneität, Mobilität und Dynamik geprägte Umfelder wie drahtlose Ad-hoc-Netze übertragen lassen, sind hier neue Protokollansätze und -konzepte gefragt. Obwohl sich die Forschungsgemeinde bereits seit einigen Jahren intensiv mit drahtlosen Ad-hoc-Netzen befasst, wurde mit der Standardisierung grundlegendster Kommunikationsprotokolle jedoch erst begonnen, so dass sich bislang noch keine tatsächlichen Anwendungen etabliert haben. Da sich Potenzial zur drahtlosen Ad-hoc-Kommunikation vor allem an Orten ergibt, an denen sich Gruppen von Benutzern mit gleichen Interessen zusammenfinden, geht man jedoch davon aus, dass eine so genannte *Gruppenkommunikation* ein bedeutender Bestandteil der drahtlosen Ad-hoc-Kommunikation sein wird. Bei dieser Kommunikationsform wird nicht mit lediglich einem sondern stattdessen mit einer ganzen Gruppe von Benutzern kommuniziert. Wie in Abbildung 1.2 dargestellt ist kann eine Gruppenkommunikation so beispielsweise der Unterstützung verteilter Anwendungen auf einem Universitätscampus dienen, indem ein zugrunde liegendes drahtloses Ad-hoc-Netz für den Datenaustausch

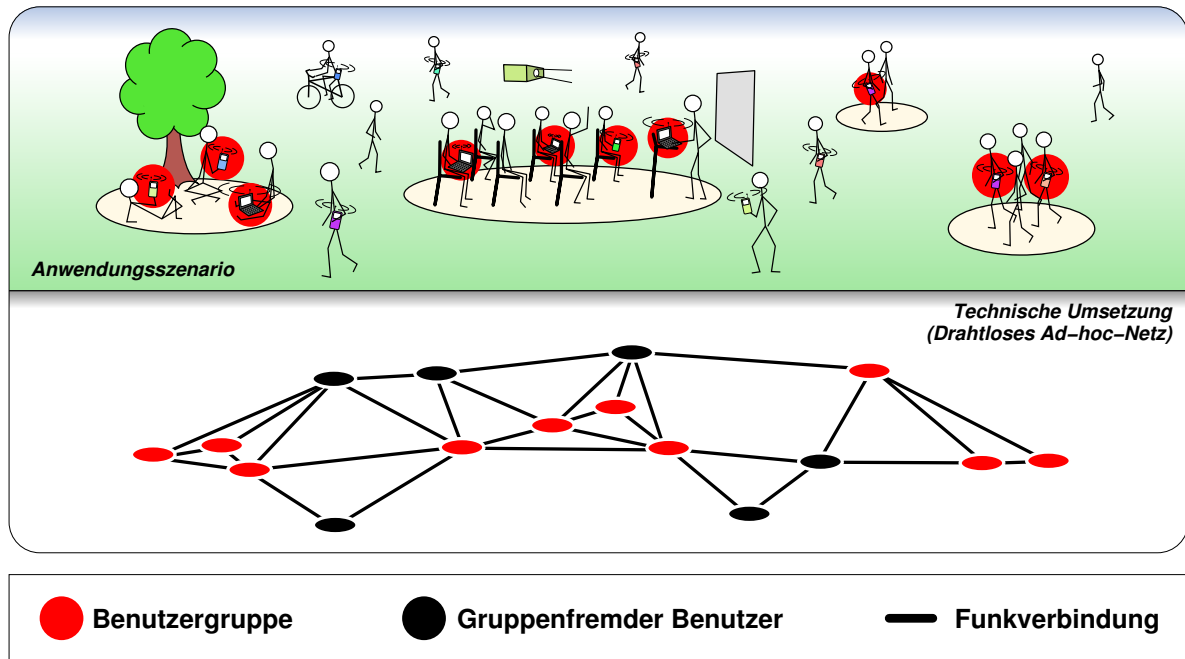


Abbildung 1.2 Unterstützung verteilter Anwendungen auf einem Universitätscampus anhand eines drahtlosen Ad-hoc-Netzes.

innerhalb einer gleich gesinnten Benutzergruppe eingesetzt wird. Andere Szenarien, in welchen sich der Einsatz einer Gruppenkommunikation anbietet, umfassen militärische oder Katastropheneinsätze aber auch Messe- und Touristenführungen.

Protokollkonzepte, welche nicht nur im Internet sondern auch bei der drahtlosen Ad-hoc-Kommunikation zur Umsetzung von Gruppenkommunikationsdiensten viel versprechend erscheinen, sind so genannte *Peer-to-Peer*-basierte Protokolle. Peer-to-Peer-basierte Gruppenkommunikationsprotokolle nutzen vergleichsweise einfache und bereits etablierte Punkt-zu-Punkt-Kommunikationsprotokolle, um zwischen den Mitgliedern einer Benutzergruppe (den so genannten Gruppenmitgliedern) ein logisches, dem physikalischen Netz überlagertes *Overlay-Netz* auszubilden. Dieses Overlay-Netz wird verwendet, um die von einem Benutzer ausgesendeten Daten an die übrigen Gruppenmitglieder zu verteilen. Im Vergleich zu bisherigen Ansätzen zur Gruppenkommunikation bieten Peer-to-Peer-basierte Gruppenkommunikationsprotokolle den Vorteil, dass die Protokollmechanismen lediglich den Gruppenmitgliedern und nicht allen Endgeräten innerhalb des drahtlosen Ad-hoc-Netzes bekannt sein müssen. Als Folge dessen erweisen sich diese Ansätze als besonders flexibel hinsichtlich der Berücksichtigung der von unterschiedlichen Anwendungen ausgehenden Anforderungen an die Datenverteilung: Während sich hier Unterschiede vor allem in Hinblick auf die im Zuge der Datenverteilung erfolgende *Wegewahl* und die *Zuverlässigkeit* der Datenverteilung zeigen, können die verwendeten Wegewahl- und Zuverlässigkeitsumsetzungen direkt *auf die betriebene An-*

wendung angepasst und in den Peer-to-Peer-basierten Gruppenkommunikationsdienst integriert werden.

1.1 Zielsetzung und Annahmen

Das Ziel dieser Arbeit ist der Entwurf und die Konzeption eines Peer-to-Peer-basierten Gruppenkommunikationsdienstes für drahtlose Ad-hoc-Netze, sowie dessen Evaluierung im Hinblick auf seine Machbarkeit. Unter dem Begriff des Peer-to-Peer-basierten Gruppenkommunikationsdienstes wird dabei ein Dienst verstanden, welcher zwischen einer Menge von Gruppenmitgliedern eine Gruppenkommunikation anhand eines Overlay-Netzes umsetzt und darüber hinaus die von der betriebenen Anwendung geforderte Zuverlässigkeit integriert. Zur Konzeption des Gruppenkommunikationsdienstes sollen an diesen die folgenden Anforderungen gestellt werden, welche auch in Abbildung 1.3 dargestellt sind:

– *Der effiziente Betrieb in drahtlosen Ad-hoc-Netzen:* Insbesondere die innerhalb des Dienstes erfolgende Gruppenkommunikation soll unter Berücksichtigung der Eigenschaften drahtloser Ad-hoc-Netze eine möglichst hohe Effizienz aufweisen: So sollen die in drahtlosen Ad-hoc-Netzen nur sehr begrenzt vorhandenen Ressourcen (wie beispielsweise Bandbreite und Stromversorgung) möglichst sparsam genutzt werden. Eine der herausragenden Eigenschaften dieser Netze ist darüber hinaus die potenzielle Mobilität der das Netz bildenden Endgeräte. Mobilität soll in dieser Arbeit ausdrücklich zugelassen und von der Gruppenkommunikation entsprechend unterstützt werden. Um einen effizienten Betrieb des Dienstes auch im Kontext der Mobilität zu gewährleisten, ist gegebenenfalls die Konzeption eines neuen Overlay-Netzes erforderlich, welches die von Benutzern ausgesendeten Daten unter Berücksichtigung der geforderten Effizienz an die Gruppenmitglieder verteilt.

– *Die Nutzung der Semi-Broadcast-Eigenschaft des geteilten Mediums:* Das drahtlose Kommunikationsmedium verfügt über eine so genannte Semi-Broadcast-Eigenschaft¹, aufgrund welcher Endgeräte auch nicht an sie gerichtete Funksignale “mithören” und empfangen können. Obgleich diese Eigenschaft bedeutende Nachteile mit sich bringt, birgt sie jedoch auch Potenzial zur Steigerung der Effizienz einer Gruppenkommunikation. Ein wichtiges Ziel des in dieser Arbeit zu konzipierenden Gruppenkommunikationsdienstes ist es daher, dieses Potenzial möglichst ausgiebig zu nutzen.

– *Die Flexibilität hinsichtlich Szenario und betriebener Anwendung:* Der zu konzipierende Gruppenkommunikationsdienst soll nicht auf den Einsatz mit bestimmten Anwendungen beschränkt sein. Obgleich in dieser Arbeit ein effizientes Overlay-Netz vorgeschlagen werden soll, soll der zu konzipierende Gruppenkommunikationsdienst nicht auf

¹Die Semi-Broadcast-Eigenschaft des drahtlosen Mediums wird in Kapitel 2.1 eingeführt und ausführlich beschrieben werden.

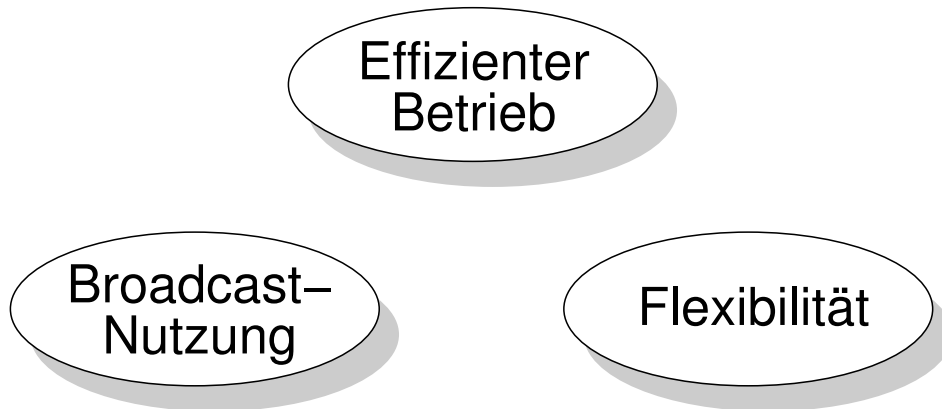


Abbildung 1.3 Anforderungen an den in dieser Arbeit konzipierten Gruppenkommunikationsdienst.

dieses Overlay-Netz zur Erbringung der Gruppenkommunikation fixiert werden. Ebenso soll der Dienst nicht auf eine bestimmte Zuverlässigkeitsumsetzung festgelegt werden. Stattdessen soll von dem Dienst eine Flexibilität in beiderlei Hinsicht (Overlay-Netz und Zuverlässigkeitsumsetzung) gewährleistet werden.

Weitere Anforderungen, die an den zu konzipierenden Gruppenkommunikationsdienst gestellt werden, sind:

- *Keine Unterstützung durch Positionsbestimmungssysteme:* Obwohl heutzutage Möglichkeiten zur Positionsbestimmung (wie etwa anhand von GPS-Empfängern) bereits in manche drahtlose Endgeräte integriert werden, kann das tatsächliche Vorhandensein solcher Systeme jedoch nicht bei allen Kommunikationspartnern vorausgesetzt werden. Aus diesem Grund soll der in dieser Arbeit zu entwickelnde Gruppenkommunikationsdienst explizit auf den Einsatz eines solchen Systems verzichten.

- *Anzahl der Kommunikationspartner:* Die Anzahl der durch den Dienst zu unterstützenden Gruppenmitglieder soll sich im Bereich von einigen Dutzend bis ca. 100 drahtlosen Endgeräten befinden.

Um diese Anforderungen zu erfüllen, sollen für die Umsetzung des Dienstes die folgenden Annahmen getroffen werden:

- *Begrenzung der Mobilität:* Auch wenn die Mobilität aller das Ad-hoc-Netz bildenden Endgeräte ausdrücklich unterstützt werden soll, so sollen die betrachteten Geschwindigkeiten der Endgeräte jedoch als gering angenommen und im Bereich von Fußgängergeschwindigkeiten angesiedelt werden.

- *Existenz einer Multi-Hop-Kommunikation für Unicast-Dateneinheiten:* Peer-to-Peer-basierte Protokolle setzen voraus, dass zwischen Endgeräten die Möglichkeit einer Punkt-zu-Punkt-Unicast-Kommunikation besteht. Insbesondere im Kontext drahtloser Ad-

hoc-Netze soll für diese Arbeit daher die Existenz eines so genannten Routing-Protokolls gefordert werden, welches eine Multi-Hop-Kommunikation zur Vermittlung von Unicast-Dateneinheiten zwischen drahtlosen Endgeräten zur Verfügung stellt.

– *Unterstützung der Semi-Broadcast-Fähigkeit durch die drahtlose Kommunikationsschnittstelle:* Obgleich es sich bei dem drahtlosen Medium prinzipiell um ein Semi-Broadcast-fähiges Medium handelt, wird die Nutzung dieser Eigenschaft nicht von allen drahtlosen Kommunikationsschnittstellen (wie etwa Bluetooth) unterstützt. In dieser Arbeit soll deshalb gefordert werden, dass eine solche Unterstützung tatsächlich vorliegt und Dateneinheiten somit als Broadcast zu allen sich in Übertragungsbereich befindenden Endgeräten übertragen werden können.

1.2 Gliederung

Zum besseren Verständnis der Arbeit erfolgt in Kapitel 2 zunächst eine Darstellung unterschiedlicher Grundlagen wie der Zugriff auf das drahtlose Medium sowie der Ablauf einer Kommunikation zwischen Endgeräten anhand drahtloser Ad-hoc-Netze. Darüber hinaus werden die Prinzipien der Peer-to-Peer-basierten Gruppenkommunikation kurz erläutert. Letztere werden in Kapitel 3 ausführlicher behandelt, wobei sowohl Anforderungen an eine in drahtlosen Ad-hoc-Netzen betriebene Overlay-Topologie erarbeitet als auch bereits bestehende Protokollkonzepte diskutiert und bewertet werden. Auf Basis der in diesem Kapitel erlangten Erkenntnisse befassen sich die Kapitel 4 und 5 mit der Konzeption verschiedener Protokollmechanismen: Diese umfassen einerseits ein neue Overlay-Topologie für drahtlose Ad-hoc-Netze und andererseits diverse Vorschläge zur Steigerung der Leistungsfähigkeit anderer Protokolle. Zur Umsetzung eines Gruppenkommunikationsdienstes erfolgt in Kapitel 6 die Darstellung einer Architektur, anhand welcher der Gruppenkommunikationsdienst durch Rückgriff auf unterschiedliche Overlay-Topologien und Zuverlässigkeitsvarianten flexibel komponiert und so auf bestimmte Anwendungen abgestimmt werden kann. Eine ausführliche Evaluation des Gruppenkommunikationsdienstes auf Basis unterschiedlicher Overlay-Topologien und Zuverlässigkeitsvarianten sowie im Kontext verschiedener Anwendungen erfolgt in Kapitel 7. Abschließend gibt Kapitel 8 eine Zusammenfassung der in dieser Arbeit gewonnenen Ergebnisse.

2. Grundlagen

In diesem Kapitel werden Grundlagen aufgeführt, welche zum Verständnis der vorliegenden Arbeit erforderlich sind. Begonnen wird mit einer Charakterisierung drahtloser Ad-hoc-Netze, welche in dieser Arbeit die zugrunde liegende Kommunikationsstruktur bilden. Dabei werden diverse Mechanismen betrachtet, welche die Kommunikation zwischen drahtlosen Endgeräten ermöglichen: Zu diesen zählen beispielsweise Medienzugriffsverfahren und Routing-Protokolle. Anschließend wird der Grundgedanke Peer-to-Peer-basierter Gruppenkommunikationsprotokolle vermittelt, welche einen neuartigen und in dieser Arbeit zum Einsatz kommenden Ansatz zur Umsetzung einer Gruppenkommunikation bilden.

2.1 Drahtlose Ad-hoc-Netze

Drahtlose Ad-hoc-Netze sind eine neuartige Form drahtloser Kommunikationsnetze. Sie entstehen durch den *spontanen Zusammenschluss* tragbarer Endgeräte, welche über eine drahtlose Kommunikationsschnittstelle verfügen. Aus heutiger Sicht können unter diesen Endgeräten Notebooks, Mobiltelefone und Organizer oder PDAs verstanden werden. Um die Kommunikation zwischen zwei Endgeräten zu ermöglichen, wird einerseits die Kompatibilität der Kommunikationsschnittstellen vorausgesetzt und andererseits eine hinreichend große *Übertragungreichweite* der Endgeräte erfordert. Befinden sich zwei Endgeräte in *gegenseitiger Übertragungreichweite*, so existiert zwischen ihnen eine *drahtlose Verbindung (Link)*, über welche Dateneinheiten ausgetauscht werden können. Unter anderem aus Gründen des Energieverbrauchs ist die Übertragungreichweite in der Regel auf einige 10 bis wenige 100 Meter beschränkt, so dass die Kommunikation

2. Grundlagen

zwischen weiter entfernten Endgeräten lediglich indirekt erfolgen kann: In derartigen Fällen müssen räumlich dazwischen liegende Endgeräte, welche sich wiederum in Übertragungsbereich befinden, eine so genannte *Weiterleitungsfunktion* übernehmen, um den Abstand zu überbrücken.

Dieser Sachverhalt ist in Abbildung 2.1 veranschaulicht, in welcher das Endgerät A eine *Kommunikationsbeziehung* mit den Endgeräten B und C eingeht. Diese Endgeräte werden somit zu *Kommunikationspartnern*, welche jeweils durch einen entsprechenden *Kommunikationsweg* verbunden sind. In Hinblick auf die Darstellung ist zu beachten, dass ein Kommunikationsweg eine Kommunikationsbeziehungen visualisiert und daher an den Kommunikationspartnern *endet* aber eventuell durch weiterleitende Endgeräte *hindurchführt*. In dem genannten Beispiel ist eine Datenübertragung zwischen A und B direkt möglich, da sich beide Kommunikationspartner in gegenseitiger Übertragungsbereich befinden und somit ein Link zwischen ihnen besteht. Zur Übertragung von Daten zwischen A und C hingegen ist der Einbezug zweier zusätzlicher Endgeräte, X und Y, erforderlich, welche eine Weiterleitungsfunktion übernehmen. Man spricht dabei auch von den einzelnen *Übertragungsabschnitten* (auch *Hops* genannt), über welche ein Kommunikationsweg führt. Aus Sicht von A besteht der Kommunikationsweg zum Erreichen von B demnach aus einem Übertragungsabschnitt, weshalb er auch als *Single-Hop-Kommunikationsweg* bezeichnet wird. Der Kommunikationsweg zum Erreichen von C besteht hingegen aus drei Übertragungsabschnitten und ist daher ein *Multi-Hop-Kommunikationsweg*. Entgegen der in Abbildung 2.1 gezeigten Beispiele sind Übertragungsabschnitte nicht notwendigerweise *bidirektional*, so dass die Datenübertragung gegebenenfalls nur in eine Richtung erfolgen kann. Während dies maßgeblich von den eventuell unterschiedlichen Übertragungsbereichen r der Endgeräte abhängt, soll dieser Aspekt hier lediglich angedeutet und erst später genauer betrachtet werden.

Um eine Datenübertragung zwischen zwei sich in gegenseitiger Übertragungsbereich befindenden Endgeräten zu ermöglichen, ist ein spezielles *Medienzugriffsverfahren* erforderlich: In der Tat wird das Funkmedium, welches in drahtlosen Ad-hoc-Netzen zur Kommunikation zwischen Endgeräten zum Einsatz kommt, von allen Endgeräten *gleichmaßen* genutzt, so dass man auch von einem *geteilten Medium* spricht. Der gleichzeitige Medienzugriff räumlich naher Endgeräte muss dabei jedoch vermieden werden, da dies sonst zur Interferenz der Funksignale und somit zum Verlust der übertragenen Dateneinheiten führt. Greifen in Abbildung 2.1 etwa A und Y gleichzeitig auf das Medium zu, so nimmt X lediglich die Interferenz beider Signale wahr, ist aber nicht in der Lage, die einzelnen Dateneinheiten zu rekonstruieren. Eine der Besonderheiten drahtloser Ad-hoc-Netze ist die Tatsache, dass das Medienzugriffsverfahren, welches die Koordination der Kommunikation umsetzt, vollkommen dezentral von den Endgeräten selbst realisiert werden muss: Da drahtlose Ad-hoc-Netze spontan gebildet werden sollen, kann

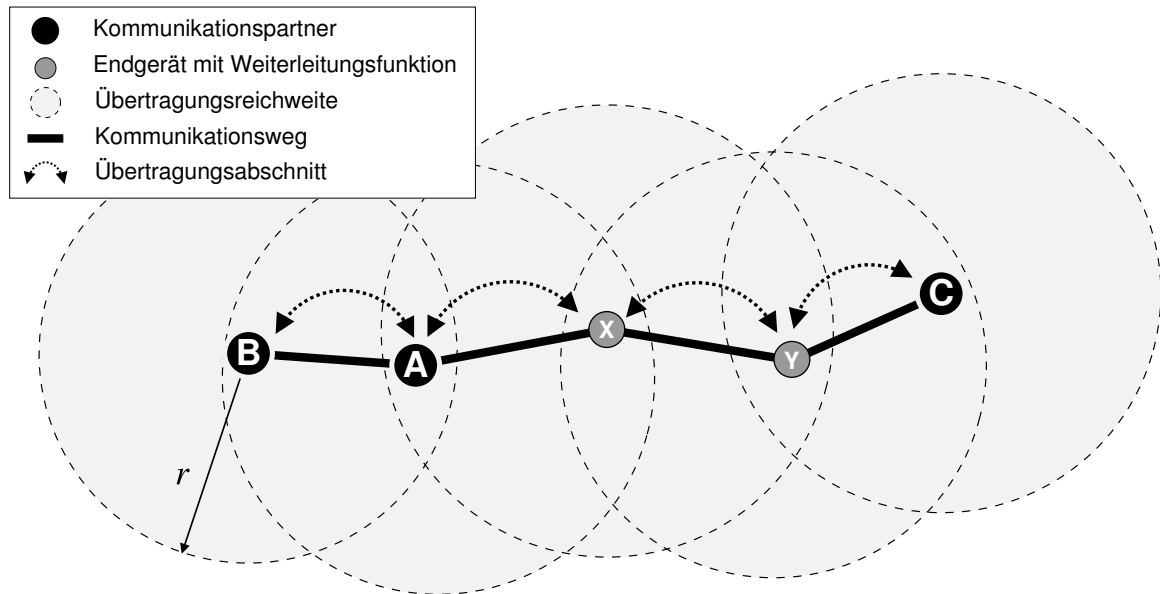


Abbildung 2.1 In drahtlosen Ad-hoc-Netzen werden Endgeräte zur Weiterleitung von Daten zwischen Kommunikationspartnern herangezogen.

eine umliegende, fest installierte Infrastruktur, welche die Aufgabe der Kommunikationskoordination übernehmen könnte, nicht vorausgesetzt werden. Darüber hinaus wird in drahtlosen Ad-hoc-Netzen eine *Gleichberechtigung* aller Endgeräte angestrebt, so dass kein Endgerät privilegierte und übergeordnete Funktionen wie etwa eine Medienzuteilung für andere Endgeräte ausüben darf.

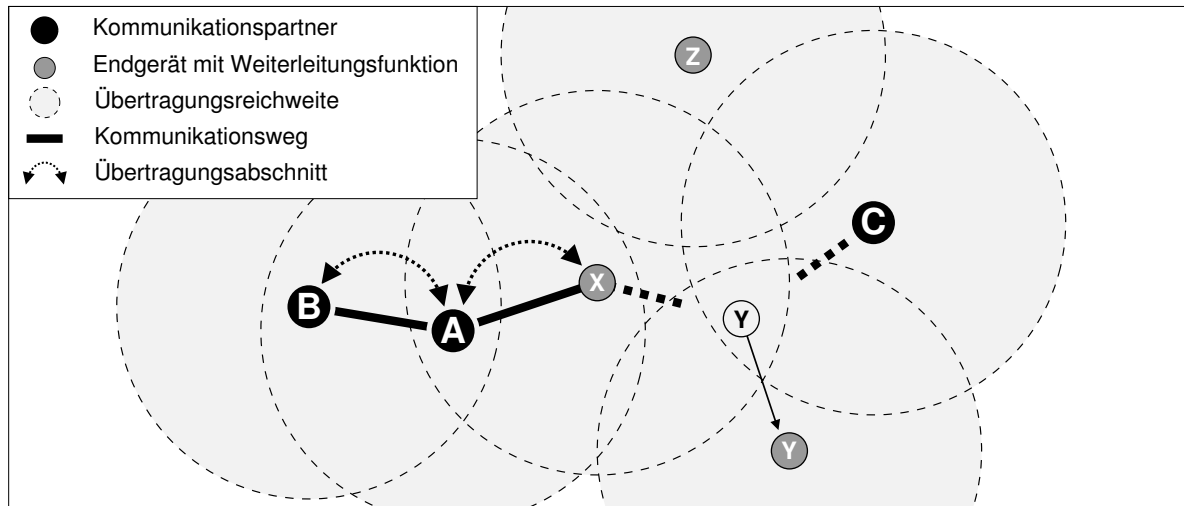
Aufgrund des geteilten Charakters des drahtlosen Mediums werden die von einem Endgerät ausgesendeten Dateneinheiten von *allen* sich in Übertragungsreichweite befindenden Endgeräten empfangen. Man spricht hierbei auch von der *Semi-Broadcast-Fähigkeit* des geteilten Mediums: Ausgehend von Abbildung 2.1 wird eine von A gesendete Dateneinheit sowohl von B als auch von X empfangen. Erst durch Verwendung von *eindeutigen Sender- und Zieladressen* und deren Integration in versendete Dateneinheiten werden Endgeräte unterscheidbar, so dass übertragene Dateneinheiten einem bestimmten Endgerät zugeordnet werden können. Die Forderung nach eindeutigen Adressen erweist sich dabei als essentiell, da andernfalls Endgeräte nicht unterscheidbar sind, und die Herkunft einer Dateneinheit nicht genau festzustellen ist. Analog zu dem oben genannten Medienzugriffsverfahren kann zur Konfiguration, d.h. zur Auswahl, einer Adresse keine fest installierte Infrastruktur mit zentral organisiertem Wissen über bereits vergebene Adressen verwendet werden. Auch hier sind daher Verfahren gefordert, um die Adressvergabe dezentral durch die Endgeräte selbst bewerkstelligen zu lassen.

Trotz dem Vorhandensein eindeutiger Adressen kann es im Zuge von Medienzugriffen allerdings auch explizit unerwünscht sein, die zu übertragende Dateneinheit an ein spe-

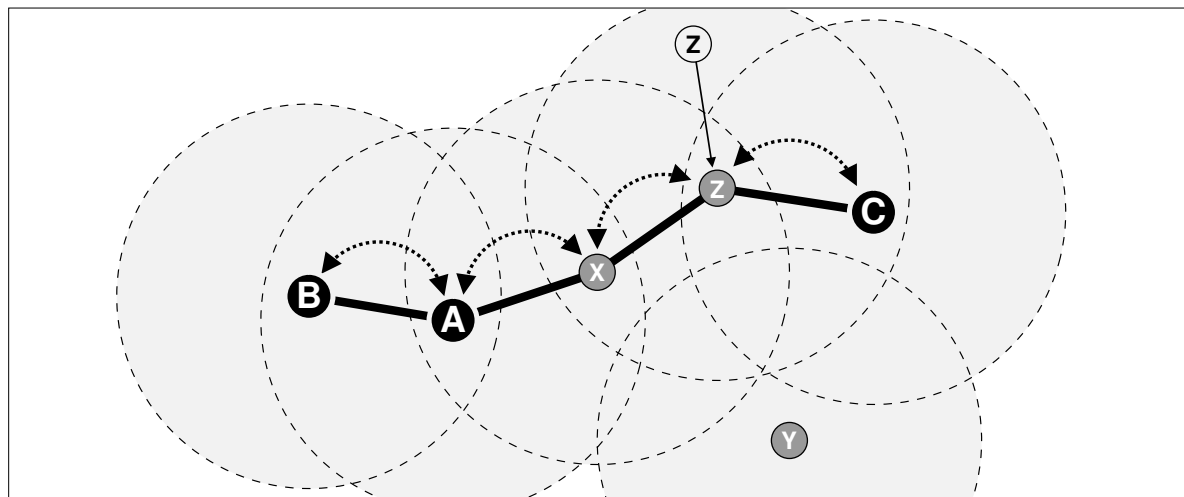
zieltes Endgerät zu richten: So kann es durchaus vorteilhaft sein, die Dateneinheit allen sich in Übertragungsbereich befindenden Endgeräten zukommen zu lassen. Man unterscheidet daher zwischen so genannten *Broadcast*- und *Unicast*-Dateneinheiten. Während zunächst beide Dateneinheitstypen von allen sich in Übertragungsbereich befindenden Endgeräten empfangen werden, verarbeitet ein Endgerät g eine Dateneinheit nur genau dann weiter, wenn es sich um eine Unicast-Dateneinheit handelt, welche an g adressiert ist, oder wenn es sich um eine Broadcast-Dateneinheit handelt. Versendet in Abbildung 2.1 etwa Endgerät A eine Unicast-Dateneinheit an X, so wird die Dateneinheit zwar auch von B empfangen aber hier aufgrund der nicht übereinstimmenden Zieladresse sofort verworfen.

Ein weiterer wichtiger Aspekt mit entscheidendem Einfluss auf die Entwicklung von Kommunikationsprotokollen für drahtlose Ad-hoc-Netze ist die potenzielle *Mobilität* der Endgeräte. Da drahtlose Ad-hoc-Netze spontan von kleinen, tragbaren Endgeräten gebildet werden, können diese auch während ihres Betriebs eine gewisse Mobilität aufweisen: So etwa in Form eines in der Jackentasche mitgeführten Mobiltelefons. Aus der Mobilität und der beschränkten Übertragungsbereich der Endgeräte resultieren Änderungen der in einem drahtlosen Ad-hoc-Netz vorhandenen Links. So wäre es beispielsweise denkbar, dass sich Y in Abbildung 2.1 mit einer gewissen Geschwindigkeit fortbewegt. Wie in Abbildung 2.2.a) dargestellt ist kann Y schließlich den Übertragungsbereich von X und C verlassen und somit von diesen Endgeräten nicht mehr erreicht werden: Der Link zwischen Y und X ist ebenso wie der Link zwischen Y und C *gebrochen*. Da in einem solchen Fall auch der Kommunikationsweg zwischen A und C unterbrochen wird, ist ein Datenaustausch zwischen diesen Endgeräten nicht mehr möglich. In Abbildung 2.2.b) betritt wiederum ein neues Endgerät Z den Übertragungsbereich von X und C, so dass dieses zum Wiederaufbau des Kommunikationswegs zwischen A und C eingesetzt werden kann. In diesem Beispiel wird der dynamische Charakter drahtloser Ad-hoc-Netze deutlich. Insbesondere wird erkennbar, dass Kommunikationswege zwischen zwei Endgeräten fortlaufenden Änderungen unterliegen können. Ein essentielles Problem, welches gelöst werden muss, um die Kommunikation zwischen Endgeräten zu ermöglichen, ist demnach nicht nur die *Findung von Kommunikationswegen* sondern auch deren *Aufrechterhaltung* zur Reaktion auf die mögliche Mobilität der beteiligten Endgeräte: Tatsächlich kann aufgrund des spontanen Charakters drahtloser Ad-hoc-Netze der Aufbau von Kommunikationswegen nicht vorweg festgelegt werden, so dass er dynamisch in Erfahrung gebracht und über die Kommunikation hinweg aufrecht erhalten werden muss.

Aufgrund der obigen Charakterisierung drahtloser Ad-hoc-Netze lassen sich drei unterschiedliche Protokolle mit klar definierten Aufgabenbereichen erkennen, welche erforderlich sind, um in einem drahtlosen Ad-hoc-Netz die Kommunikation zwischen zwei



a) Unterbrechung eines Kommunikationswegs aufgrund der Mobilität von Endgeräten.



b) Wiederaufbau des Kommunikationswegs nach Auffinden eines neuen Endgeräts.

Abbildung 2.2 Aufgrund von Mobilität können Kommunikationswege zwischen Endgeräten fortlaufenden Änderungen unterliegen.

Endgeräten zu ermöglichen. So sind Protokolle für den *Medienzugriff*, für die *Adressvergabe* und für die *Auffindung und die Aufrechterhaltung von Kommunikationswegen zwischen Endgeräten* erforderlich. Während die Adressvergabe durchaus als kritisch für eine reibungslose Kommunikation anzusehen ist, sind die genauen Abläufe und Verfahren, welche dabei zum Einsatz kommen, für diese Arbeit nicht weiter von Bedeutung: Protokollansätze sollen an dieser Stelle daher nicht näher betrachtet sondern lediglich referenziert werden [111, 112]. Die beiden übrigen Aufgabenstellungen, der Medienzugriff und die Kommunikationswegfindung, hingegen haben kontinuierlichen Einfluss auf die in dieser Arbeit entwickelten Protokolle und Mechanismen. In den zwei folgenden

Abschnitten sollen daher die wichtigsten Ansätze für den Medienzugriff und die Kommunikationswegfindung, welche von der Forschungsgemeinde auch weitläufig zur Untersuchung von drahtlosen Ad-hoc-Netzen herangezogen werden, kurz vorgestellt werden. Um Begriffe sowie die Aufgabenbereiche der einzelnen Protokolle im Hinblick auf eine spätere Referenzierung zu vereinheitlichen, sollen die vorgestellten Ansätze jeweils im Kontext des gängigsten Modells zur Strukturierung von Kommunikationssystemen, dem ISO/OSI Schichtenmodell [119], betrachtet werden.

2.1.1 Das Medienzugriffsprotokoll IEEE 802.11

Das Verfahren, aufgrund dessen Endgeräte auf das drahtlose Medium zugreifen, ist von entscheidender Bedeutung für die gesamte, innerhalb eines drahtlosen Ad-hoc-Netzes ablaufende Kommunikation, da es bei der Übertragung jeder Dateneinheit zum Einsatz kommt. Die Hauptaufgabe eines Medienzugriffsprotokolls ist es zu vermeiden, dass sich Endgeräte bei der Übertragung ihrer Dateneinheiten gegenseitig stören. Aufgrund der Eigenschaften des drahtlosen Mediums zeigt sich diese Aufgabe als Herausforderung.

Zur Koordination der Medienzugriffe unterschiedlicher Endgeräte verfügt der Standard IEEE 802.11 über zwei unterschiedliche Betriebsmodi. Bei dem ersten Betriebsmodus, dem so genannten *Infrastrukturmodus*, wird die Koordination über eine *Point Coordination Function (PCF)* zentral umgesetzt. Dieser Betriebsmodus wird deshalb im Wesentlichen im Zusammenhang mit fest installierter Infrastruktur, wie beispielsweise *WLAN Access Points*, eingesetzt. Diese Infrastruktur übt eine den drahtlosen Endgeräten übergeordnete Funktion aus, indem sie ihnen Medienzugriffe explizit zuweist. Bei dem *Ad-hoc-Betriebsmodus* hingegen erfolgt die Koordination der Medienzugriffe verteilt, anhand einer so genannten *Distributed Coordination Function (DCF)*. Da hier die Endgeräte anhand eines *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)* genannten Verfahrens selbst für die Koordination verantwortlich sind, ist dieser Betriebsmodus unabhängig von jeglicher fest installierter Infrastruktur. Die DCF beschreibt die in dieser Arbeit zum Einsatz kommende Medienzugriffskoordination, so dass im Folgenden auf die wesentlichen Kernpunkte des Protokolls eingegangen wird.

2.1.1.1 Senden und Empfangen von Funksignalen

Wird von einem Endgerät ein Medienzugriff zur Übertragung einer Dateneinheit durchgeführt, breitet sich ausgehend von der Antenne der drahtlosen Kommunikationsschnittstelle ein Funksignal aus. Aufgrund der räumlichen Ausbreitung nimmt die Signalstärke mit der Entfernung zum sendenden Endgerät ab. Endgeräte, welche sich in einer gewissen Entfernung zum sendenden Endgerät befinden, können die Dateneinheit empfangen, falls die Signalstärke des Funksignals einen bestimmten Schwellenwert überschreitet.

Zur Untersuchung drahtloser Netze wurden von der Forschungsgemeinde Modelle entwickelt, welche die Ausbreitung und Abschwächung von elektromagnetischen Signalen

in einer bestimmten Höhe über einer unendlichen Ebene approximieren [83]. Dabei wird eine gleichförmige Ausbreitung der Signale ausgehend von der Antenne des Senders angenommen, so dass die Unterschreitung des Schwellenwerts zum Empfang einer Dateneinheit in einer bestimmten Entfernung d_{Rx} ($Rx = Receive$) zum Sender stattfindet. Diese Entfernung, bis zu welcher Endgeräte eine Dateneinheit empfangen, d.h. *rekonstruieren*, können, wird als der *Übertragungsbereich* des sendenden Endgeräts bezeichnet. Übersteigt die Entfernung zwischen dem sendenden und einem empfangenden Endgerät diesen Wert, kann letzteres das Signal noch *erkennen* aber die Dateneinheit nicht mehr rekonstruieren. Dieser *Erkennungsbereich* ist wiederum auf eine Entfernung $d_{CS} \geq d_{Rx}$ ($CS = Carrier Sense$) beschränkt. Ab dieser Entfernung kann ein Endgerät das Signal nicht mehr erkennen: Es wird lediglich als *Rauschen* wahrgenommen.

Die Ausdehnungen des Übertragungs- und des Erkennungsbereichs hängen maßgeblich von der Sendeleistung des sendenden Endgeräts einerseits und von den Empfindlichkeiten der empfangenden Endgeräte gegenüber der Erkennung und der Rekonstruktion von Funksignalen andererseits ab. Die Empfindlichkeiten drahtloser Kommunikationsschnittstellen gegenüber Funksignalen werden bei dem Standard IEEE 802.11 nicht direkt vorgegeben: Vielmehr werden lediglich Minimalwerte genannt, so dass einzelne Hersteller unterschiedliche Umsetzungen bieten [22]. Darüber hinaus kann die Sendeleistung variiert werden, um Übertragungsbereiche zu vergrößern oder zu verkleinern. Um einen Eindruck von der Entwicklung des Übertragungs- und des Erkennungsbereichs in Abhängigkeit der Sendeleistung zu bekommen, sind diese beispielhaft in Abbildung 2.3 aufgetragen. Die Kurvenverläufe wurden durch die genannten physikalische Modelle gewonnen, wobei (durchaus übliche [22]) Signalstärken von -81dBm zur Rekonstruktion und -91dBm zur Erkennung der Funksignale gefordert wurden.

Das im Folgenden für den Standard IEEE 802.11 beschriebene Medienzugriffsverfahren basiert auf der Erkennung (*Sensing*) von Funksignalen auf dem drahtlosen Medium und dem Vermeiden gleichzeitiger Medienzugriffe: Dabei nimmt ein Endgerät das Medium genau dann als belegt wahr, wenn die auf dem Medium vorhandene, von der Antenne erfasste Signalstärke die zur Erkennung eines Funksignals erforderliche Signalstärke übersteigt. Wichtig hierbei ist, dass sich die von der Antenne erfasste Signalstärke in der Regel aus einer *Überlagerung* zahlreicher Funksignale ergibt. Folglich ist es möglich, dass ein Endgerät ein belegtes Medium wahrnimmt, obwohl keines der einzelnen, bei diesem Endgerät eintreffenden Funksignale an sich erkannt wird.

2.1.1.2 Die Koordination der Medienzugriffe

Vor der Durchführung eines Medienzugriffs muss ein sendewilliges Endgerät für eine festgelegte Dauer *DIFS* (*Distributed Inter Frame Space*) die Freiheit des Mediums durch aktives Abhören sichergestellt haben. Ist dies erfolgreich geschehen, darf auf das Medium zugegriffen und eine Dateneinheit übertragen werden. Wurde das Medium hingegen

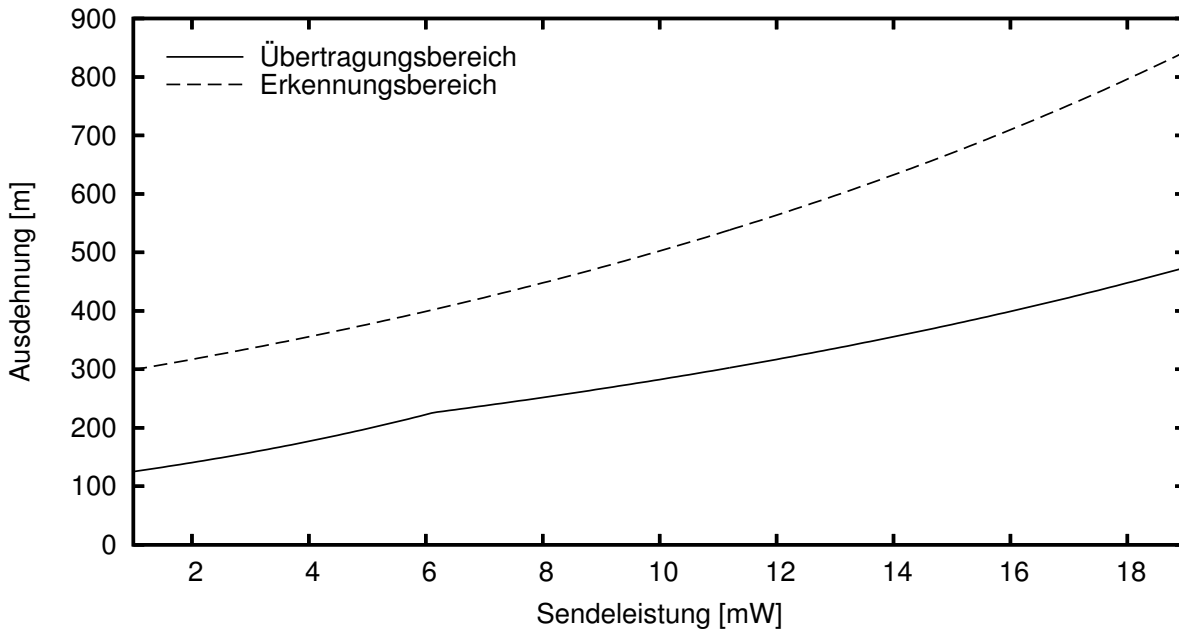


Abbildung 2.3 Approximation der Ausdehnung des Übertragungsbereichs und des Erkennungsbereichs von WLAN-Funksignalen in Abhängigkeit der Sendeleistung gemäß [83] bei einer Antennenhöhe von 1,5m.

zwischenzeitlich als belegt erkannt, ist zunächst das Freiwerden des Mediums abzuwarten und seine Freiheit erneut für die Dauer DIFS zu prüfen.

Für den Fall, dass während der Medienbelegung Sendewünsche auf umliegenden Endgeräten eintreffen, birgt dieses einfache Verfahren ein erhöhtes Kollisionsrisiko. Tatsächlich würden alle Endgeräte nach Freiwerden des Mediums und nach der zusätzlichen Dauer DIFS auf das Medium zugreifen und somit Kollisionen verursachen. Um dieses Risiko zu reduzieren, werden zusätzliche Wartezeiten, so genannte *Back-off*-Zeiten, verwendet. Eine *Back-off*-Zeit t wird von einem Endgerät gewählt, falls ein eigener Sendewunsch aufgrund einer erkannten Medienbelegung verzögert wird. Die Wahl erfolgt dabei zufällig aus einem quantisierten Intervall CW , das so genannte *Contention Window* (dt. *Wettbewerbsfenster*), welches später näher erläutert wird. Kann das Endgerät *zusätzlich* zur Dauer DIFS die Freiheit des Mediums auch für die Dauer t feststellen, darf der Medienzugriff erfolgen. Wird hingegen ein fremder Medienzugriff festgestellt, wiederholt sich das Verfahren, wobei für die *Back-off*-Zeit jedoch keine neue Zufallszahl gezogen sondern die verbleibende *Back-off*-Zeit verwendet wird.

2.1.1.3 Die Sicherung von Unicast-Dateneinheiten

In Abhängigkeit der Übertragungsart einer Dateneinheit schließen sich an den Medienzugriff und die damit erfolgende Übertragung gegebenenfalls zusätzliche *Sicherungsmechanismen* an. Handelt es sich bei der übertragenen Dateneinheit um eine *Broadcast*-

Übertragung, erfolgen nach der initialen Übertragung der Dateneinheit *keine* weiteren Sicherungsmechanismen. Überträgt das Endgerät die Dateneinheit hingegen als *Unicast*, d.h. unter genauer Adressierung eines Empfängergeräts, so wird auf eine Bestätigung (*Acknowledgement, ACK*) der Dateneinheit durch das Empfängergerät gewartet. Dieses bestätigt eine Dateneinheit nach ihrem vollständigen, störungsfreien Empfang² und dem Verstreichen einer zusätzlichen Dauer *SIFS* (*Short Inter Frame Space*). Durch die Wahl von $SIFS < DIFS$ wird dabei eine Priorisierung von Bestätigungen gegenüber Nutzdateneinheiten ermöglicht³. Wurde auch nach Ablauf einer festgelegten Zeitspanne keine Bestätigung von Seiten des Empfängergeräts erhalten, nimmt das Sendergerät einen Verlust der Dateneinheit aufgrund einer Kollision an. Für einen derartigen Fall sieht IEEE 802.11 die Durchführung von *Übertragungswiederholungen* vor, im Laufe welcher die soeben unbestätigt gebliebene Dateneinheit erneut übertragen wird. Weil das Ausbleiben der Bestätigung auf eine Kollision, d.h. auf einen fremden Medienzugriff, zurückgeführt wird, erfolgt der Medienzugriff für die Übertragungswiederholung unter Verwendung einer Back-off-Zeit. Um jedoch das Kollisionsrisiko für die bevorstehende Übertragungswiederholung zu reduzieren, wird das Intervall, aus welchem die Back-off-Zeit gewählt wird, vergrößert. Die Größe des Intervalls CW_i lässt sich dabei direkt aus der Anzahl i der für eine Dateneinheit bereits durchgeführten Übertragungswiederholungen ableiten. Die Wahl der Back-off-Zeit t_i für die i -te Übertragungswiederholung lässt sich letztendlich durch den Ausdruck

$$\begin{aligned} t_i &= t_{slot} \cdot \text{rand}[0, \dots, CW_i] \\ &= 20\mu s \cdot \text{rand}[0, \dots, \min\{2^{5+i}, 1024\}] \end{aligned} \quad (2.1)$$

beschreiben. Während der Ausdruck $\text{rand}[a, \dots, b]$ eine ganzzahlige Zufallszahl $\in [a, b)$ zurückgibt, bezeichnet t_{slot} eine Größe zur Umrechnung des quantisierten Intervalls auf entsprechende zeitliche Verzögerungen. Die Formel gilt insbesondere auch zur Berechnung der Back-off-Zeit für die initiale Übertragung ($i = 0$) der Dateneinheit.

Jede als Unicast versendete Dateneinheit wird bis zu 7 mal übertragen (1 initiale Übertragung + 6 Übertragungswiederholungen), wobei das dabei exponentiell vergrößerte Back-off-Intervall auf einen Maximalwert von 1024 beschränkt wird. Wird auch nach der letzten Übertragungswiederholung keine Bestätigung von Seiten des Empfängergeräts erhalten, nimmt das Sendergerät den Link zum Empfängergerät als gebrochen an. Da dies eine *Unerreichbarkeit* des Empfängergeräts bedeutet, wird die zu übertragende

²Für diese Arbeit ist nur der Bestätigungsprozess an sich von Bedeutung, nicht aber die Möglichkeiten zur Detektion von zerstörten Dateneinheiten. Diese werden daher nicht näher betrachtet.

³An dieser Stelle wird auch deutlich weshalb ein Endgerät bei einem eintreffenden Sendewunsch auch bei Freiheit des Mediums dieses noch für die Dauer DIFS zu prüfen hat: Auf diese Weise werden Medienzugriffe zwischen dem Ende einer Unicast-Übertragung und ihrer Bestätigung durch das Empfängergerät verhindert.

Dateneinheit verworfen und die Unerreichbarkeit des Empfängergeräts an das Routing-Protokoll (vgl. Abschnitt 2.1.2) gemeldet.

Zur Verdeutlichung des soeben beschriebenen Medienzugriffsverfahrens soll das in Abbildung 2.4 dargestellte Beispiel dienen, in welchem sich die Endgeräte A, B, C und D in gegenseitiger Übertragungsreichweite befinden. Das Beispiel beginnt mit einem zum Zeitpunkt t_0 bei A eintreffenden Sendewunsch. A erkennt das Medium sowohl zu diesem Zeitpunkt als auch für die nachfolgende Dauer DIFS als frei und greift daher zum Zeitpunkt $t_0 + DIFS$ auf das Medium zur Übertragung einer Broadcast-Dateneinheit zu. Während die Übertragung von A zum Zeitpunkt t_3 endet, sind bereits zu den Zeitpunkten t_1 und t_2 Sendewünsche bei B und C eingetroffen. Da diese Endgeräte jedoch die von A vorgenommene Medienbelegung erkennen, lauschen sie zunächst aktiv bis zum Zeitpunkt t_3 auf das Freiwerden des Mediums. Daraufhin prüfen B und C die Freiheit des Mediums für die Dauer DIFS. Da die Medienzugriffe beider Endgeräte aufgrund eines fremden Medienzugriffs verhindert wurden, wählen nun beide Endgeräte eine Back-off-Zeit gemäß Formel 2.1. Wie der Abbildung entnommen werden kann wählt B aus dem vorgegebenen Intervall einen Zufallswert von 4 und C einen Wert von 6. Aus diesem Grund läuft die Back-off-Zeit für B zuerst ab, so dass B zum Zeitpunkt t_4 das Medium zur Übertragung einer Broadcast-Dateneinheit belegt. Diese Medienbelegung wird wiederum von den übrigen Endgeräten erkannt, so dass C seinen verbleibenden Back-off-Wert bestimmt. Die Übertragung von B endet zum Zeitpunkt t_5 , so dass C ab diesem Zeitpunkt das Medium für die Dauer DIFS prüft. Da auch während des sich anschließenden restlichen Back-off keine Medienbelegung erkannt wird, kann C mit der Übertragung einer Unicast-Dateneinheit beginnen. Während diese Übertragung zum Zeitpunkt t_8 endet, liegen bereits zu den Zeitpunkten t_6 und t_7 erneut Sendewünsche bei A und B vor. Ab Zeitpunkt t_8 prüfen diese daher die Freiheit des Mediums für die Dauer DIFS. Doch bereits zum Zeitpunkt $t_8 + SIFS < t_8 + DIFS$ erfolgt eine Medienbelegung durch D zur Bestätigung der soeben von C erhaltenen Unicast-Dateneinheit. Während die Übertragung der Bestätigung zum Zeitpunkt t_9 endet, beginnen ab hier A und B erneut die Freiheit des Mediums für die Dauer DIFS zu prüfen. Da das Medium zu den Zeitpunkten t_6 und t_7 durch C belegt war, bestimmen beide wiederum eine Back-off-Zeit. In diesem Fall wählen beide Endgeräte aus dem Intervall einen Wert von 3, so dass die Back-off-Zeiten beider Endgeräte zum gleichen Zeitpunkt t_{10} ablaufen. Aus diesem Grund greifen hier beide Endgeräte auf das Medium zu, A zur Übertragung einer Broadcast- und B zur Übertragung einer Unicast-Dateneinheit. Es entsteht eine *Kollision*. Zum Zeitpunkt t_{11} stellt B das Ausbleiben der erwarteten Bestätigung fest. Ab dem Zeitpunkt t_{12} sowie der Dauer DIFS und einem neu gewählten Back-off beginnt B daher mit der Übertragungswiederholung seiner Unicast-Dateneinheit. Da diese nun zum Zeitpunkt t_{13} erfolgreich von C empfangen wurde, erfolgt hier eine Bestätigung der Dateneinheit.

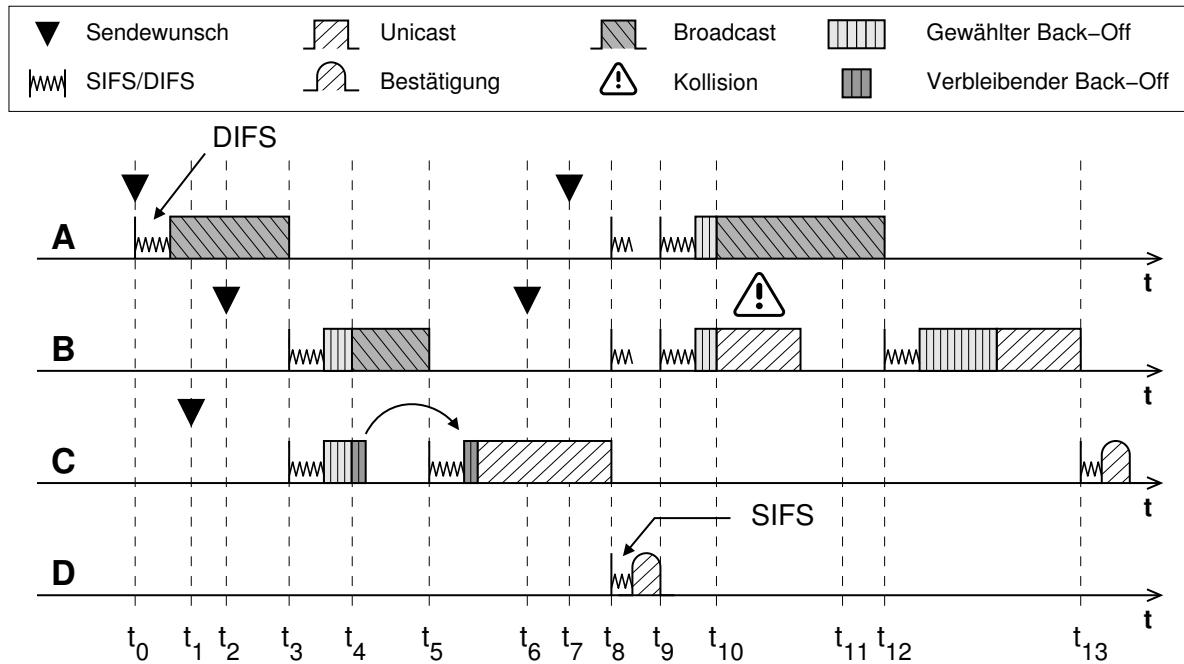
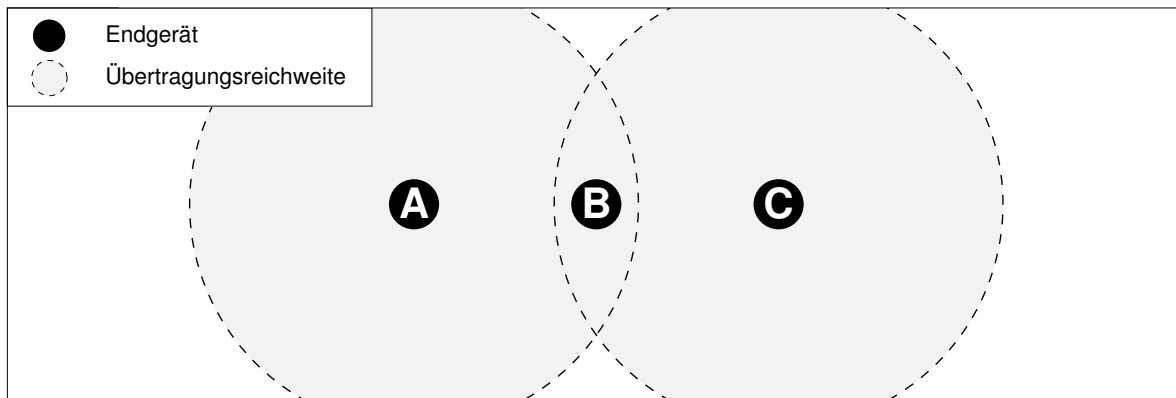


Abbildung 2.4 Ablauf des CSMA/CA-Mediumzugriffs im Rahmen der Koordination durch die DCF bei IEEE 802.11.

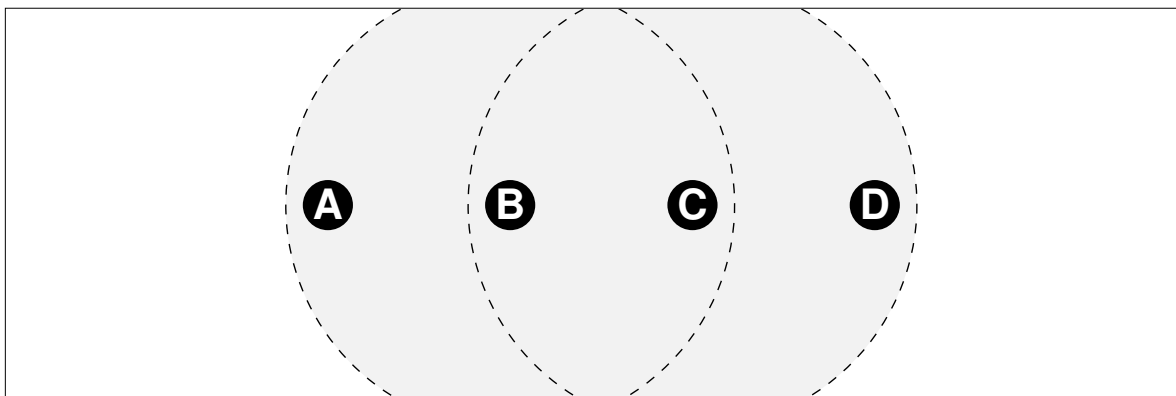
Aufgrund der in diesem Abschnitt beschriebenen Sicherung von Unicast-Dateneinheiten kann Stellung zu der eingangs erwähnten Problematik von uni- oder bidirektionalen Übertragungsabschnitten genommen werden. Insbesondere für den Austausch von Unicast-Dateneinheiten wird deutlich, dass deren erfolgreiche Übertragung das Vorhandensein eines bidirektionalen Übertragungsabschnitts voraussetzt. Ist ein Übertragungsabschnitt unidirektional, wird das sendende Endgerät durch die ausbleibende Bestätigung einer Dateneinheit das Empfängergerät als nicht erreichbar erklären, sobald die maximale Anzahl durchzuführender Übertragungswiederholungen erreicht ist.

2.1.1.4 Die RTS/CTS-Erweiterung

Der bislang definierte Mediumzugriff weist zwei gravierende Nachteile auf, welche als Probleme der *versteckten* und *ausgelieferten* Endgeräte bezeichnet werden. Beide Probleme sind in Abbildung 2.5 dargestellt, in welcher eine Kette von Endgeräten aufgetragen ist: In dieser Kette kann ein Endgerät lediglich die Mediumzugriffe der jeweils benachbarten Endgeräte feststellen. In Abbildung 2.5.a) ist die Problematik versteckter Endgeräte aufgeführt: Hier können die Endgeräte A und C ihre Mediumzugriffe gegenseitig nicht erkennen. Bei der jeweiligen Übertragung einer Dateneinheit zu B kommt es demnach unweigerlich zur Kollision: A und C sind voreinander versteckt. In Abbildung 2.5.b) hingegen ist die Problematik ausgelieferter Endgeräte skizziert: Sendet B eine Dateneinheit an A, so verhindert der damit verbundene Mediumzugriff die Übertragung einer Daten-



a) Kollision zweier Dateneinheiten aufgrund versteckter Endgeräte



b) Verzögerung des Medienzugriffs aufgrund ausgelieferter Endgeräte

Abbildung 2.5 Problematik versteckter und ausgelieferter Endgeräte während des Medienzugriffs von IEEE 802.11.

einheit von C nach D. Diese könnte jedoch ungestört stattfinden, da der Medienzugriff von B den Empfang bei D nicht weiter beeinträchtigt: C ist B demnach ausgeliefert.

Bemerkung: Von dieser Stelle an wird der eigentliche Inhalt und die mit Dateneinheiten verbundene Semantik relevant, da sie den Kommunikationsablauf steuern. Aus diesem Grund wird statt dem Begriff der Dateneinheit nun verstärkt der Begriff Paket verwendet. Um den Einfluss von Paketen auf den Kommunikationsablauf zu verdeutlichen, sind Pakete üblicherweise mit einer Typ-Angabe verbunden.

Die RTS/CTS-Erweiterung von IEEE 802.11 versucht auf Basis zusätzlicher Dateneinheiten, so genannter *Request-To-Send-* (RTS) und *Clear-To-Send-* Pakete (CTS), diese Problematik zumindest bei der Versendung von Unicast-Dateneinheiten einzudämmen. Vor der Übertragung der eigentlichen Dateneinheit sendet ein sendewilliges Endgerät ein RTS-Paket, in welchem das Empfängergerät und die Dauer des bevorstehenden Datenaustauschs kodiert ist. Das RTS-Paket wird durch das Empfängergerät anhand

eines CTS-Pakets beantwortet. In diesem ist ebenfalls das sendewillige Endgerät und die Übertragungsdauer kodiert. Erst bei Erhalt des CTS-Pakets beginnt die Übertragung der eigentlichen Dateneinheit sowie deren Bestätigung. Anhand der RTS/CTS-Erweiterung wird in Abbildung 2.5.a) demnach ein von A ausgesendetes RTS-Paket von B beantwortet. C wird durch Mithören von der Übertragung in Kenntnis gesetzt und verzögert die eigene Übertragung. In Abbildung 2.5.b) wird das von B versendete RTS-Paket von A beantwortet. Durch Mithören des RTS-Pakets erlangt C wiederum Wissen über die bevorstehende Übertragung. Da C das von A versendete CTS-Paket jedoch nicht empfängt, braucht C die eigene Übertragung zu D nicht zu verzögern, da die Entfernung zu A groß genug ist, um die Übertragung von B nicht zu stören.

Von unterschiedlichen Forschungsgruppen wurden zahlreiche Arbeiten durchgeführt, um den tatsächlichen Nutzen der RTS/CTS-Erweiterung zu untersuchen [33, 57, 106, 115]. Dabei hat sich ergeben, dass sich die Erweiterung nicht immer gewinnbringend, oftmals sogar kontraproduktiv zeigt. Ob anhand der RTS/CTS-Erweiterung letztendlich ein Nutzen oder Verlust erzielt wird, hängt von zahlreichen Größen ab: Diese umfassen beispielsweise die Bandbreite der drahtlosen Kommunikationsschnittstelle, die Größe der Dateneinheiten, die Anzahl der um den Medienzugriff konkurrierenden Endgeräte und die Sendeleistung bzw. den daraus resultierenden Übertragungs- und Erkennungsbereich. Da eine Entscheidung über den Einsatz der RTS/CTS-Erweiterung somit nicht allgemein möglich sondern stark abhängig von den angestrebten Anwendungsszenarien ist, soll sie auf das Evaluationskapitel vertagt werden.

2.1.2 Wegewahl in drahtlosen Ad-hoc-Netzen

Um in drahtlosen Ad-hoc-Netzen die Kommunikation zwischen zwei entfernten Endgeräten zu ermöglichen, ist es erforderlich, zwischen ihnen eine so genannte *Route* aufzubauen. Das Aufbauen einer Route entspricht dem Finden eines Kommunikationswegs zwischen den Kommunikationspartnern sowie der dazugehörigen Übertragungsabschnitte. Zur Verdeutlichung des Unterschieds zwischen einem *Kommunikationsweg* und einer *Route* sei an dieser Stelle definiert, dass in dieser Arbeit unter dem Begriff des Kommunikationswegs lediglich eine *Konkatenation von Übertragungsabschnitten* verstanden wird. Der Begriff der Route hingegen bezeichnet die Information, welche einzelnen Endgeräten zur *Beschreibung eines Kommunikationswegs* vorliegt und diesen damit für die Kommunikation erst nutzbar macht.

Die Aufgabe der Findung einer Route zwischen Endgeräten, deren Aufrechterhaltung im Laufe der Kommunikation sowie die Weiterleitung von Dateneinheiten entlang des zugehörigen Kommunikationswegs wird von so genannten *Routing-Protokollen* erfüllt, welche auf der *Vermittlungsschicht* (auch *Routing-Schicht* genannt) des ISO/OSI Schichtenmodells angesiedelt sind. In Abhängigkeit der verwendeten Protokollmechanismen und der

zur Routen-Bestimmung ausgetauschten Dateneinheiten verursachen Routing-Protokolle unterschiedlichen Aufwand. Das einheitliche Ziel ist dabei in der Regel jedoch das Gleiche: So wird angestrebt, dass die Anzahl der in einer Route enthaltenen Übertragungsabschnitte möglichst klein gehalten wird. Da Routing-Protokolle ein elementarer, zur Kommunikation erforderlicher Bestandteil sind, hat sich die Forschungsgemeinde ausgiebig mit der Thematik befasst und zahlreiche Lösungsansätze hervorgebracht. Nur die wenigsten davon wurden standardisiert, so dass hier lediglich die prominentesten Vertreter genannt werden sollen.

Routing-Protokolle lassen sich in zwei Klassen einteilen: *Proaktive* und *reaktive* Routing-Protokolle. Zwar existieren auch so genannte *hybride* Ansätze, welche proaktive und reaktive Techniken kombinieren, doch haben diese zur Untersuchung drahtloser Ad-hoc-Netze bislang keine herausragende Bedeutung erlangt. Proaktive Protokolle streben einen Zustand an, in welchem jedem Endgerät zu jedem Zeitpunkt eine Route zu jedem anderen Endgerät bekannt ist. Dieser Zustand wird unabhängig von der Tatsache angestrebt, ob ein Endgerät tatsächlich mit einem anderen Endgerät kommunizieren will. Dahingegen versuchen reaktive Routing-Protokolle eine Route zwischen zwei Endgeräten erst bei Bedarf, d.h. im Falle einer tatsächlichen Kommunikation, zu finden.

Beide Protokollklassen haben unterschiedliche Vor- und Nachteile. Während reaktive Routing-Protokolle erst im Falle eines Kommunikationswunschs Aufwand zur Routen-Bestimmung betreiben, verzögert sich aufgrund des zunächst erforderlichen Routen-Aufbaus die eigentliche Kommunikation. Proaktive Routing-Protokolle hingegen betreiben fortlaufend eine Routen-Bestimmung und verursachen dabei einen konstanten Aufwand, welcher bei einem geringen Kommunikationsaufkommen oftmals höher als bei einem reaktiven Routing-Protokoll ist. Jedoch entfällt bei proaktiven Protokollen die Verzögerung der Kommunikation, da dem Endgerät bereits eine Route bekannt ist.

Um einen Einblick in die tatsächliche Funktionsweise der Protokolle zu erhalten, wird in den folgenden zwei Abschnitten jeweils der prominenteste Vertreter der reaktiven und proaktiven Routing-Protokolle kurz beschrieben.

2.1.2.1 Ad-hoc On Demand Distance Vector Routing

Das *Ad-hoc On Demand Distance Vector Routing* (AODV [76]) ist der wichtigste und bereits standardisierte Vertreter der reaktiven Routing-Protokolle. Es stößt genau dann einen Routen-Aufbau an, wenn ein Sendergerät eine Dateneinheit an ein bestimmtes Empfängergerät senden möchte, aber bislang keine Route zu diesem vorliegt.

Der Routen-Aufbau selbst erfolgt indem das drahtlose Ad-hoc-Netz mit einem Paket des Typs *RouteRequest* (*RREQ*) geflutet wird, welches die Adresse des zu suchenden Empfängergeräts und des ursprünglichen Sendergeräts enthält. Das Fluten wird erreicht indem das RREQ-Paket als Broadcast versendet und daher von allen Endgeräten in

Übertragungsreichweite empfangen wird. Diese leiten das Paket ebenfalls als Broadcast weiter, so dass sich das Paket allmählich durch das drahtlose Ad-hoc-Netz bis hin zum gesuchten Empfängergerät propagiert. Durch Verwendung von Sequenznummern innerhalb der Pakete sowie deren Zwischenspeicherung in den Endgeräten wird erreicht, dass jedes Endgerät das gleiche RREQ-Paket nur ein einziges mal weiterleitet. Während der Weiterleitung speichern Endgeräte jeweils von welchem Endgerät das RREQ-Paket erhalten wurde. Auf diese Weise entsteht zunächst der so genannte *Reverse Path*, welcher vom Empfängergerät zum ursprünglichen Sendergerät führt. Entlang des Reverse Path wird anschließend ein vom Empfängergerät initiiertes *RouteReply*-Paket (*RREP*) geleitet, welches nun jedoch als Unicast übertragen wird. Auch hier speichern die an der Zurückleitung des RREP-Pakets beteiligten Endgeräte dasjenige Endgerät, von welchem das RREP-Paket empfangen wurde: Hierdurch entsteht wiederum der *Forward Path*, welcher letztendlich einer Route vom Sendergerät zum Empfängergerät entspricht und zur Kommunikation genutzt werden kann.

Zur Veranschaulichung des Protokolls soll das in Abbildung 2.6 dargestellte Beispiel dienen, welches ein aus den Endgeräten A, B, C, D und E bestehendes drahtloses Ad-hoc-Netz zeigt. Das Beispiel demonstriert einen Routen-Aufbau von A nach E. Hierzu beginnt A in Schritt I mit dem Fluten eines RREQ-Pakets. Dieses wird in Schritt II von B und C weitergeleitet, wobei beide Endgeräte das Endgerät A als Ursprung des RREQ-Pakets speichern. Durch Verzögerung der RREQ-Weiterleitung um eine kurze, zufällig gewählte Dauer wird das Risiko gleichzeitiger Medienzugriffe reduziert. In Schritt III empfängt D das RREQ-Paket von B und leitet es entsprechend weiter. In Schritt IV empfängt schließlich E, das gesuchte Empfängergerät, das RREQ-Paket von D. Zu diesem Zeitpunkt existiert der Reverse Path, welcher einer Route von E nach A entspricht. Der Reverse Path wird in den Schritten IV - VI genutzt, um das von E initiierte RREP-Paket zurück zu A zu leiten. Hierzu werden jetzt nur noch die Endgeräte in die Weiterleitung einbezogen, welche Bestandteil des Reverse Path sind: So erfolgt die Weiterleitung des RREP-Pakets anhand von Unicast-Übertragungen, welche von E über D und B zu A gesendet werden. Während der Weiterleitung speichern die Endgeräte jeweils von welchem Endgerät das RREP-Paket empfangen wurde. In Schritt VII ist schließlich der Forward Path aufgebaut, welcher einer Route von A nach E entspricht.

Hinsichtlich der Problematik unidirektionaler Übertragungsabschnitte ist hier zu bemerken, dass AODV nur bidirektionale Routen aufbaut. Dies ergibt sich aufgrund der Tatsache, dass insbesondere zur Rückleitung eines RREP-Pakets Unicast-Dateneinheiten eingesetzt werden, welche gemäß Abschnitt 2.1.1.3 nur über bidirektionale Übertragungsabschnitte übertragen werden können. Dies hat insbesondere die Konsequenz, dass in obigem Beispiel nach dem Aufbau der Route von A nach E ebenfalls eine Route von E nach A existiert und diese nicht getrennt aufgebaut werden muss.

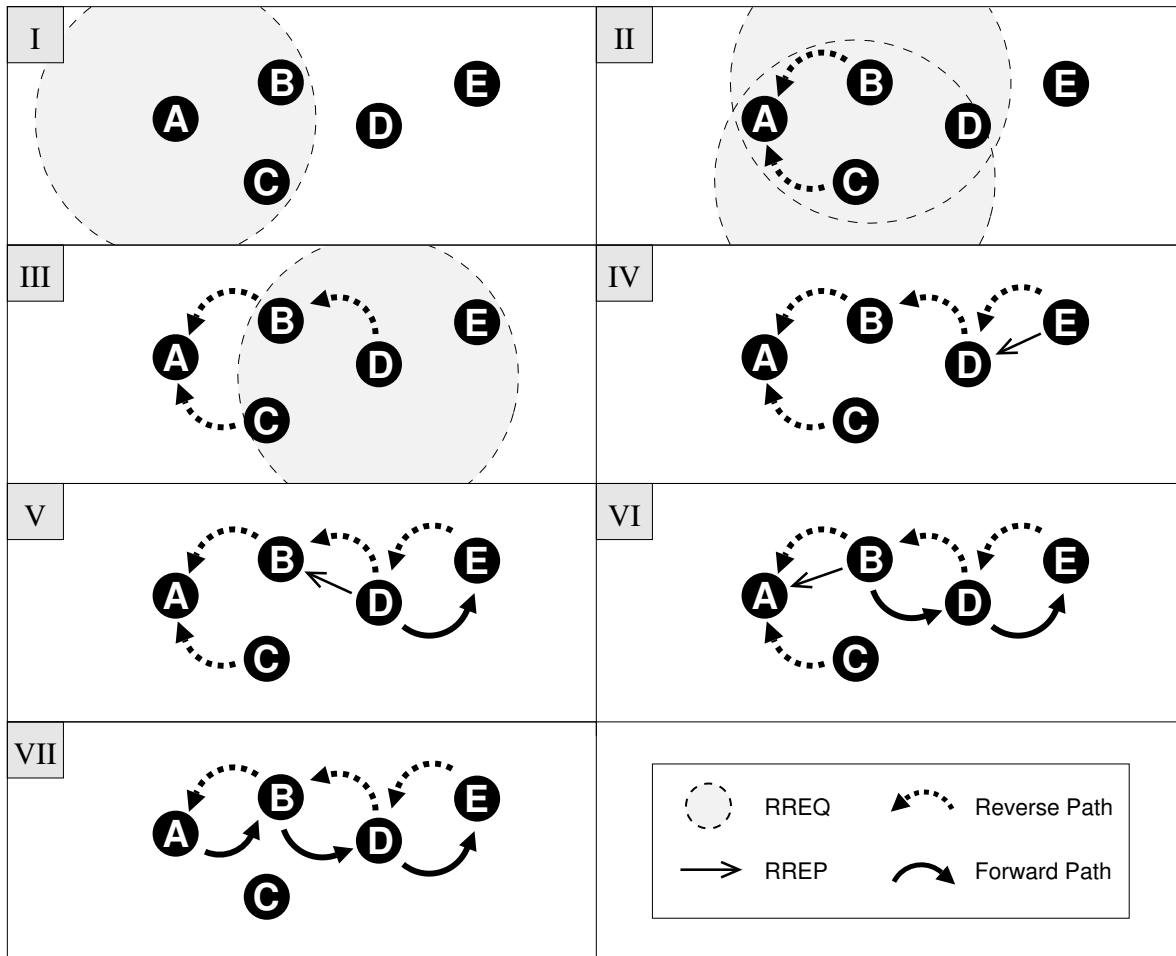


Abbildung 2.6 Der reaktive Routen-Aufbau des AODV-Protokolls.

Die Aufgabe der Aufrechterhaltung von Routen zur Berücksichtigung von Mobilität erfüllt AODV, indem es gegebenenfalls die Suche nach neuen Routen anstößt. Wird etwa innerhalb einer Route bei der Weiterleitung einer Unicast-Dateneinheit von dem Medienzugriffsprotokoll ein Link-Bruch detektiert, so wird dieser dem AODV-Protokoll gemeldet. Da das nächste Endgerät innerhalb der Route nicht mehr erreichbar ist, markiert AODV die Route als ungültig. Um dem ursprünglichen Sendergerät dies mitzuteilen, versendet das Endgerät, welches den Link-Bruch detektiert hat, auf dem Reverse Path ein *RouteError*-Paket (*RERR*). Erreicht das Paket das Sendergerät, kann dieses die Suche nach einer neuen Route anstoßen.

Zusätzlich zu den genannten Protokollmechanismen verfügt AODV über einige Mechanismen, welche der Vollständigkeit halber kurz erwähnt werden sollen, ohne jedoch die genauen, darin umgesetzten Abläufe näher zu diskutieren.

– *Das Verwerfen ungenutzter Routen:* Wird eine Route für eine bestimmte Dauer nicht für die Übertragung von Dateneinheiten genutzt, wird sie automatisch verworfen. Dies

beruht auf der Annahme, dass die Route zum Zeitpunkt der nächsten Nutzung aufgrund von Mobilität bereits gebrochen sein wird. Insofern wird zu diesem Zeitpunkt bevorzugt nach einer neuen Route gesucht.

– *Die Begrenzung der Suche nach Zielgeräten:* Da jedes Endgerät in die Weiterleitung involviert wird, verursacht insbesondere das Fluten von RREQ-Paketen zahlreiche Medienzugriffe und somit eine deutliche Belastung des drahtlosen Ad-hoc-Netzes. Handelt es sich bei dem gesuchten Empfängergerät um ein nur wenig entferntes Endgerät, kann auf das Netz-weite Fluten des RREQ-Pakets verzichtet werden. Hierzu wird die *Lebenszeit* von RREQ-Paketen auf einem Maximalwert begrenzt: Der gewählte Wert entspricht der Anzahl maximaler Weiterleitungen des Pakets, nach welcher dieses verworfen wird. Wird auf ein RREQ-Paket nach einer bestimmten Dauer kein RREP erhalten, so wird die Lebenszeit der RREQ-Pakete erhöht und die Suche wiederholt. Während sich durch diesen Mechanismus die Belastung entfernter Endgeräte reduzieren lässt, kann sich der Routen-Aufbau aufgrund der eventuell wiederholt erforderlichen Suchiterationen jedoch stärker verzögern.

– *Das Caching von Routen:* Während des Flutens eines RREQ-Pakets kann dieses von einem Endgerät empfangen werden, welches bereits über eine Route zu dem gesuchten Empfängergerät verfügt. In einem solchen Fall stößt das besagte Endgerät bereits selbst die Zurückleitung eines RREP-Pakets an und beschleunigt somit den Routen-Aufbau.

– *Die lokale Reparatur von Routen:* Stellt ein Endgerät bei der Weiterleitung einer Dateneinheit fest, dass das nächste Endgerät auf der Route zum Empfängergerät nicht erreichbar ist (vgl. Abschnitt 2.1.1.3), so versendet es gewöhnlich ein RERR-Paket zurück zum Sendergerät. AODV verfügt über eine optionale Erweiterung namens *Local Repair*, bei welcher auf das Zurückleiten eines RERR-Pakets verzichtet wird. Stattdessen versucht das Endgerät, welches den Link-Bruch detektiert hat, die Route *lokal* zu reparieren. Dies erfolgt indem es von sich aus erneut nach dem Empfängergerät sucht. Hierzu wird weitgehend der gleiche Mechanismus verwendet wie für den initialen Routen-Aufbau.

2.1.2.2 Optimized Link-State Routing

Bei dem *Optimized Link-State Routing* Protokoll (*OLSR* [2]) handelt es sich um ein proaktives Routing-Protokoll, welches in jedem Endgerät eine globale Sicht auf die innerhalb des drahtlosen Ad-hoc-Netzes vorhandenen Links aufbaut. Da OLSR innerhalb dieser Arbeit insbesondere während der letztendlichen Evaluation nicht zum Einsatz kommt, soll seine Funktionsweise nur knapp wiedergegeben werden.

Die globale Sicht auf vorhandene Links wird von OLSR aufgebaut, indem jedes Endgerät periodisch seine so genannten *Link-States* innerhalb des drahtlosen Ad-hoc-Netzes flutet. Die Link-States eines Endgeräts g werden durch eine Liste von Endgeräten be-

schrieben, zu welchen aus Sicht von g ein Link besteht. Diese Liste wird auch als *1-Hop-Nachbarschaft* von g bezeichnet.

Die Information über die 1-Hop-Nachbarschaft wird erhalten, indem jedes Endgerät periodisch ein so genanntes **Hello-Paket** als Broadcast versendet. Empfängt ein Endgerät g ein Hello-Paket von einem Endgerät g' , so vermerkt g das Endgerät g' als Teil seiner 1-Hop-Nachbarschaft.

Innerhalb der Hello-Pakete kodiert ein Endgerät jeweils seine 1-Hop-Nachbarschaft. Durch den Empfang von Hello-Paketen erhält ein Endgerät g folglich nicht nur Kenntnis seiner 1-Hop- sondern auch seiner *2-Hop-Nachbarschaft*: Zusätzlich zur 1-Hop-Nachbarschaft umfasst diese diejenigen Endgeräte g^* , von welchen g zwar keine Hello-Pakete empfängt, die aber in einem empfangenen Hello-Paket aufgeführt sind.

Die 2-Hop-Nachbarschaft wird herangezogen, um das Fluten der Link-State-Information effizienter zu gestalten. Tatsächlich erweist sich das periodische Fluten von Link-States in Abhängigkeit der Verteilung von Endgeräten innerhalb des drahtlosen Ad-hoc-Netzes als sehr aufwendig. Werden alle Endgeräte in das Fluten involviert, wird ein Link-State-Paket in einem Netzareal mit einer hohen Endgerätdichte auf engem Raum oftmals unnötig wiederholt. Zur Reduktion dieses Aufwands setzt OLSR eine Technik auf Basis so genannter *Multi-Point Relays (MPRs)* ein. Jedes Endgerät g bestimmt für sich selbst eine Menge von MPRs, welche einer Teilmenge der 1-Hop-Nachbarschaft von g entspricht. Die Teilmenge wird so berechnet, dass g jedes Endgerät aus seiner 2-Hop-Nachbarschaft über mindestens einen der gewählten MPRs erreichen kann. Während die Menge der MPRs möglichst klein gehalten wird, werden die letztendlich als MPR ausgewählten 1-Hop-Nachbarn durch Kennzeichnung in den ausgesendeten Hello-Paketen von ihrer Wahl informiert. Wünscht g nun ein Link-State-Paket zu fluten, erfolgt die Weiterleitung der Dateneinheit nur noch durch Endgeräte, welche als MPR gewählt wurden.

Während sich durch die MPR-Technik der Aufwand zum Fluten von Link-States deutlich reduzieren lässt [79], kennt im Anschluss an das Fluten jedes Endgerät alle innerhalb des drahtlosen Ad-hoc-Netzes existierenden Links. Wünscht ein Sendergerät eine Dateneinheit zu einem bestimmten Empfängergerät zu senden, kann es anhand des vorliegenden Wissens eine kürzeste Route zu diesem berechnen und die Dateneinheit entsprechend weiterleiten.

2.1.3 Diskussion drahtloser Ad-hoc-Netze

Die bislang in diesem Abschnitt beschriebenen Protokolle für den Medienzugriff und zur Wegewahl ermöglichen in einem drahtlosen Ad-hoc-Netz die paarweise Kommunikation zwischen Endgeräten. Aus den genannten Eigenschaften des drahtlosen Mediums und

den beschriebenen Protokollverhalten lassen sich zwei Konsequenzen für die Kommunikation in drahtlosen Ad-hoc-Netzen ableiten, welche auch im Zusammenhang dieser Arbeit von Bedeutung sind.

– *Die Reduktion der in Multi-Hop-Kommunikationswegen zur Verfügung stehenden Bandbreite:* Befinden sich zwei Kommunikationspartner in gegenseitiger Übertragungreichweite, so kann annähernd die gesamte von der drahtlosen Kommunikationsschnittstelle unterstützte Bandbreite zur Datenübertragung genutzt werden⁴. Sind die Kommunikationspartner hingegen über einen Multi-Hop-Kommunikationsweg verbunden, so wird die für die Kommunikation zur Verfügung stehende Bandbreite deutlich verringert. Wie in Abbildung 2.7 verdeutlicht ist ergibt sich dies durch die Weiterleitung über die einzelnen Übertragungsabschnitte sowie den Übertragungs- und Erkennungsbereich der Funksignale. Bei der Kommunikation von A nach B beansprucht die von X vorgenommene Weiterleitung der Dateneinheiten zu Y die bei A zur Verfügung stehende Bandbreite. Ebenso nimmt A zusätzlich den von Y durchgeführten Medienzugriff zur Weiterleitung der Dateneinheiten nach B wahr, da sich A innerhalb des Erkennungsbereichs von Y befindet. In dem gezeigten Beispiel könnte von A daher maximal ein Drittel der Bandbreite der drahtlosen Kommunikationsschnittstelle zur Kommunikation mit B ausgeschöpft werden, da aus Sicht von A jede selbst versendete Dateneinheit drei Medienbelegungen verursacht.

– *Die Detektion und Konsequenzen von Link-Brüchen:* Bleibt die Übertragung einer Unicast-Dateneinheit entlang eines Übertragungsabschnitts unbestätigt, so werden von dem Medienzugriffsprotokoll IEEE 802.11 bis zu 6 Übertragungswiederholungen durchgeführt, wobei exponentiell wachsende Wettbewerbsfenster zur zufälligen Verzögerung verwendet werden. Ist auch die letzte Übertragung unbestätigt geblieben, wird die Dateneinheit verworfen und der angenommene Link-Bruch an das Routing-Protokoll gemeldet. Aufgrund der Durchführung der Übertragungswiederholungen verzögert sich die Detektion eines Link-Bruchs. Um einen Eindruck für die dabei tatsächlich zu erwartenden Verzögerungen zu bekommen, sind diese in Abbildung 2.8 dargestellt. Auf der x-Achse ist die Nummer der erfolgten Übertragung aufgetragen, wobei mit 1 die erste, initiale Übertragung der Dateneinheit gemeint ist. Während auf der linken y-Achse die Größe des für eine Übertragung verwendeten Wettbewerbsfensters aufgetragen ist, ist auf der rechten y-Achse die *kumulierte Verzögerung* einer Übertragung dargestellt. Diese summiert alle *Verzögerungen*, welche bis zu (inklusive) dieser Übertragung verwendet wurden: Zusätzliche Wartezeiten, wie beispielsweise Inter Frame Spaces für den Medienzugriff und die Übertragungsdauer der Dateneinheit, werden nicht berücksichtigt. Es zeigt sich, dass allein durch den Back-off-Algorithmus die Detektion eines Link-

⁴Dies gilt lediglich annähernd, da jede Übertragung von Nutzdaten mit zeitlichen Verzögerungen beim Medienzugriff und der Übertragung zusätzlicher Information (wie etwa die Adressen von Endgeräten) verbunden ist.

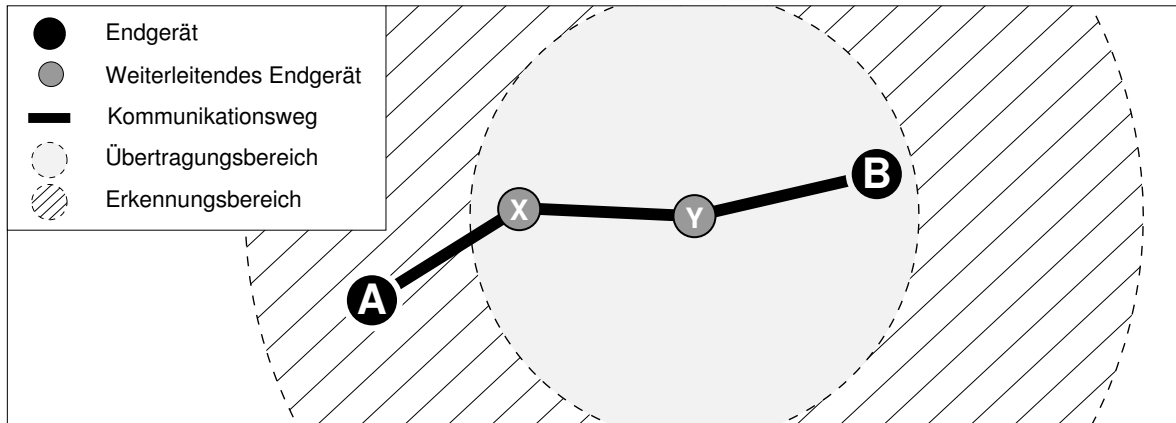


Abbildung 2.7 Auch die Weiterleitung einer Dateneinheit von Y nach B wird von A noch wahrgenommen.

Bruchs um bis zu $60ms$ verzögert wird. Diese Größe ist von Interesse, da sie einen Anhaltspunkt bietet, mit welcher Verzögerung ein Sendergerät von einem Link-Bruch zu einem Kommunikationspartner erfährt. Sendet das Sendergerät in dieser Zeitspanne weitere Dateneinheiten zu dem Kommunikationspartner, so veranlasst das Routing-Protokoll bei Kenntnisnahme der Unerreichbarkeit das Verwerfen dieser Dateneinheiten. Link-Brüche können sich daher vor allem dann negativ auswirken, falls zwischen den Kommunikationspartnern eine zuverlässige Datenübertragung gewünscht wird.

2.2 P2P-basierte Gruppenkommunikation

Die am häufigsten verwendete Kommunikationsform ist die so genannte *Unicast*-Kommunikation, bei welcher die von einem Sendergerät versendeten Dateneinheiten an genau ein Empfängergerät weitergeleitet werden. Entgegen hierzu werden bei einer *Multicast*-Kommunikation, auch *Gruppenkommunikation* genannt, die Dateneinheiten zu mehreren Endgeräten weitergeleitet. Eine Gruppenkommunikation bietet sich damit in Anwendungsszenarien an, in welchen Benutzer Daten mit einer ganze *Gruppe* von Benutzern austauschen [113]. Die einzelnen Mitglieder einer solchen Gruppe sollen fortan auch als *Gruppenmitglieder* bezeichnet werden.

Während das Prinzip der Gruppenkommunikation für das Internet bereits seit den 80er Jahren definiert ist [42, 110, 113], hat sich auch in drahtlosen Ad-hoc-Netzen die Forschungsgemeinde intensiv mit der Problematik befasst [66, 67, 89]. Sowohl im Internet als auch in drahtlosen Ad-hoc-Netzen ist der erwartete Durchbruch eines Protokolls bislang jedoch ausgeblieben. Der mitunter wichtigste Grund hierfür ist, dass die Schwierigkeiten bei Gruppenkommunikationsprotokollen nicht nur in deren Entwicklung und Standardisierung liegen, sondern vor allem in ihrer Inbetriebnahme und Etablierung. An dieser Stelle weisen die bislang hervorgebrachten Ansätze deutliche Defizite auf. In

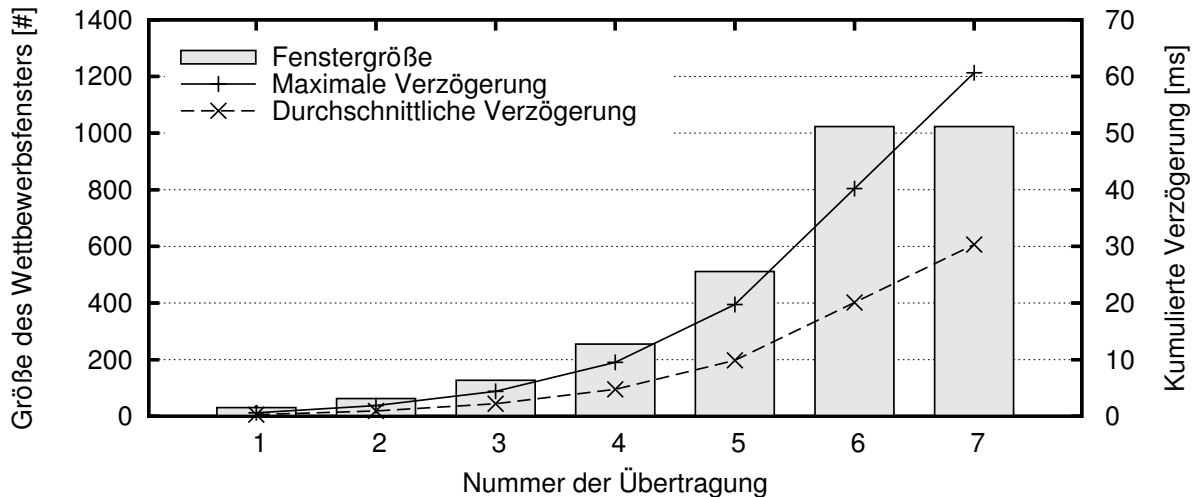


Abbildung 2.8 Exponentielles Wachstum des Wettbewerbsfensters und der Medienzugriffsverzögerung für Unicast-Übertragungen.

der Tat handelt es sich bei den oben genannten Protokollvorschlägen um so genannte *Multicast-Routing-Protokolle*, welche die Mechanismen der Gruppenkommunikation auf Vermittlungsschicht umsetzen. Um die zur Gruppenkommunikation erforderliche Protokollfunktionalität gleichermaßen in einem Netz verfügbar zu machen, ist es erforderlich, dass alle das Netz bildende Geräte das Protokoll beherrschen. Dies ergibt sich aufgrund der Tatsache, dass potenziell jedes dieser Geräte in die zur Gruppenkommunikation essentiellen Aufgaben der *Duplizierung und Weiterleitung* von Dateneinheiten an einzelne Gruppenmitglieder involviert sein kann und daher über Wissen über den Aufbau der *Multicast-Gruppe* verfügen muss. Darüber hinaus wird diese Problematik durch die Tatsache verschärft, dass einzelne Anwendungsszenarien unterschiedliche Anforderungen im Hinblick auf die vorgenommene Datenverteilung haben können: So muss ein Gruppenkommunikationsprotokoll, um den jeweiligen Anforderungen gerecht zu werden, flexibel parametrisierbar sein. Eben diese Flexibilität sowie den hohen, gewünschten Parametrisierbarkeitsgrad in einem gesamten Netz einheitlich zur Verfügung zu stellen, ist nicht nur eine kaum zu bewältigende Herausforderung sondern widerspricht auch den ursprünglichen Entwurfskriterien und -entscheidungen (Einfachheit und Zustandslosigkeit) von Netzen. Dies trifft insbesondere auf ein drahtloses Ad-hoc-Netz zu, welches nicht nur spontan sondern auch von stark heterogenen Endgeräten gebildet wird. Die Vereinheitlichung und die Inbetriebnahme eines von Anwendungen weitläufig konfigurierbaren Gruppenkommunikationsprotokolls erscheint in einem derartigen Umfeld weder bewältigbar noch gerechtfertigt.

Mit dem Erscheinen der ersten *Tauschbörsen-* und *File-sharing-*Anwendungen im Internet [30, 87] sind Ende der 90er Jahre neuartige Kommunikationsprotokolle erschienen,

bei welchen Gruppen von Endgeräten untereinander Daten austauschen. Bei diesen Protokollen handelt es sich um so genannte *Peer-to-Peer-* (kurz *P2P*) oder *Endsystem-* basierte Protokolle. Der Kernaspekt dieser Protokolle liegt darin, *neue Dienste durch Rückgriff auf bestehende Infrastruktur* zu schaffen, indem die für den Dienst erforderliche Funktionalität auf Anwendungsschicht der tatsächlich Dienst-nutzenden Endgeräte verfügbar gemacht wird. Wie dies erfolgt und wie die damit verbundenen Konzepte im Kontext einer Gruppenkommunikation genutzt werden können soll in den nächsten Abschnitten kurz beschrieben werden. Die dabei betrachteten Aspekte sollen im Wesentlichen der Begriffsdefinition dienen. Aus diesem Grund werden die aufgezeigten Konzepte an dieser Stelle noch allgemein gehalten und nur an manchen Stellen auf den Einsatz in drahtlosen Ad-hoc-Netzen bezogen.

2.2.1 Allgemeine Prinzipien

Im Kontext eines Gruppenkommunikationsprotokolls bedeutet ein P2P-basierter Ansatz, dass lediglich die Gruppenmitglieder über Wissen über den Aufbau der Multicast-Gruppe verfügen. Als Folge dessen sind auch nur die Gruppenmitglieder in die für Gruppenkommunikationsprotokolle typischen Aufgaben der Duplizierung und Weiterleitung von Dateneinheiten involviert. Die eigentliche Weiterleitung der Dateneinheiten zwischen den Gruppenmitgliedern erfolgt durch Nutzung von etablierten Unicast-Routing-Protokollen wie sie in Abschnitt 2.1.2 beschrieben sind.

Zur Umsetzung einer P2P-basierten Gruppenkommunikation ist es erforderlich, dass sich die Gruppenmitglieder untereinander selbstständig organisieren und die im Laufe der Gruppenkommunikation ablaufende Datenverteilung unter sich aufteilen. Hierfür werden die Gruppenmitglieder anhand einer dezentralen, sich selbst organisierenden Netzstruktur verbunden, dem so genannten *Overlay-Netz*. Dieses ist die Hauptkomponente einer P2P-basierten Gruppenkommunikation, so dass es nachfolgend näher beschrieben werden soll.

2.2.2 Das Overlay-Netz

Ein Overlay-Netz entsteht, indem zwischen den Gruppenmitgliedern der Multicast-Gruppe so genannte *Overlay-Verbindungen* aufgebaut werden. Bei diesen handelt es sich um Kommunikationsbeziehungen zwischen den jeweiligen Endpunkten der Overlay-Verbindungen, so dass ihnen im zugrunde liegenden Netz entsprechende Kommunikationswege unterliegen. Dieser Sachverhalt ist in Abbildung 2.9 dargestellt, in welcher eine Reihe von Endgeräten, in diesem Fall Gruppenmitglieder, aufgetragen sind. Die Abbildung ist in zwei Hälften aufgeteilt: Während in der unteren Hälfte die Kommunikationswege durch das nur schematisch dargestellte, zugrunde liegende Netz abgebildet sind, ist in der oberen Hälfte das zwischen den Gruppenmitgliedern aufgebaute Overlay-Netz mit den entsprechenden Overlay-Verbindungen sichtbar.

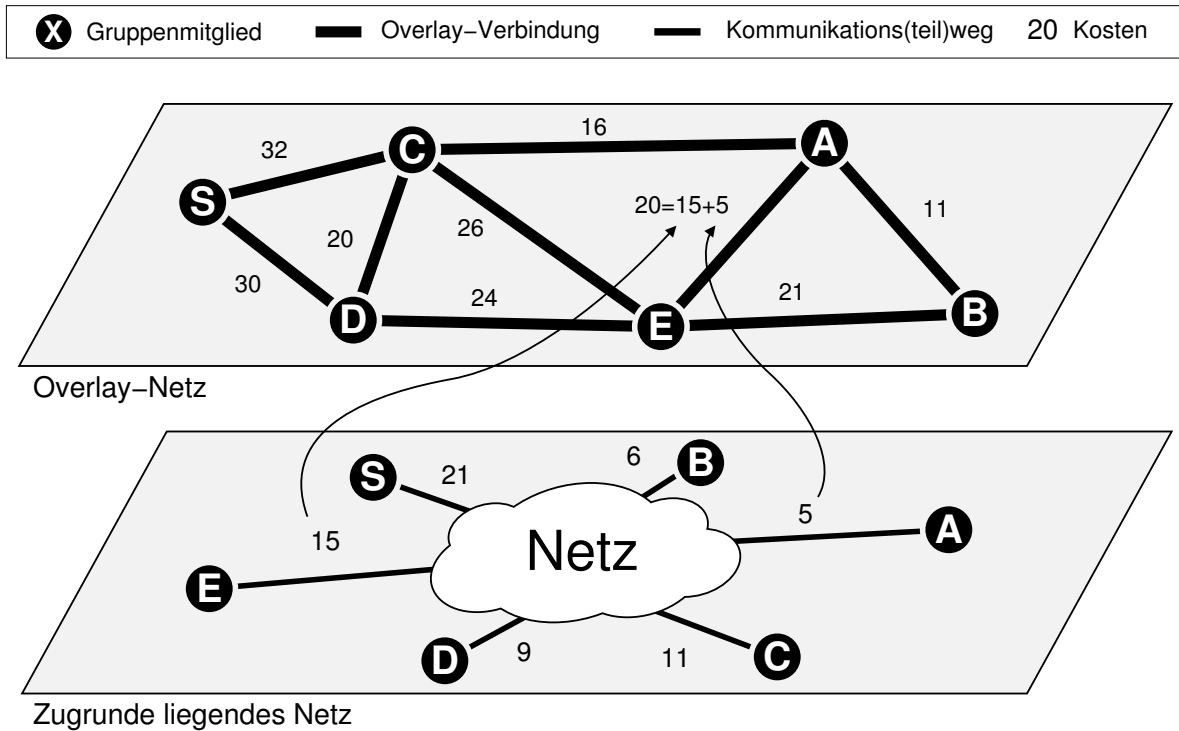


Abbildung 2.9 Ein Overlay-Netz wird über einem zugrunde liegenden Netz aufgebaut, welches Kommunikationswege bereitstellt.

Overlay-Verbindungen werden zwischen Gruppenmitgliedern nicht willkürlich aufgebaut. Vielmehr wird über ihren Aufbau anhand spezieller Algorithmen entschieden, welche gemäß einer Metrik eine *Bewertung* der Verbindungen vornehmen. So werden *Kosten-* oder auch *Nutz-*Faktoren ermittelt, aufgrund welcher Overlay-Verbindungen gegenüber anderen bevorzugt werden können. Ein einfaches Beispiel ist in Abbildung 2.9 aufgetragen, wobei die dargestellten Kosten jeweils die Kosten desjenigen Kommunikations(teil)wegs⁵ widerspiegeln, über welchen ein Gruppenmitglied den Zugang zu dem zugrunde liegenden Netz bildet. So erfordert der Zugang von Gruppenmitglied A die Kosten 5, während sich für E die Kosten auf 15 belaufen. Zur Vereinfachung seien die Kosten der Weiterleitung im zugrunde liegenden Netz als konstant angenommen, so dass sie hier nicht näher betrachtet werden. Für den gesamten Kommunikationsweg von A nach E summieren sich die Kosten der einzelnen Kommunikations(teil)wege zu $5+15 = 20$, was den Kosten der Overlay-Verbindung zwischen A und E entspricht. In Abhängigkeit der Regeln, nach welchen Overlay-Verbindungen aufgebaut werden, entsteht für ein Overlay-Netz eine bestimmte *Overlay-Topologie*. Overlay-Topologien kön-

⁵Der Begriff des Kommunikations(teil)wegs verdeutlicht, dass es sich dabei lediglich um einen Abschnitt eines vollständigen Kommunikationswegs handelt. Ein vollständiger Kommunikationsweg setzt sich im gezeigten Beispiel immer aus zwei Kommunikations(teil)wegen zusammen.

nen stark unterschiedliche Eigenschaften haben, etwa bezüglich der später erläuterten Weiterleitung von Dateneinheiten.

Overlay-Verbindungen werden zum Austausch verschiedener Pakete genutzt. Diese umfassen die von einer *Anwendung* ausgesendeten *Nutzdatenpakete* (kurz *Datenpakete*) oder um so genannte *Kontrollpakete*. Mit Kontrollpaketen werden dabei jegliche Pakete bezeichnet, welche für die Verwaltung des Overlay-Netzes anfallen. Typische Kontrollpakete sind beispielsweise *Heartbeat*-Pakete, welche von Gruppenmitgliedern periodisch untereinander ausgetauscht werden und Information über die in der Overlay-Topologie derzeit aufgebauten Overlay-Verbindungen beinhalten. In Abhängigkeit der Algorithmen zum Aufbau einer Overlay-Topologie sowie der Anzahl eingegliedeter Gruppenmitglieder und aufgebauter Overlay-Verbindungen, erfordert die Verwaltung eines Overlay-Netzes eine unterschiedliche Menge an Kontrollpaketen: Diese bilden den so genannten *Kontrollfluss*. Da der zwischen den Gruppenmitgliedern ausgetauschte Kontrollfluss das zugrunde liegende Netz belastet, ist er oftmals ausschlaggebend für die Skalierbarkeit einer Overlay-Topologie.

2.2.2.1 Gruppenverwaltung

Ein Overlay-Netz bildet ein in sich abgeschlossenes Netz, so dass lediglich die darin vertretenen Gruppenmitglieder über Information zu diesem Netz verfügen. Um ein außenstehendes Endgerät in ein Overlay-Netz *einzugliedern* muss diesem ein *Einstiegspunkt* bekannt sein, d.h. ein Gruppenmitglied, welches bereits in das Overlay-Netz eingegliedert ist. Von diesem Gruppenmitglied kann dann Information über weitere Gruppenmitglieder bezogen werden, über welche dann die letztendliche Eingliederung des neuen Endgeräts in die Overlay-Topologie erfolgt.

Um außenstehenden Endgeräten diesen Prozess des *Gruppenbeitritts*, auch *Bootstrapping* genannt, zu ermöglichen, wird üblicherweise eine vorweg bekannte, zentrale Instanz herangezogen. Auf dieser sind Listen von bereits beigetretenen Gruppenmitgliedern abgelegt, die von außenstehenden, neu beitretenden Endgeräten abgerufen werden können. Während sich dieses Verfahren im Internet als quasi Standard etabliert hat, sind in spontanen Umgebungen wie drahtlosen Ad-hoc-Netzen andere Mechanismen erforderlich, auf welche in Kapitel 3 näher eingegangen wird.

2.2.2.2 Paketduplizierung und -Weiterleitung

Sind die Gruppenmitglieder durch ein Overlay-Netz verbunden, können die darin bestehenden Overlay-Verbindungen zur Weiterleitung von Datenpaketen zwischen den einzelnen Gruppenmitgliedern genutzt werden. Hierfür benötigen die Gruppenmitglieder Information, aufgrund welcher sie darauf schließen können, an welche der im Overlay-Netz benachbarten Gruppenmitglieder ein Datenpaket weiterzuleiten ist. Ähnlich zu

Abschnitt 2.1.2 spricht man hierbei von einem *Overlay-Routing*-Protokoll, welches über einer Overlay-Topologie eingesetzt wird.

Bei der Weiterleitung eines Datenpakets entsteht innerhalb des Overlay-Netzes in Abhängigkeit des Sendergeräts ein *Multicast-Baum*. In diesem existiert ausgehend von der Wurzel des Baums (dem Sendergerät) ein *Pfad* zu jedem Gruppenmitglied. Jeder Abschnitt des Pfades entspricht der Weiterleitung entlang einer Overlay-Verbindung, so dass man im Kontext des Overlay-Routing von einem *Overlay-Hop* spricht. Betrachtet man das Beispiel in Abbildung 2.10, so agiert das Gruppenmitglied S als Sendergerät. Ausgehend von S wird der Multicast-Baum derart berechnet, dass der Pfad von S zu jedem Gruppenmitglied minimale Kosten aufweist, wobei die Kosten eines Pfades als Summe der Kosten der einzelnen, darin enthaltenen Overlay-Verbindungen zu sehen ist. So sendet S Duplikate seiner Datenpakete jeweils an C und D. Während D wiederum ein Duplikat an E leitet, erfolgt von C eine Weiterleitung zu Gruppenmitglied A, welches letztendlich wiederum eine Kopie des Datenpakets an B sendet.

Je nach aufgebauter Overlay-Topologie und dem eingesetzten Overlay-Routing-Protokoll werden Multicast-Bäume auf unterschiedliche Art und Weise berechnet. Um einen Multicast-Baum entsprechend dem aus Abbildung 2.10 zu berechnen, ist beispielsweise in den Endgeräten ein globales Wissen über die aufgebaute Overlay-Topologie erforderlich. Während sich mit diesem zwar tatsächlich optimale Multicast-Bäume berechnen lassen und sich die Weiterleitung der Datenpakete somit effizient gestalten lässt, muss dies jedoch wiederum mit einem Informationsaustausch über die Struktur der Overlay-Topologie zwischen den Gruppenmitgliedern erkaufte werden. Dieser zählt zu dem in Abschnitt 2.2.2 eingeführten Kontrollfluss und wirkt sich als zusätzliche Belastung des zugrunde liegenden Netzes aus. Im genannten Beispiel, in welchem die vollständige Topologieinformation ausgetauscht wird, würde der dabei anfallende Aufwand nur schlecht mit der Anzahl der Gruppenmitglieder skalieren, so dass das Overlay-Netz nur im Zusammenhang mit kleinen Multicast-Gruppen sinnvoll einsetzbar wäre.

2.2.3 Zuverlässigkeit

In Abhängigkeit der betriebenen Anwendung muss die von dem Gruppenkommunikationsprotokoll durchgeführte Datenverteilung eine bestimmte *Zuverlässigkeit* aufweisen: Während manche Anwendungen mit einzelnen, fehlenden Datenpaketen umgehen können, erfordern andere Anwendungen wiederum die korrekte Zustellung jedes Datenpakets. Die Umsetzung von Zuverlässigkeitsmechanismen im Sinne von der *Vollständigkeit*, der *Reihenfolgetreue* und der *Duplikatfreiheit* von Datenpaketen erweist sich im Rahmen einer Gruppenkommunikation als komplex, da die Aufgaben der Weiterleitung von Datenpaketen einerseits und der erneuten Übertragung eventuell verlorener Datenpakete andererseits strikt getrennt voneinander betrachtet werden. Multicast-Routing-

2. Grundlagen

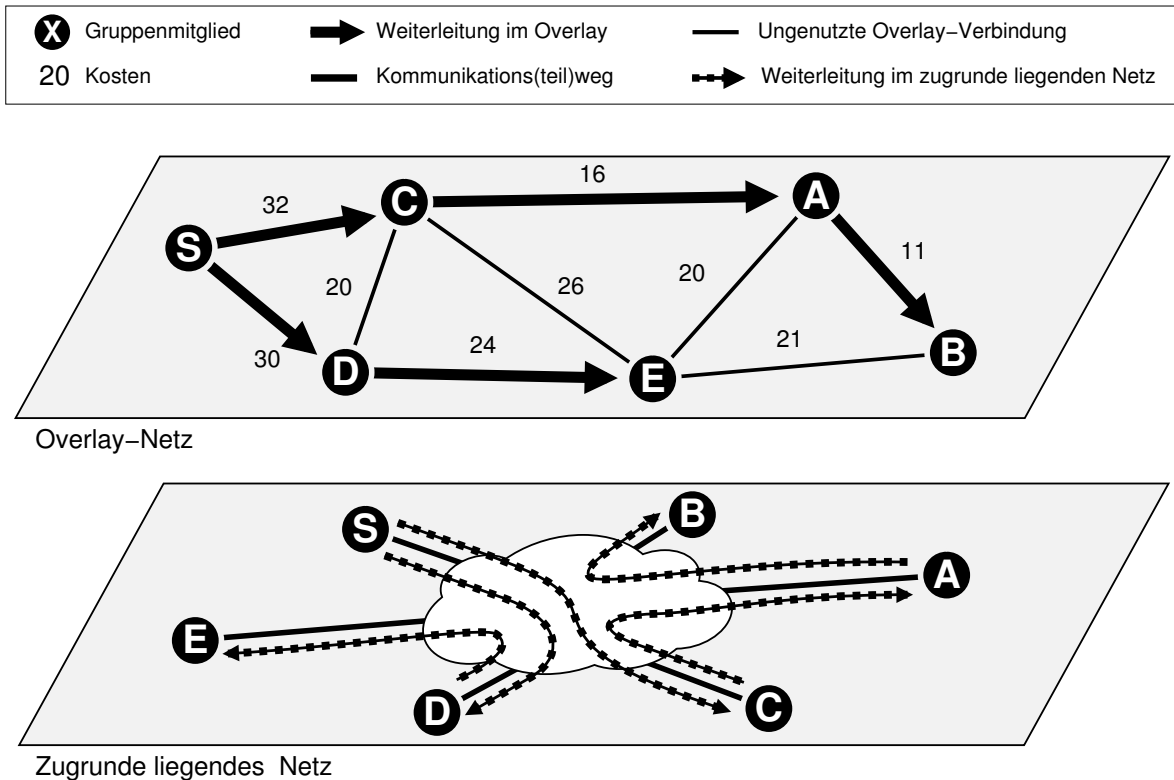


Abbildung 2.10 Nutzung einer Overlay-Topologie zur Weiterleitung von Datenpaketen zwischen den Gruppenmitgliedern.

Protokolle wie DVMRP [42, 110] sind beispielsweise lediglich für die Weiterleitung von Datenpaketen verantwortlich, so dass Zuverlässigkeitsaspekte durch zusätzliche Protokolle umzusetzen sind. Diese Protokolle arbeiten jedoch ausschließlich auf einer Ende-zu-Ende-Sicht, so dass einerseits lediglich Empfängergeräte das Fehlen von Datenpaketen erkennen können und andererseits die Anforderung einer Übertragungswiederholung ohne Einblick in die tatsächlich stattgefundenene Weiterleitung des Datenpakets erfolgen muss. Zuverlässigkeitsmechanismen, welche eine Übertragungswiederholung direkt durch das ursprüngliche Sendergerät veranlassen, erweisen sich daher als äußerst ineffizient: Dies ergibt sich aufgrund der Tatsachen, dass einerseits die zu jedem Empfängergerät erfolgende Übertragungswiederholung nicht unbedingt von jedem Empfängergerät gewünscht wird. Andererseits ist das Sendergerät eventuell kostspieliger zu erreichen als ein umliegendes Empfängergerät, welches das Datenpaket empfangen hat und daher zur Durchführung einer Übertragungswiederholung zur Verfügung steht.

Zur Umsetzung von Zuverlässigkeitsmechanismen bieten P2P-basierte Gruppenkommunikationsprotokolle bedeutende Vorteile, da hier die Weiterleitung von Datenpaketen zwischen den Gruppenmitgliedern anhand von Ende-zu-Ende-basierten Overlay-Verbindungen erfolgt. So kann die Zuverlässigkeit der *Multicast-Kommunikation* erreicht

werden, indem Zuverlässigkeitsmechanismen für die innerhalb des Overlay-Netzes stattfindende *Unicast-Kommunikation* definiert werden. Derartige Zuverlässigkeitsmechanismen werden in so genannten *Transportprotokollen* umgesetzt, deren prominentester Vertreter TCP [78] ist. Während sich TCP im Internet als Standardprotokoll zur Umsetzung von Zuverlässigkeit auf Ende-zu-Ende-Verbindungen durchgesetzt hat, ist es, wie später verdeutlicht werden wird, für drahtlose Ad-hoc-Netze jedoch ungeeignet, so dass andere Lösungsansätze gefunden werden müssen.

2.2.4 Diskussion von P2P-Multicast

P2P-basierte Gruppenkommunikationsprotokolle sind neuartige Ansätze zur Umsetzung einer Gruppenkommunikation. Während sich dabei die Mitglieder einer Multicast-Gruppe vollständig dezentral und selbstständig innerhalb eines Overlay-Netzes organisieren, müssen die hierfür erforderlichen Protokollmechanismen lediglich in den Gruppenmitgliedern verfügbar gemacht werden. Aus diesem Grund wird nicht nur die Inbetriebnahme des Protokolls vereinfacht, sondern auch seine Flexibilität gegenüber der Spezialisierung auf die Anforderungen bestimmter Anwendungen erhöht: So kann sowohl das zwischen den Gruppenmitgliedern aufgebaute Overlay-Netz als auch das eingesetzte Transportprotokoll zur Umsetzung von Zuverlässigkeitsmechanismen auf eine bestimmte Anwendung angepasst werden.

Während P2P-basierte Ansätze demnach deutliche Vorteile gegenüber Multicast-Routing-Protokollen zeigen, werden diese teilweise mit Nachteilen erkauft.

Zu nennen wäre hier zuerst die Effizienz P2P-basierter Protokolle bei der Weiterleitung von Datenpaketen, welche niemals die Effizienz von Multicast-Routing-Protokollen übertreffen kann. Dies liegt darin begründet, dass bei P2P-basierten Ansätzen die Duplizierung von Datenpaketen lediglich von den Gruppenmitgliedern vorgenommen wird und nicht von dem zugrunde liegenden Netz selbst. Bei der Weiterleitung ergeben sich demnach erhöhte Kosten, da die Datenpakete über Overlay-Verbindungen von Gruppenmitglied zu Gruppenmitglied geleitet werden anstatt den direkten Kommunikationsweg zwischen dem Sendergerät und den Gruppenmitgliedern zu bestreiten. Dies soll anhand von Tabelle 2.1 verdeutlicht werden. In dieser sind auf Basis des Beispiels aus Abbildung 2.10 die *aus Sicht des Sendergeräts S* zum Erreichen der Gruppenmitglieder erforderlichen Kosten aufgetragen. Für ein Multicast-Routing-Protokoll ergeben sich die Kosten aus der Summe der Kosten der Kommunikations(teil)wege des Sendergeräts S und des jeweils zu erreichenden Gruppenmitglieds, da die Duplizierung eines Datenpakets innerhalb des Netzes stattfindet. Für den P2P-basierten Ansatz hingegen ergeben sich die Kosten aus der Summe der Kosten der einzelnen Overlay-Verbindungen, über welche ein Datenpaket bis zum Erreichen des Gruppenmitglieds geleitet werden muss. In der letzten Tabellenzeile sind wiederum die Gesamtkosten der Weiterleitung

	Kosten Multicast-Routing	Kosten P2P-basierter Ansatz
A	$26 = (21+5)$	$48 = (21+11)+(11+5)$
B	$27 = (21+6)$	$59 = (21+11)+(11+5)+(5+6)$
C	$32 = (21+11)$	$32 = (21+11)$
D	$30 = (21+9)$	$30 = (21+9)$
E	$36 = (21+15)$	$54 = (21+9)+(9+15)$
Σ	$67 = 21+5+6+11+9+15$	$113 = (21+11)+(11+5)+(5+6)+(21+9)+(9+15)$

Tabelle 2.1 Im Vergleich zu Multicast-Routing-Protokollen entstehen bei der Datenweiterleitung durch P2P-basierte Protokolle höhere Kosten.

eines Datenpakets aufgetragen. Diese berechnen sich aus der Summe der Kosten der für die Weiterleitung des Datenpakets beanspruchten Kommunikations(teil)wege.

Insbesondere aus der letzten Tabellenzeile und der in Abbildung 2.10 aufgezeigten Weiterleitungen im zugrunde liegenden Netz wird ein weiterer Nachteil P2P-basierter Gruppenkommunikation deutlich: Im Zuge der Weiterleitung eines Datenpakets durchläuft dieses eventuell mehrfach die gleichen Kommunikations(teil)wege, so dass diese stärker belastet werden und für die Weiterleitung entsprechend mehr Bandbreite benötigt wird. Die tatsächlich anfallende Belastung eines Kommunikations(teil)wegs steht dabei in direktem Zusammenhang zu der aufgebauten Overlay-Topologie und dem darüber eingesetzten Overlay-Routing, da diese unmittelbaren Einfluss auf die Duplizierung und die Weiterleitung von Datenpaketen haben. Insbesondere in drahtlosen Ad-hoc-Netzen sind derartige Effekte geeignet zu berücksichtigen, da hier die Weiterleitung von Datenpaketen über Multi-Hop-Kommunikationswege die verfügbare Bandbreite weiter reduziert (vgl. Abschnitt 2.1.3).

2.3 Zusammenfassung

In diesem Kapitel wurden diverse Grundlagen gelegt, welche zum Verständnis dieser Arbeit erforderlich sind. Es wurde insbesondere der Begriff des *drahtlosen Ad-hoc-Netzes* definiert. In diesem Zusammenhang wurde aufgezeigt wie mit einer drahtlosen Kommunikationsschnittstelle ausgerüstete Endgeräte auf das drahtlose Medium zugreifen können, und wie der Zugriff mehrerer Endgeräte durch ein *verteiltetes Medienzugriffsprotokoll* koordiniert werden kann. Auf diese Weise wurde die Kommunikation zwischen Endgeräten ermöglicht, welche sich in gegenseitiger Übertragungreichweite befinden. Die anschließend näher betrachteten *Verfahren zur Wegewahl* in drahtlosen Ad-hoc-Netzen ermöglichen unter Berücksichtigung der potenziellen *Mobilität* von drahtlosen Endgeräten sowohl das Auffinden von Endgeräten in mehreren Übertragungsabschnitten Entfernung als auch die Kommunikation mit diesen.

Im Anschluss wurde eine kurze Einführung in *P2P-basierte Gruppenkommunikationsprotokolle* gegeben. Diese teilen die Weiterleitung von Datenpaketen anhand eines dezentral organisierten *Overlay-Netzes* zwischen den einzelnen Gruppenmitgliedern auf. Im Vergleich zu üblichen *Multicast-Routing-Protokollen* besitzen P2P-basierte Ansätze zahlreiche Vorteile, darunter eine *einfachere Inbetriebnahme* sowie eine *erhöhte Flexibilität* gegenüber der Heterogenität von Anwendungsanforderungen. Da diese Vorteile jedoch unter anderem mit einer *ineffizienteren Weiterleitung* der Datenpakete erkauft werden, ist bei der Konzeption von P2P-basierten Gruppenkommunikationsprotokollen besondere Sorgfalt und die Berücksichtigung zahlreicher Eigenschaften und Anforderungen vonnöten.

3. P2P-Multicast-Protokolle in drahtlosen Ad-hoc-Netzen

Die Ende-zu-Ende-Kommunikation von Endgeräten auf Basis von Overlay-Netzen wurde seit dem Auftauchen erster Internet-Tauschbörsen [30, 87] zunehmend populär. Die Dienste, welche über Overlay-Netze angeboten und genutzt werden können, beschränken sich heutzutage jedoch nicht nur auf das Tauschen von Dateien: Stattdessen zeigen sie einen weitaus vielfältigeren Charakter, wie beispielsweise die Verwaltung und die Durchführung von Transaktionen in verteilten, elektronischen Märkten [39]. Die Aufgabe, vor welche Overlay-Netze im Allgemeinen gestellt werden und welche sich in den jeweiligen Entwürfen von Overlay-Topologien und -Mechanismen widerspiegelt, ist die Adressierung und Datenübertragung zu bestimmten Endgeräten innerhalb eines Overlay-Netzes. In aktuellen Forschungsarbeiten werden derzeit drei unterschiedlichen Klassen von Adressierbarkeit unterschieden und untersucht:

- Die Adressierung *eines bestimmten Endgeräts* innerhalb des Overlay-Netzes: Derartige Overlays werden als *strukturierte Overlay-Topologien* bezeichnet und dienen üblicherweise der Umsetzung verteilter Hash-Tabellen (*Distributed Hash Tables, DHTs* [34, 61, 84, 88, 98]). Sie ermöglichen durch Verwendung eindeutiger Knotenkennungen (engl. *Node IDs*) das Auffinden des mit einer Kennung assoziierten Endgeräts innerhalb des Overlay-Netzes. Durch geschickte Vermaschung der Endgeräte liegt der zur Lokalisierung eines Endgeräts anfallende Vermittlungsaufwand üblicherweise in der Klasse $O(\log n)$ (mit n als Anzahl der innerhalb des Overlay vertretenen Endgeräte) und zeigt sich daher als äußerst effizient. Strukturierte Overlay-Topologien eignen sich vorzugsweise für den Einsatz in Anwendungen,

welche Tauschbörsen oder eine verteilte Speicherung von Daten umsetzen: Aufbauend auf einer Hash-Funktion und einem daraus resultierenden Schlüsselraum erstellen diese Anwendungen ein verteiltes Verzeichnis, welches auf die einzelnen Knotenkennungen der innerhalb des Overlay-Netzes vertretenen Endgeräte abgebildet wird.

- Die Adressierung vom *mindestens einem Endgerät* innerhalb des Overlay-Netzes: Diese Ansätze widmen sich typischerweise der Umsetzung so genannter Anycast-Anfragen [23, 32], bei welchen sichergestellt wird, dass eine ausgesendete Anfrage an mindestens ein Mitglied des Overlay-Netzes vermittelt wird. Derartige Techniken haben bislang keine herausragende Bedeutung erlangt, so dass sie hier lediglich der Vollständigkeit wegen aufgeführt sind.
- Die Adressierung *aller Endgeräte* innerhalb des Overlay-Netzes: Jene Overlay-Netze sind darauf ausgelegt, eine Verteilstruktur aufzubauen, über welche ein ausgesendetes Datenpaket möglichst effizient zu allen im Overlay-Netz vertretenen Endgeräten geleitet wird. Sie bieten die Umsetzung von P2P- bzw. Overlay-Multicast-Mechanismen und bilden den Fokus dieser Arbeit.

Strukturierte Overlay-Topologien lassen sich durchaus auch zur Umsetzung einer Gruppenkommunikation auf Anwendungsschicht verwenden [31, 85]. Aufgrund ihrer Spezialisierung auf die Lokalisierung einzelner Endgeräte anstatt der Adressierung aller Overlay-Mitglieder, weisen ihre Mechanismen zur Topologiekonstruktion und Topologierhaltung sowie zur Datenvermittlung jedoch andere Charakteristiken auf als speziell für diesen Zweck entworfene Overlay-Multicast-Ansätze. Im Vergleich zu diesen erzielten strukturierte Overlay-Topologien daher eine geringere Leistungsfähigkeit bzw. gehen mit einem höheren Aufwand einher [101].

Aufgrund der zunehmenden Popularität von P2P-Anwendungen im Internet einerseits und der noch immer bestehenden Schwierigkeiten bei der Inbetriebnahme von Multicast-Routing-Protokollen andererseits, werden P2P-basierte Gruppenkommunikationsprotokolle zusehends als lohnenswerte Alternative gesehen. So wurden in den vergangenen Jahren für das Internet bereits zahlreiche Overlay-Topologien entwickelt und untersucht [1, 5, 6, 38, 75, 85, 94, 114]. Auch in drahtlosen Ad-hoc-Netzen werden Overlay-Topologien zunehmend erforscht [37, 48, 51], da hier die Betriebsumgebung (das drahtlose Ad-hoc-Netz) und die Grundsätze P2P-basierter Protokolle zahlreiche Ähnlichkeiten aufweisen: Beide Strukturen arbeiten vollständig dezentral und selbst organisierend, so dass in der Literatur oftmals von Synergieeffekten die Rede ist [54, 93].

In diesem Kapitel sollen die wichtigsten bislang entwickelten Overlay-Topologien miteinander verglichen und, im Kontext ihres Betriebs in drahtlosen Ad-hoc-Netzen, auf

potenzielle Schwachstellen untersucht werden. Hierfür wird in Abschnitt 3.1 zunächst eine Liste von Anforderungen und Bewertungskriterien erarbeitet, welche sich als entscheidend für das effiziente Ablaufen einer P2P-basierten Gruppenkommunikation in drahtlosen Ad-hoc-Netzen erweisen. Dabei wird sich insbesondere herausstellen, dass zur Berücksichtigung der Mobilität von Endgeräten eine Overlay-Topologie fortlaufend anhand einer Metrik bewertet und entsprechend angepasst werden muss. Zu diesem Zweck sollen in Abschnitt 3.2 die gängigsten Bewertungsmetriken vorgestellt und im Hinblick auf Vor- und Nachteile diskutiert werden. Bevor in Abschnitt 3.4 die eigentliche Bewertung von Overlay-Topologien erfolgt, wird in Abschnitt 3.3 die Möglichkeit zur Integration von Anwendungs-spezifischen Zuverlässigkeitsmechanismen in einen P2P-basierten Gruppenkommunikationsdienst angesprochen: Dies erweist sich insbesondere dann als unverzichtbar, wenn der Gruppenkommunikationsdienst die Unterstützung unterschiedlicher Anwendungen ermöglichen soll, welche potenziell heterogene Anforderungen an die Zuverlässigkeit der Datenverteilung haben.

3.1 Anforderungen und Bewertungskriterien

Die charakteristischen Eigenschaften drahtloser Ad-hoc-Netze führen zu besonderen Anforderungen an eine zwischen den Gruppenmitgliedern aufgebaute Overlay-Topologie. So ist nicht jede, beispielsweise aus dem Internet bekannte, Overlay-Topologie zur Gruppenkommunikation innerhalb eines Ad-hoc-Netzes geeignet: Dies liegt darin begründet, dass die im Festnetz getroffenen Annahmen über die Eigenschaften des zugrunde liegenden Kommunikationssystems (wie die vernachlässigbaren Änderungen der physikalischen Netztopologie) für drahtlose Ad-hoc-Netze nicht zutreffend sind. Dieser Abschnitt fasst daher wichtige Anforderungen zusammen, welche von einer Overlay-Topologie in drahtlosen Ad-hoc-Netzen zu erfüllen sind, um über die Kommunikationsdauer hinweg einen effizienten Betrieb bzw. eine effiziente Kommunikation zu gewährleisten.

3.1.1 Die Suche nach vorhandenen Gruppenmitgliedern

Um seine Eingliederung in eine Overlay-Topologie zu bewerkstelligen, benötigt ein beitretenes Gruppenmitglied die Kenntnis mindestens eines anderen Gruppenmitglieds, welches bereits erfolgreich in das Overlay eingegliedert ist. Ein solches Gruppenmitglied kann gegebenenfalls Wissen über weitere Gruppenmitglieder bereitstellen und so bei der Eingliederung des neuen Gruppenmitglieds behilflich sein. Obgleich die Entwicklung neuartiger Verfahren zur Bestimmung beigetretener Gruppenmitglieder vorschreitet [40], erfordern die bislang im Festnetz etablierten Verfahren üblicherweise eine zentrale Anlaufstelle, einen so genannten *Host Cache*, auch *Rendez-vous-Knoten* (*Rendez-vous-Point*, *RVP*) genannt. Dieser stellt aktuell gehaltene Listen zur Verfügung, welche Adressen von innerhalb eines Overlay vertretenen Gruppenmitgliedern

aufnehmen. In drahtlosen Ad-hoc-Netzen kann eine solche Anlaufstelle jedoch nicht vorausgesetzt werden, da derartige Netze durch einen spontanen Zusammenschluss von Endgeräten entstehen, und die dauerhafte Erreichbarkeit eines bestimmten, eindeutig bekannten Endgeräts demnach nicht notwendigerweise gegeben ist. Zur Bestimmung bereits beigetretener Gruppenmitglieder können stattdessen so genannte *inkrementelle Ringsuchen* (engl. *Expanding Ring Search, ERS*) eingesetzt werden: Diese greifen auf ein partielles Fluten von Anfragepaketen innerhalb des drahtlosen Ad-hoc-Netzes zurück. Durch die vorgenommene Beschränkung der Paketlebenszeit wird einerseits ein vollständiges Fluten des Netzes vermieden und somit die Netzbelastung reduziert. Andererseits können dann zur Bestimmung von Gruppenmitgliedern jedoch mehrere Suchiterationen erforderlich sein: Bleibt eine Suchiteration erfolglos, d.h. bleibt sie bis zu einem bestimmten Zeitpunkt nach Aussendung der Anfrage unbeantwortet, wird davon ausgegangen, dass die lokale Suche auf ein zu kleines Gebiet beschränkt und die Lebenszeit des Anfragepakets zu klein gewählt wurde. Zur Vergrößerung des Suchbereichs wird demzufolge die Paketlebenszeit mit einem bestimmten Faktor multipliziert und die Anfrage erneut geflutet. Durch die sich so ergebende exponentielle Vergrößerung des Suchbereichs kann gewährleistet werden, dass Gruppenmitglieder, sofern sie vorhanden sind, mit wenigen Suchiterationen lokalisiert werden können.

Für die eigentliche Durchführung des Flutens von Paketen innerhalb eines drahtlosen Ad-hoc-Netzes wurden diverse Verfahren entwickelt. Unter Einsatz verschiedener Mechanismen erzielen diese unterschiedliche Effizienz. Da sie jedoch nicht unmittelbar Gegenstand dieser Arbeit sind, sei an dieser Stelle auf entsprechende Literatur verwiesen [77, 116, 117].

Folgerung: *Um in einem drahtlosen Ad-hoc-Netz einsetzbar zu sein, ist es wichtig, dass ein Overlay-Netz die Lokalisierung bestehender Gruppenmitglieder nicht über RVPs sondern mittels ERS gestaltet. Als Anforderung soll hier demnach die Eigenschaft der "Initialen Nachbarsuche" festgehalten werden.*

3.1.2 Die Eingliederung in die Overlay-Topologie

In Abhängigkeit des Aufbaus und der Eigenschaften einer Overlay-Topologie kann die Kontaktaufnahme mit einer Vielzahl von Gruppenmitgliedern erforderlich sein, um ein beitretendes Gruppenmitglied in die Overlay-Topologie einzubinden. Sowohl im Festnetz als auch in drahtlosen Ad-hoc-Netzen kann dieser Prozess eine erhebliche Zeitspanne in Anspruch nehmen [6], während welcher das Gruppenmitglied noch nicht an der eigentlichen Kommunikation teilnehmen kann [41]. Insbesondere in drahtlosen Ad-hoc-Netzen kann sich im Laufe der Eingliederung eines Gruppenmitglieds bereits eine deutliche Belastung des drahtlosen Mediums ergeben: Vor allem im Zusammenspiel mit reaktiven Routing-Protokollen auf Vermittlungsschicht kann sich die Kontaktaufnahme mit vielen Gruppenmitgliedern in Form von entsprechend zahlreich aufgebauten Routen und

dem damit verbundenen Fluten des Netzes auswirken. Bei rasch aufeinander folgenden Beitritten zahlreicher Gruppenmitglieder kann dies bereits zu einer Überlastung des drahtlosen Ad-hoc-Netzes führen, so dass sich der Beitrittsprozess als kritisch erweist und eines sorgfältigen Entwurfs bedarf.

Folgerung: *Das Ausmaß des bei der Eingliederung eines Gruppenmitglieds anfallenden Aufwands soll zur späteren Bewertung von Overlay-Netzen als Kriterium des "Eingliederungsaufwands" festgehalten werden.*

3.1.3 Die Anpassung der Overlay-Topologie

Die Berücksichtigung der Eigenschaften des zugrunde liegenden physikalischen Netzes zeigt sich sowohl im Festnetz als auch in drahtlosen Ad-hoc-Netzen als eine essentielle, von Overlay-Topologien zu erfüllende Anforderung, um über die Kommunikationsdauer hinweg deren Effizienz sicherzustellen. Werden Overlay-Verbindungen zwischen den Gruppenmitgliedern ohne Berücksichtigung der Eigenschaften der zugrunde liegenden Kommunikationswege aufgebaut führt dies, wie in Abbildung 3.1.a) dargestellt, zu einer ineffizienten Kommunikation: Dies liegt darin begründet, dass etwaige Kosten, ungeachtet ihrer Natur, unbeachtet bleiben und negative Auswirkungen, wie beispielsweise erhöhte Latenzen bei der Vermittlung von Daten, auf die Kommunikation haben können. Um dies zu vermeiden, ist eine Overlay-Topologie stattdessen derart zu konzipieren, dass Kommunikationswege zwischen Gruppenmitgliedern anhand einer bestimmten *Metrik* bewertet werden: In Abhängigkeit der dabei gemessenen Qualität ist der Kommunikationsweg in Form einer entsprechend aufgebauten Overlay-Verbindung in die Overlay-Topologie aufzunehmen. Dies ist in Abbildung 3.1.b) dargestellt, in welcher zur Verbindung aller Gruppenmitglieder die Kosten der Kommunikationswege minimiert wurden. Darüber hinaus ist insbesondere in drahtlosen Ad-hoc-Netzen darauf zu achten, dass die aufgrund der Metrik gemessene Qualität der Kommunikationswege bzw. Overlay-Verbindungen über die gesamte Kommunikationsdauer hinweg *überwacht* und die Overlay-Topologie gegebenenfalls entsprechend angepasst wird. In der Tat muss in drahtlosen Ad-hoc-Netzen von sich häufig ändernden Eigenschaften der Kommunikationswege ausgegangen werden, sei es aufgrund der Mobilität von Endgeräten oder stattfindenden Belastungen des drahtlosen Mediums. Derartige Änderungen sind mit Verschlechterungen aber auch Verbesserungen der Verbindungsqualität zwischen einzelnen Gruppenmitgliedern verbunden. Die Datenverteilung des Gruppenkommunikationsdienstes ist deshalb jeweils an die aktuelle Netzsituation anzupassen, um vor- und nachteilhafte Änderungen zu nutzen oder zu umgehen. Die besagten Anpassungen vollziehen sich dabei durch eine partielle Umstrukturierung der Overlay-Topologie, d.h. durch den Auf- und Abbau von Overlay-Verbindungen zwischen einzelnen Gruppenmitgliedern. Im Laufe derartiger Umstrukturierungen ist darauf zu achten, dass das Overlay-Netz zusammenhängend bleibt und nicht in einzelne Partitionen zerbricht.

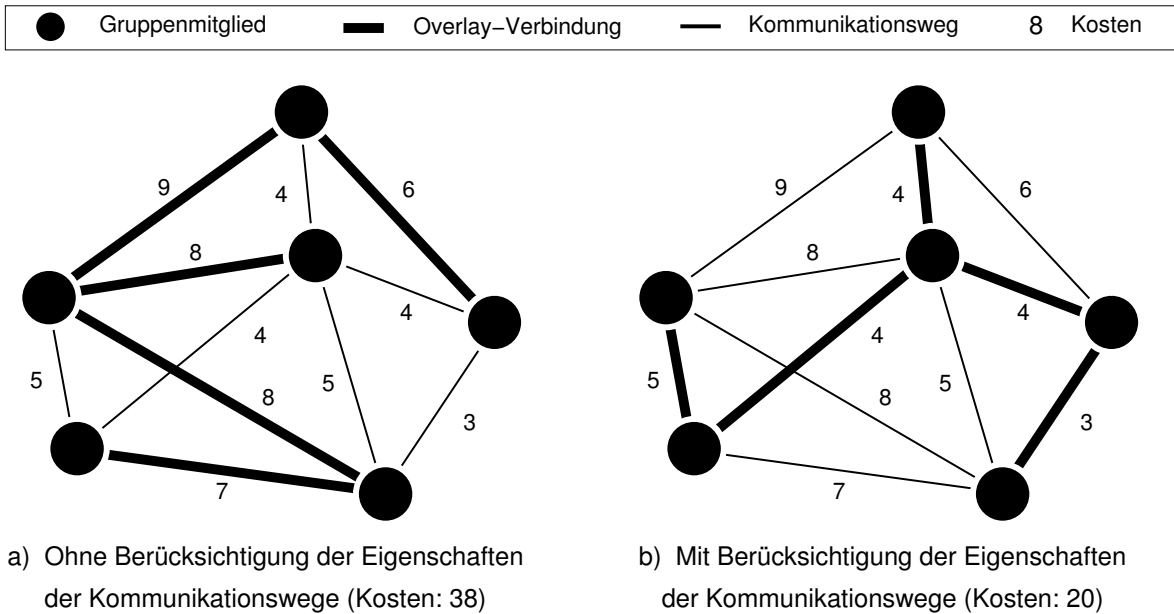


Abbildung 3.1 Aufbau einer Overlay-Topologie über Kommunikationswegen ohne und mit Berücksichtigung deren Eigenschaften.

Folgerung: Zum späteren Vergleich der Eigenschaften unterschiedlicher Overlay-Topologien wird an dieser Stelle die Existenz einer *“Optimierungsmetrik”* gefordert. Die letztendliche Natur der verwendeten Metrik und die Wahl der in die Bewertung einfließenden Größen erweist sich wiederum als kompliziert, so dass eine kurze Abhandlung gängiger Metriken in Abschnitt 3.2 gegeben wird.

3.1.4 Die Flexibilität der Overlay-Topologie

Um Änderungen innerhalb des zugrunde liegenden, physikalischen Netzes geeignet berücksichtigen zu können, ist von einer Overlay-Topologie ein entsprechendes Maß an Flexibilität zu fordern. Für die Flexibilität einer Overlay-Topologie lässt sich allerdings nur schwer ein quantitatives Bewertungsmaß finden, da sie eng an die Protokollmechanismen zum Aufbau der Overlay-Topologie gebunden ist. So hängt die Flexibilität einer Overlay-Topologie maßgeblich davon ab, inwieweit der Auf- und Abbau von Overlay-Verbindungen durch die Protokollmechanismen eingeschränkt wird. Können beispielsweise Overlay-Verbindungen zwischen *beliebigen* Gruppenmitgliedern aufgebaut werden, so ist die Overlay-Topologie flexibel, da keinerlei Einschränkungen bezüglich der Nutzung von Kommunikationswegen gemacht werden. Sieht die Overlay-Topologie hingegen die Einhaltung einer bestimmten Struktur vor, welche den Aufbau von Overlay-Verbindungen zu bestimmten Gruppenmitgliedern verhindert, wird sie unflexibel, so dass sie sich gegebenenfalls nicht an die Eigenschaften der zugrunde liegenden Kommunikationswege anpassen kann.

Folgerung: Die Fähigkeit einer Overlay-Topologie, sich an eine bestimmte Netzsituation anzupassen, wird zu späteren Bewertungszwecken an dieser Stelle als (qualitatives) Kriterium der “Flexibilität” festgehalten.

3.1.5 Die Konsistenz des Overlay-Routing

Im Fall des Verlusts von Kontrollpaketen, welche Information über eine gerade erfolgende oder bereits erfolgte Umstrukturierung der Overlay-Topologie beinhalten, entstehen zwischen den Gruppenmitgliedern unterschiedliche Sichten auf die Overlay-Topologie. Diese Situation hält so lange an, bis eines der zwischen Gruppenmitgliedern periodisch ausgetauschten Kontrollpakete mit aktueller Topologieinformation erfolgreich übertragen wird. In dieser Zeitspanne ist das Overlay-Routing *inkonsistent*, da es auf der den Gruppenmitgliedern jeweils vorliegenden Topologieinformation basiert. Demzufolge werden Datenpakete unter Umständen falsch weitergeleitet und erreichen möglicherweise nicht alle Gruppenmitglieder. Ein einfach gehaltenes Beispiel einer Routing-Inkonsistenz ist in Abbildung 3.2 dargestellt, in welcher Datenpakete einer Quelle Q an die Gruppenmitglieder A, B und C weitergeleitet werden sollen. Zur Verdeutlichung der Inkonsistenz sind für Q, A und B die jeweils geführten Listen von Nachfolgern, an welche Datenpakete weitergeleitet werden, dargestellt. Abbildung 3.2.a) zeigt eine, evtl. aufgrund einer Bewegung von Gruppenmitglied C entstandene, Ineffizienz in der Datenweiterleitung: In der Tat wäre eine Weiterleitung der Daten an C durch A günstiger als, wie abgebildet, durch B (Kosten 3:6). In Abbildung 3.2.b) wird daher eine Topologieanpassung durchgeführt, im Zuge welcher sich C als Nachfolger von A an- und als Nachfolger von B abmeldet. Während die Abmeldung erfolgreich vollzogen wird, ist die Übertragung des Kontrollpakets zur Anmeldung jedoch erfolglos. Als Folge dessen tritt C in Abbildung 3.2.c) in keiner Nachfolgerliste mehr auf und erhält demnach auch keine Datenpakete mehr: Das Routing ist inkonsistent. Erst durch das Eintreffen eines weiteren Kontrollpakets, Abbildung 3.2.d), bei A, anhand dessen sich C periodisch als Nachfolger bei A meldet, kann die Inkonsistenz aufgelöst und C wieder mit Datenpaketen, Abbildung 3.2.e), versorgt werden.

Das Problem im oben genannten Beispiel liegt in der parallel stattfindenden An- und Abmeldung von C als Nachfolger. Indem C seine Abmeldung bei B erst nach seiner erfolgreichen Anmeldung bei A vollzieht, könnte die Konsistenz des Routing *erhöht*, eine lückenlose Datenweiterleitung aber nicht *garantiert* werden. Der Grund hierfür ist in Abbildung 3.3 gezeigt. In Teilbild 3.3.a) erfolgt eine Datenweiterleitung von Q an dessen Nachfolger, A und B. Die Daten erreichen Gruppenmitglied A erfolgreich, welches aufgrund der leeren Nachfolgerliste keine Weiterleitungen vornimmt. Die Weiterleitung von Q an B wird hingegen durch eine temporäre Stausituation auf dem Kommunikationsweg zwischen Gruppenmitgliedern verzögert. Diese Verzögerung hält auch in Teilbild 3.3.b) an, wobei zeitgleich in den Schritten ①, ② und ③ jedoch erfolg-

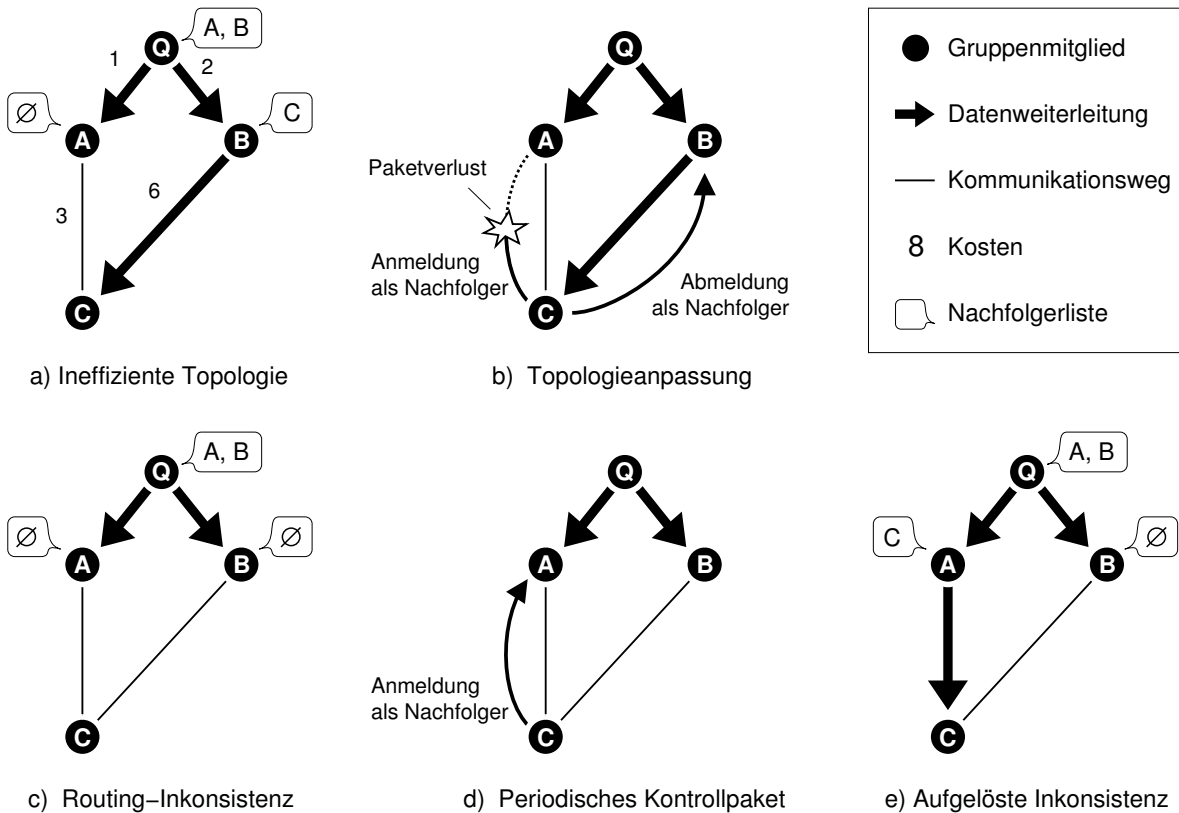


Abbildung 3.2 Entstehung und Auflösung einer Routing-Inkonsistenz als Folge eines Paketverlusts während einer Topologieanpassung.

reich eine konsistente Topologieanpassung durchgeführt wird. In Teilbild 3.3.c) ist die Stausituation aufgelöst, so dass das ursprünglich von Q versendete Datenpaket nun Gruppenmitglied B erreicht. Da B inzwischen jedoch über eine leere Nachfolgerliste verfügt, nimmt es keinerlei Weiterleitungen des Datenpakets vor: Als Konsequenz dessen erreicht das Datenpaket Gruppenmitglied C nicht.

Das Vermeiden solcher Paketlücken obliegt jedoch nicht dem Overlay-Netz, da dieses nur eine Vermittlungsfunktion erfüllt. Tatsächlich wäre zusätzliche Information erforderlich, wie etwa die Sequenznummern der vermittelten Datenpakete. Sequenznummern werden jedoch von einem Transportprotokoll vergeben, so dass dieses für die Auflösung solcher Paketlücken zuständig ist. Wichtig hierfür ist jedoch ein Zusammenspiel zwischen Overlay-Topologie und eingesetztem Transportprotokoll: In der Tat muss die Overlay-Topologie stattfindende Topologieanpassungen an das Transportprotokoll kommunizieren, damit dieses die geänderte Datenvermittlung berücksichtigen und z.B. die Quittierung der Daten einer bestimmten Quelle entsprechend anpassen kann.

Die oben gewählten Beispiele sind bewusst einfach gehalten, um die Problematik des konsistenten Routing zu verdeutlichen. In real eingesetzten Overlay-Topologien sind To-

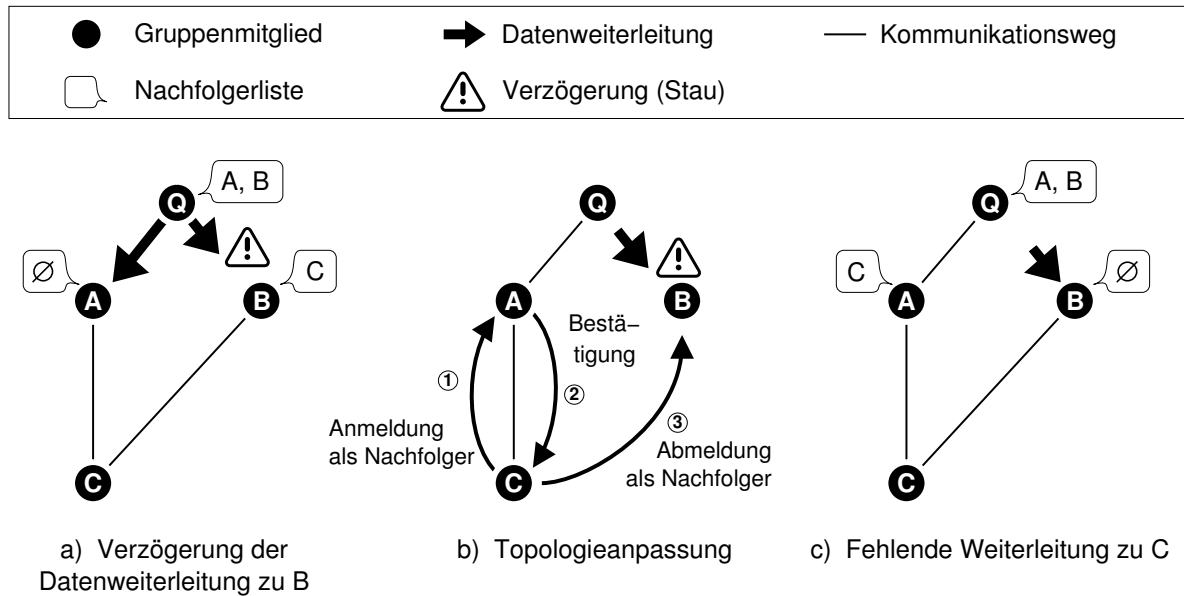


Abbildung 3.3 Fehlende Datenweiterleitung als Folge einer Topologieanpassung.

Topologieanpassungen oftmals kaum konsistent durchzuführen, da die Information über bestehende Verbindungen und Datenweiterleitungen über zahlreiche Gruppenmitglieder verteilt ist. Vor allem in drahtlosen Ad-hoc-Netzen ist der für eine konsistente Topologieanpassung erforderliche Informationsaustausch auf Anhieb kaum fehlerfrei zu bewerkstelligen: Vielmehr ist hier die wiederholte Übertragung einzelner Kontrollpakete erforderlich. Demzufolge nehmen in Abhängigkeit von der Struktur einer Overlay-Topologie Topologieanpassungen in drahtlosen Ad-hoc-Netzen einen Zeitraum in Anspruch, welcher sich über mehrere 10 Sekunden erstrecken kann. Erfolgen in einem solchen Zeitraum weitere, parallele Topologieanpassungen, kann es mitunter zu Inkonsistenzen kommen, welche sich unter Umständen nicht mehr beheben lassen und welche somit die Gruppenkommunikation in Mitleidenschaft ziehen oder sogar komplett verhindern.

Die Struktur einer Overlay-Topologie und die Mechanismen für ihren Aufbau sind ausschlaggebend für die Konsistenz ihres Overlay-Routing. Da sich besagte Mechanismen von Overlay zu Overlay jedoch grundlegend unterscheiden, ist es schwer, ein allgemeines Bewertungsmaß für die Konsistenz von Overlay-Routing-Protokollen zu finden. Im Hinblick auf die später erfolgende Bewertung von Overlay-Topologien wird an dieser Stelle daher festgehalten:

Folgerung: Zur Beurteilung der Konsistenz des Routing einer bestimmten Overlay-Topologie wird das Kriterium der *“Routing-Konsistenz”* verwendet. Dieses bewertet inwieweit Topologieanpassungen das Entstehen von Paketlücken in der Gruppenkommunikation zur Folge haben können. Als Ursache der Paketlücken werden allerdings lediglich die unterschiedlichen Sichten einzelner Gruppenmitglieder auf die Overlay-Topologie

verstanden und somit nicht jene Aspekte, welche aufgrund von Abbildung 3.3 im Zusammenhang mit stark unterschiedlichen Laufzeiten von Datenpaketen genannt wurden.

3.1.6 Die Anzahl unterhaltener Overlay-Verbindungen

Die Anzahl der innerhalb einer Overlay-Topologie unterhaltenen Overlay-Verbindungen ist gering zu halten. Dies ergibt sich aufgrund der Tatsache, dass zur Anpassung der Overlay-Topologie Overlay-Verbindungen kontinuierlich zu überwachen sind, und der Aufwand hierfür mit steigender Anzahl unterhaltener Overlay-Verbindungen wächst. Insbesondere im Zusammenhang mit reaktiven Routing-Protokollen ergibt sich durch den Verzicht auf Overlay-Verbindungen und der damit verbundenen Routen-Instandhaltung auf Vermittlungsschicht ein deutliches Verbesserungspotenzial. Weist eine Overlay-Topologie eine starke Vermaschung der Gruppenmitglieder untereinander auf, so schlagen sich Overlay-Verbindungen, welche für die Datenvermittlung nicht verwendet werden, in Form einer unnötigen Belastung des Ad-hoc-Netzes nieder: In der Tat wird die Qualität dieser Overlay-Verbindungen fortlaufend überwacht. Obgleich der Aufwand hierfür auf Anwendungsschicht sehr gering ist und beispielsweise dem Versenden eines *Ping*-Pakets gleichkommt, kann er auf Vermittlungsschicht das Auffinden und die Instandhaltung einer Route nach sich ziehen. Durch den Verzicht auf Overlay-Verbindungen, welche zur Datenvermittlung nicht genutzt werden, kann demnach eine Entlastung des Ad-hoc-Netzes erreicht werden. Bei Einsatz eines proaktiven Routing-Protokolls fallen derartige Entlastungen hingegen weniger drastisch aus, da Routen hier ohnehin instand gehalten werden und die Qualitätsbestimmung einer Overlay-Verbindung demnach keine weiteren Effekte auf Vermittlungsschicht nach sich zieht.

Folgerung: *Die Anzahl der von einer Overlay-Topologie aufgebauten Overlay-Verbindungen wird als Kriterium “#Overlay-Verbindungen” festgehalten und zur späteren Bewertung von Overlay-Topologien verwendet.*

3.1.7 Die induzierte Netzbelastung

Zur Beurteilung der Effizienz einer Overlay-Topologie kann ebenfalls das Volumen des in Abschnitt 2.2.2 eingeführten Kontrollflusses herangezogen werden, welcher üblicherweise durch den Austausch aktueller Topologie- und Routing-Informationen entsteht. Das Volumen des Kontrollflusses sollte möglichst gering gehalten werden, so dass die in drahtlosen Ad-hoc-Netzen zur Verfügung stehende Bandbreite bevorzugt zur Übertragung von Nutzdaten verwendet werden kann.

Um Aussagen bezüglich des von einer Overlay-Topologie induzierten Kontrollflusses treffen zu können, ist dessen alleinige Erfassung auf Anwendungsebene der Gruppenmitglieder jedoch nicht aussagekräftig. Für eine genaue Bewertung sind stattdessen die tatsächlichen Auswirkungen des Kontrollflusses auf das drahtlose Ad-hoc-Netz unter

Berücksichtigung Schichten-übergreifender Effekte in Betracht zu ziehen. Diese können, beispielsweise bei Verwendung eines reaktiven Routing-Protokolls, den Aufbau neuer Routen auf Vermittlungsschicht umfassen. Doch auch im Laufe des Medienzugriffs können sich, sowohl durch den Routen-Aufbau von Seiten des Routing-Protokolls als auch bei der Vermittlung des Overlay-Kontrollflusses, weitere Effekte ergeben: Besonders im Zusammenhang mit Unicast-Übertragungen sind hier die Sendewiederholungen zu nennen, welche von gängigen Medienzugriffsprotokollen (vgl. Abschnitt 2.1.1) durchgeführt werden, falls der Empfang der Dateneinheit von dem Empfängergerät unbestätigt bleibt.

Folgerung: *Um die Belastung des drahtlosen Ad-hoc-Netzes, welche aus der alleinigen Instandhaltung einer Overlay-Topologie hervorgeht, zu bewerten, wird das Kriterium der "Netzbelastung" verwendet. Um aussagekräftig zu sein und um die tatsächlich auf das Netz wirkende Last zu erfassen, ist die Netzbelastung auf der untersten Kommunikationsschicht, das heißt direkt bei den durchgeführten Medienzugriffen, zu messen.*

3.1.8 Die Nutzung des Semi-Broadcast-Mediums

Die Nutzung der Semi-Broadcast-Eigenschaft des geteilten Mediums gilt als ein wichtiger Aspekt bei der P2P-basierten Gruppenkommunikation in drahtlosen Ad-hoc-Netzen. Dies gilt insbesondere in Anwendungsszenarien, in welchen von Arealen mit einer hohen Gruppenmitgliedsdichte ausgegangen werden kann. Befinden sich zahlreiche Gruppenmitglieder in gegenseitiger Übertragungreichweite ist die Weiterleitung von Daten anhand der üblichen Unicast-Overlay-Verbindungen eines Overlay sehr ineffizient. Wie in Abbildung 3.4 verdeutlicht ist, resultiert dies aus der Tatsache, dass das drahtlose Medium in einem solchen Fall auf engem Raum entsprechend oft zur Übertragung eines einzelnen Datenpakets belegt wird. Die hierbei entstehende Redundanz ist mit einem deutlichen Einbruch des erzielbaren Durchsatzes verbunden. Um diesen Einbruch zu verhindern, kann und muss die Semi-Broadcast-Eigenschaft des geteilten Mediums gezielt genutzt werden. Dabei kann unterschieden werden, ob eine Overlay-Topologie die Semi-Broadcast-Eigenschaft zur Vermittlung der Nutzdaten oder auch des eigens implizierten Kontrollverkehrs einsetzt.

Folgerung: *Die Tatsache, inwieweit eine Overlay-Topologie die Semi-Broadcast-Eigenschaft des geteilten Mediums einsetzt, wird anhand des Kriteriums "Semi-Broadcast-Nutzung" beurteilt. Aus später vermittelten Gründen bezieht sich das Kriterium lediglich auf die Nutzung der Eigenschaft zur Vermittlung des Kontrollverkehrs der Overlay-Topologie und nicht zur Vermittlung des Datenverkehrs.*

3.1.9 Implizites und explizites Routing

Im Hinblick auf die Vermittlung von Nutzdaten innerhalb der Overlay-Topologie lässt sich das vorgenommene Overlay-Routing nach einem impliziten oder expliziten Charakter unterscheiden. Explizites Overlay-Routing basiert auf zwischen Gruppenmitgliedern

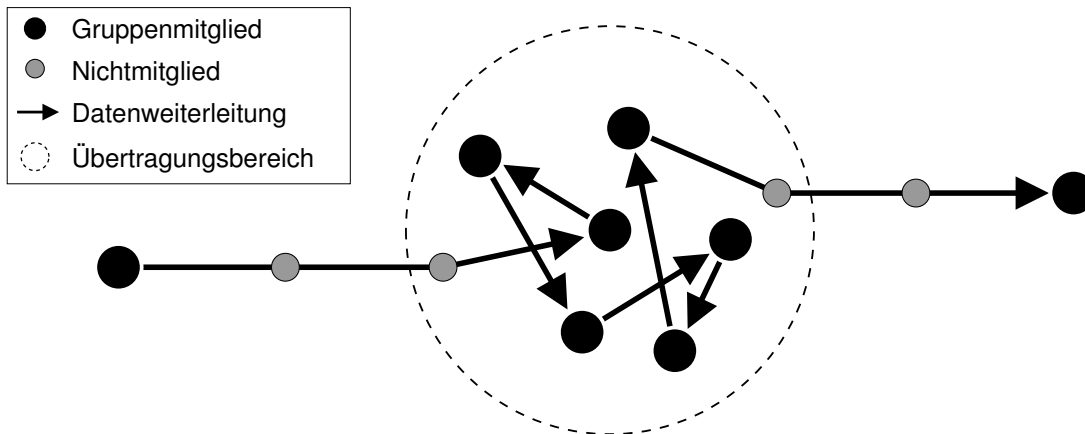


Abbildung 3.4 Redundante Medienbelegung aufgrund mangelnder Nutzung der Semi-Broadcast-Eigenschaft des geteilten Mediums.

ausgetauschter Topologie- und Routing-Information, wie beispielsweise Link-States. Bei implizitem Routing hingegen ist die Wegewahl direkt durch die Overlay-Topologie vorgegeben. Implizites Routing bietet gegenüber explizitem Routing demnach den Vorteil, keinen zusätzlichen Informationsaustausch zwischen den Gruppenmitgliedern zu erfordern. Zu bemerken sei an dieser Stelle allerdings, dass implizites Routing eine gewisse Strukturierung der Overlay-Topologie erfordert: Diese Strukturierung kann sich wiederum in Form von Einbußen im Zusammenhang mit anderen Bewertungskriterien auswirken, wie etwa der Flexibilität einer Overlay-Topologie oder der Routing-Konsistenz.

Folgerung: Die Tatsache, ob eine Overlay-Topologie auf explizitem oder implizitem Routing basiert, wird zu späteren Vergleichszwecken als Kriterium der "Routing-Information" festgehalten.

3.1.10 Die Stabilität der Multicast-Bäume

Strukturänderungen innerhalb einer Overlay-Topologie haben direkte Auswirkungen auf die Weiterleitung von Datenpaketen. Wird beispielsweise eine Overlay-Verbindung, welche zur Weiterleitung von Daten verwendet wird, aufgrund einer schlechten Bewertung abgebaut, so wird damit auch eine Änderung der Wegewahl für diejenigen Datenpakete *erzwungen*, welche bislang über die abgebaute Verbindung geleitet wurden. Mit anderen Worten ändert sich damit der mit einer Multicast-Quelle assoziierte Multicast-Baum, falls in diesem die abgebaute Overlay-Verbindung enthalten war.

Änderungen von Multicast-Bäumen sind jedoch nicht immer durch Strukturänderungen innerhalb der Overlay-Topologie bedingt: So können bereits sich verschlechternde Overlay-Verbindungen aus Multicast-Bäumen ausgeschlossen werden, ohne auch gleich in der Overlay-Topologie abgebaut zu werden. Während eine Berücksichtigung der Qualität

von Overlay-Verbindungen durchaus wichtig und zur Anpassung der Wegewahl heranzuziehen ist, kann es in Abhängigkeit des eingesetzten Verfahrens zur Wegewahl jedoch mitunter zu sehr häufigen Änderungen innerhalb eines Multicast-Baums kommen. Ein solches Verhalten ist wiederum unerwünscht, da ein sich fortlaufend ändernder Multicast-Baum einen erhöhten Aufwand bei der Quittierung von Datenpaketen verursacht.

Insbesondere im Zusammenhang mit einer zuverlässigen Datenübertragung ist also darauf zu achten, dass Multicast-Bäume auf einer gleich bleibenden Overlay-Topologie eine gewisse *Stabilität* aufweisen und nicht *kontinuierlichen* Änderungen unterliegen.

Folgerung: *Um bei einer Overlay-Topologie die Stabilität der darüber ausgebildeten Multicast-Bäume und die Häufigkeit von Änderungen bewerten zu können, wird das Kriterium der "Verteilbaumstabilität" festgehalten. Dieses umfasst jedoch nur das Ausmaß der Verteilbaumänderungen über einer ansonsten gleichbleibenden Overlay-Topologie: Es schließt somit Änderungen aus, welche durch den Abbau einer Overlay-Verbindung direkt erzwungen werden.*

3.1.11 Die Ausgliederung aus der Overlay-Topologie

Wünscht ein Gruppenmitglied seine Gruppenmitgliedschaft zu beenden, gliedert es sich aus der Overlay-Topologie aus. In Abhängigkeit der Struktur einer Overlay-Topologie ist dafür ein erheblicher Aufwand erforderlich: So muss unter Umständen Kontakt zu einer Vielzahl von Gruppenmitgliedern aufgenommen werden, um diese über die erfolgende Ausgliederung zu informieren. In der Regel übersteigt der Aufwand, welcher für die Ausgliederung eines Gruppenmitglieds aus der Overlay-Topologie anfällt, den Aufwand zur Eingliederung eines Gruppenmitglieds ebenso wie den Aufwand, welcher zur Durchführung von Strukturänderungen innerhalb der Overlay-Topologie benötigt wird. Im Internet werden daher sowohl strukturierte Overlay-Topologien, wie sie bei verteilten Hash-Tabellen zum Einsatz kommen, als auch P2P-basierte Gruppenkommunikationsprotokolle oftmals auf ihr Verhalten in Abhängigkeit einer Abwanderungsrate der Gruppenmitglieder (engl. "*churn rate*") untersucht [26, 69]. In drahtlosen Ad-hoc-Netzen gestalten sich derartige Untersuchungen als überaus schwierig, da hier die Problematik besteht, Ausgliederungen aus einer Overlay-Topologie konsistent durchzuführen. Aufgrund der hohen Fehleranfälligkeit der Kommunikation kann sich analog zur Problematik des konsistenten Routing die Ausgliederung eines Gruppenmitglieds über mehrere 10 Sekunden hinweg ziehen. Je nach Overlay-Topologie kann die Datenweiterleitung in dieser Zeitspanne inkonsistent und die Gruppenkommunikation gestört sein.

Im Zusammenhang mit der Ausgliederung von Gruppenmitgliedern ist zu bemerken, dass mit diesem Prozess die Anzahl der innerhalb einer Overlay-Topologie aufgebauten Overlay-Verbindungen in der Regel abnimmt. Weist eine Overlay-Topologie nur eine geringe Vermaschung der Gruppenmitglieder auf, kann es deshalb mitunter zu einer

Partitionierung der Overlay-Topologie kommen. Ein derartiges Szenario ist beispielhaft in Abbildung 3.5 dargestellt: Hier fungiert ein Gruppenmitglied A als einzige Verbindung zwischen zwei Mengen von Gruppenmitgliedern M_1 und M_2 . Nachdem A in Abbildung 3.5.a) die Verbindungen zu seinen im Overlay benachbarten Gruppenmitgliedern trennt, zeigt sich in Abbildung 3.5.b) die entstandene Partitionierung der Overlay-Topologie. Um die Gruppenkommunikation aufrecht zu erhalten, ist es entscheidend, solche Partitionierungen gezielt zu vermeiden oder nach Abschluss der Ausgliederung entsprechend zu beheben.

Folgerung: *Der Aufwand, welcher im Zuge der Ausgliederung eines Gruppenmitglieds aus der Overlay-Topologie anfällt, wird zu späteren Vergleichszwecken als Kriterium des "Ausgliederungsaufwands" festgehalten.*

3.1.12 Die Reaktivität gegenüber Ausfällen

Besonders in dynamischen und von spontanem Verhalten geprägten Kommunikationsnetzen wie drahtlosen Ad-hoc-Netzen können einzelne Geräte zeitweise nicht erreichbar sein oder sogar gänzlich ausfallen. Handelt es sich bei einem ausfallenden Gerät um ein Gruppenmitglied, ist dieses nicht mehr in der Lage, seine Aufgaben innerhalb der Overlay-Topologie zu erfüllen. Nimmt das Gruppenmitglied an der Duplizierung und Weiterleitung von Datenpaketen teil, ist nach dem Ausfall des Gruppenmitglieds die Gruppenkommunikation gestört oder sogar gänzlich unterbrochen. Um eine derartige Störung zu beheben, ist das ausgefallene Gruppenmitglied zu identifizieren und aus der Overlay-Topologie auszugliedern.

Während die Problematik der Ausgliederung eines Gruppenmitglieds aus der Overlay-Topologie anfangs einem Gruppenaustritt ähnlich scheint, ist der damit verbundene Prozess weitaus komplizierter und zeitaufwendiger. In der Tat ist zunächst der Ausfall des Gruppenmitglieds festzustellen. In gängigen Overlay-Topologien erfolgt diese Detektion aufgrund des wiederholten Ausbleibens der Kontrollpakete, welche periodisch zwischen im Overlay benachbarten Gruppenmitgliedern ausgetauscht werden. In Abhängigkeit von deren Periodizität einerseits⁶ und der Anzahl auszubleibender Kontrollpakete⁷ beansprucht die Ausfallserkennung eine Zeitdauer, welche durchaus im Minutenbereich liegen kann. In diesem Zeitraum ist die Gruppenkommunikation evtl. bereits gestört oder unterbrochen.

Auch die tatsächliche Ausgliederung des Gruppenmitglieds aus der Overlay-Topologie selbst ist komplizierter als bei einem beabsichtigten Gruppenaustritt. Während bei

⁶Die Periodizität der zwischen Gruppenmitgliedern ausgetauschten Kontrollpakete liegt bei gängigen Overlay-Topologien üblicherweise zwischen 10s und 20s.

⁷Die Anzahl aufeinander folgender Kontrollpakete, welche ausbleiben müssen, um ein Gruppenmitglied als ausgefallen zu deklarieren, liegt üblicherweise zwischen 4 und 6.

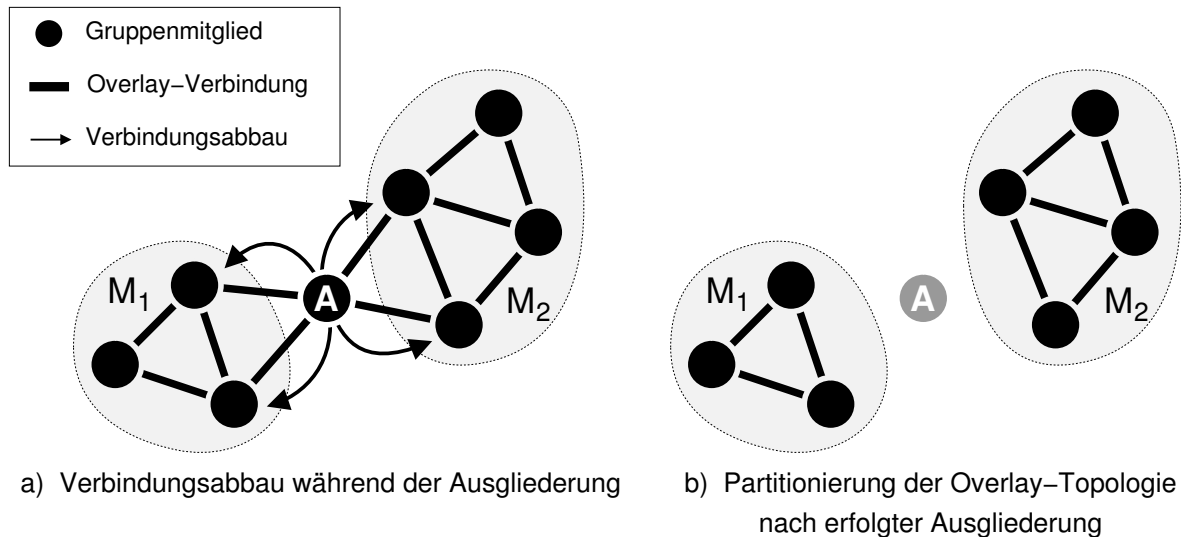


Abbildung 3.5 Partitionierung einer Overlay-Topologie als Folge der Ausgliederung eines Gruppenmitglieds aus dem Overlay.

diesem das auszugliedernde Mitglied noch aktiv an der eventuell erforderlichen Umstrukturierung der Overlay-Topologie behilflich sein und Information über benachbarte Gruppenmitglieder bereitstellen kann, ist dies bei einem Ausfall von den verbleibenden Gruppenmitgliedern selbstständig durchzuführen. In Abhängigkeit des Aufbaus einer Overlay-Topologie kann hierfür ein aufwendiger Austausch von Kontrollpaketen erforderlich sein: So kann beispielsweise bei einer Overlay-Topologie mit Cluster-Struktur der Ausfall eines Cluster-Repräsentanten eine Neuwahl des Repräsentanten erfordern.

Für Protokolle zur P2P-basierten Gruppenkommunikation im Internet wurden Verfahren entwickelt, um die Auswirkungen von Gruppenmitgliedsausfällen im Hinblick auf die Weiterleitung von Datenpaketen zu reduzieren [7, 25–27]. Diese basieren auf zusätzlichen Overlay-Verbindungen, welche zwischen zufälligen Gruppenmitgliedern aufgebaut und über welche Datenpakete mit einer bestimmten Wahrscheinlichkeit zusätzlich zur Weiterleitung des Overlay-Routing versendet werden. Indem also ein Anteil der Datenpakete an verschiedenen Stellen im Overlay eingeschleust und dort weitergeleitet wird, entsteht ein gewisses Maß an Redundanz. Für drahtlose Ad-hoc-Netze erscheinen diese Ansätze weniger geeignet: Einerseits gehen die zusätzlich aufgebauten Overlay-Verbindungen mit einer Erhöhung des Verwaltungsaufwands einher, und andererseits berücksichtigen sie aufgrund ihres zufälligen Charakters nicht die Eigenschaften der Kommunikationswege zwischen den entsprechenden Gruppenmitgliedern.

Folgerung: Der Aufwand, welcher beim Ausfall eines Gruppenmitglieds zur Reparatur der Overlay-Topologie erforderlich ist, wird zu späteren Vergleichszwecken von Overlay-Topologien als Kriterium der „Ausfallsbehandlung“ festgehalten.

3.2 Metriken zur Verbindungsbewertung

Um die Eigenschaften der Kommunikationswege zwischen Gruppenmitgliedern berücksichtigen und Änderungen dieser Eigenschaften bemerken zu können, ist eine fortlaufende Überwachung der Kommunikationswege zwischen den Gruppenmitgliedern erforderlich. Hierfür bedarf es einer Metrik, anhand welcher eine Bewertung vorgenommen werden kann. Zwei Metriken, die sich in drahtlosen Ad-hoc-Netzen anbieten, sind die zwischen zwei Gruppenmitgliedern auftretende Paketumlaufzeit (*Round-Trip Time, RTT*) einerseits und der zwischen ihnen herrschende Hop-Abstand (*Hop-Count*) andererseits. Der Hop-Abstand gleicht der Anzahl der Übertragungsabschnitte, welche erforderlich sind, um eine Kommunikation zwischen beiden Gruppenmitgliedern zu ermöglichen. Die Paketumlaufzeit hingegen berechnet sich aus der Differenz $\Delta t = t_2 - t_1$ zweier Zeitstempel. Diese ergeben sich aus dem Versenden (t_1) eines so genannten Ping-Pakets durch das eine Gruppenmitglied und dem Eintreffen (t_2) eines zugehörigen Pong-Pakets, welches von dem anderen Gruppenmitglied sofort nach Erhalt des Ping-Pakets versendet wird. Um die zielgerichtete Optimierung einer Overlay-Topologie zu ermöglichen, muss eine Metrik zwei Anforderungen erfüllen, welche in den folgenden Abschnitten kurz im Zusammenhang zu den genannten Metriken diskutiert werden.

3.2.1 Die eindeutige Wiedergabe des Netzzustands

Um eine Aussage über den aktuellen Netzzustand treffen zu können, müssen die gemessenen Größen, welche die Grundlage der Bewertungsmetrik bilden, bei einem gleich bleibenden Netzzustand möglichst gleich bleibende Werte annehmen, d.h. eine möglichst geringe Varianz besitzen. Ist dies nicht der Fall, kann eine Metrik Oszillationen der Overlay-Topologie und die damit verbundenen, in Abschnitt 3.1.5 beschriebenen Schwierigkeiten verursachen.

In diesem Zusammenhang erweist sich die Verwendung von Paketumlaufzeiten als nachteilhaft: In der Tat kann gemäß Abschnitt 2.1.1.2 vor allem im Zusammenhang mit einer Netzbelastung für den Medienzugriff in jedem Übertragungsabschnitt eine zufällige Back-off-Zeit gemäß Formel 2.1 gewählt werden, welche die Weiterleitung der Ping/Pong-Pakete entsprechend verzögert. Im Falle einer Kollision beim Medienzugriff erfolgt eine exponentielle Vergrößerung des Back-off-Fensters, so dass nicht nur die erwarteten Verzögerungen sondern auch deren Varianz mit der Anzahl der Übertragungsabschnitte sowie der Anzahl der auf das Medium zugreifenden Endgeräte und der Netzlast steigen. Genaue Werte hierfür wurden bereits in unterschiedlichen Arbeiten bestimmt, sowohl theoretisch [8, 24, 45] als auch simulativ [28], so dass sie in dieser Arbeit daher nicht näher erläutert werden sollen. Um die deutlichen Schwankungen der Paketumlaufzeiten zu mindern, sind starke Glättungen der Messwerte erforderlich, welche sich jedoch wiederum in einer erhöhten Trägheit der Topologieanpassung niederschla-

gen. Ein weiterer Nachteil einer Metrik auf Basis von Paketumlaufzeiten ergibt sich im Zusammenhang mit reaktiven Routing-Protokollen auf Vermittlungsschicht. Ist für den zu bewertenden Kommunikationsweg keine Route bekannt, so muss diese zunächst bestimmt werden. Die hierfür benötigte Dauer verfälscht die Messung der Paketumlaufzeit vor allem gegenüber Kommunikationswegen, für welche bereits eine Route vorliegt.

3.2.2 Die implizierte Belastung des Netzes

Die Netzbelastung, welche durch die Feststellung bzw. die Messung der in die Bewertungsmetrik einfließenden Größen bedingt wird, muss möglichst gering gehalten werden. Auch hier erweist sich die Verwendung von Paketumlaufzeiten als nachteilhaft, da zu deren Erfassung das explizite Versenden von Ping/Pong-Paketen erforderlich ist. Je nach auf Vermittlungsschicht verwendetem Routing-Protokoll existiert beim Versenden dieser Pakete nicht notwendigerweise eine Route zwischen Sender und Empfänger, so dass diese erst etabliert und hierfür der entsprechende Aufwand betrieben werden muss.

In Abhängigkeit des auf Vermittlungsschicht eingesetzten Routing-Protokolls kann im Gegensatz zur Paketumlaufzeit die Anzahl der Übertragungsabschnitte zwischen zwei Endgeräten durch Schichten-übergreifende Kommunikation oftmals direkt aus der Routing-Tabelle auf Vermittlungsschicht ausgelesen werden. Für den Fall, dass auf Vermittlungsschicht zu dem Zeitpunkt der Messung eine Route zwischen den Gruppenmitgliedern bekannt ist, ist somit kein explizites Versenden von Kontrollpaketen erforderlich. Andernfalls ist eine Route zwischen den Gruppenmitgliedern zunächst aufzubauen, so dass anschließend das Auslesen der Übertragungsabschnittszahl erfolgen kann.

Folgerung: *Angesichts der oben beschriebenen Eigenschaften der Metriken "Paketumlaufzeit" und "Hop-Abstand" erweist sich der Hop-Abstand als aussagekräftigeres Abstandsmaß und ist daher der Paketumlaufzeit vorzuziehen. Aufgrund welcher Metrik eine Overlay-Topologie letztendlich Overlay-Verbindungen bewertet, soll zu späteren Vergleichszwecken als Protokolleigenschaft der "Optimierungsmetrik" bezeichnet werden.*

3.3 Zuverlässigkeit und Staukontrolle

Sowohl im Internet als auch in drahtlosen Ad-hoc-Netzen ist das Erreichen einer zuverlässigen Gruppenkommunikation eine ausführlich studierte Thematik. Die Mehrheit der Untersuchungen konzentriert sich dabei auf Zuverlässigkeitsmechanismen, welche auf einem unzuverlässigen Multicast-Routing-Protokoll aufsetzen. Die Umsetzung solcher Mechanismen erweist sich aufgrund der in Abschnitt 2.2.3 dargestellten Problematik jedoch als äußerst komplex.

Im Kontext einer P2P-basierten Gruppenkommunikation erweisen sich Zuverlässigkeitssaspekte als einfacher beherrschbar, da hier die Datenverteilung von Gruppenmitglied

zu Gruppenmitglied entlang entsprechender Overlay-Verbindung stattfindet. Die Zuverlässigkeit der Gruppenkommunikation kann deshalb erreicht werden, indem die Zuverlässigkeit innerhalb von Overlay-Verbindungen anhand entsprechender Transportprotokolle hergestellt wird.

Transportprotokolle sind in der Transportschicht von Endgeräten untergebracht. Da sie hier in der Regel eng mit dem Betriebssystem verwoben sind, sind neue Konzepte nur schwerlich einzubringen. Analog zur Duplizierung und Weiterleitung von Datenpaketen bieten P2P-basierte Gruppenkommunikationsprotokolle jedoch die Möglichkeit, Transportprotokolle und die dazugehörigen Zuverlässigkeitsmechanismen direkt in das eigentliche Gruppenkommunikationsprotokoll zu integrieren.

In den folgenden Abschnitten soll zunächst kurz auf das Transportprotokoll TCP sowie die Schwierigkeiten eingegangen werden, welche mit diesem Protokoll in drahtlosen Ad-hoc-Netzen auftreten. Anschließend soll die Integration Anwendungs-abhängiger Zuverlässigkeits- und Staukontrollmechanismen in ein P2P-basiertes Gruppenkommunikationsprotokoll näher betrachtet werden.

3.3.1 TCP in drahtlosen Ad-hoc-Netzen

Das *Transmission Control Protocol* (TCP [78, 96, 97]) gilt im Internet als Standardprotokoll zur zuverlässigen Ende-zu-Ende-Kommunikation. Während es Protokollmechanismen zur Durchführung von Übertragungswiederholungen beinhaltet, integriert es ebenfalls Konzepte zur Durchführung einer Staukontrolle auf Ende-zu-Ende-Basis. Diese beruht auf einer *Fenster-technik*, bei welcher die *Fensterkapazität* die Anzahl der Datenpakete widerspiegelt, welche ein Sendergerät versenden darf, ohne auf eine Bestätigung der Datenpakete durch das Empfängergerät warten zu müssen⁸. Während die Fensterkapazität anfangs gering ist, versendet ein Sendergerät Datenpakete zunächst nur mit großem zeitlichen Abstand. Diese Phase ist als die TCP *Slow-Start*-Phase bekannt. Das Empfängergerät quittiert Datenpakete kumulativ, wobei empfangene Quittungen aus Sicht des Sendergeräts als eine nicht ausgeschöpfte, zur Verfügung stehende Bandbreite interpretiert werden. Aus diesem Grund steigert das Sendergerät bei eintreffenden Quittungen die Kapazität seines Fensters exponentiell und versendet Datenpakete dementsprechend zügiger. Das Ausbleiben einer erwarteten Quittung interpretiert TCP wiederum als Überlastsituation, im Zuge welcher Pakete (wie etwa die vermisste Quittung oder die entsprechenden Datenpakete) verworfen wurden. Während inzwischen unterschiedliche Protokollvarianten existieren, setzt das ursprüngliche TCP, um dieser Überlast entgegenzuwirken, seine Fensterkapazität zurück und geht erneut in die Slow-Start-Phase über.

⁸Um die Überlastung des Empfängers zu vermeiden, sieht TCP ein zusätzliches Flusskontrollfenster vor. Die Schwachstellen von TCP in drahtlosen Ad-hoc-Netzen sind jedoch unabhängig von diesem Fenster, so dass es an dieser Stelle nicht näher betrachtet wird.

Zahlreiche Untersuchungen belegen, dass TCP aufgrund dieser Staukontrollmechanismen nur sehr bedingt in drahtlosen Ad-hoc-Netzen eingesetzt werden kann [4, 49, 52]. Dies geht auf die Tatsache zurück, dass Quittungen oftmals auch ohne tatsächlich vorhandene Überlastsituationen ausbleiben können: Vielmehr werden Quittungen aufgrund von Link-Brüchen verworfen, so dass TCP daraufhin in die Slow-Start-Phase übergeht. Nach Umgehung des gebrochenen Links, d.h. nach Reparatur der Route zwischen den Kommunikationspartnern, steht oftmals jedoch die bisherige Bandbreite erneut zur Verfügung, so dass sich die von TCP zurückgesetzte Fensterkapazität negativ auf den erzielbaren Durchsatz auswirkt.

Für drahtlose Ad-hoc-Netze wurden daher zahlreiche Modifikationen von TCP vorgeschlagen, wobei bislang jedoch keine einen nennenswerten Durchbruch erlangt hat. Wie in Abschnitt 3.3.3 verdeutlicht wird, liegt dies darin begründet, dass eine effiziente Staukontrolle in drahtlosen Ad-hoc-Netzen nur sehr bedingt auf Basis einer Ende-zu-Ende-Kommunikation bereitgestellt werden kann.

3.3.2 Zuverlässigkeitsmechanismen

Anwendungen unterscheiden sich in der von ihnen geforderten Zuverlässigkeit der Datenübertragung, so dass bereits eine Vielzahl von Konzepten entstanden sind, um eine zuverlässige Gruppenkommunikation zu ermöglichen [19, 81, 82, 102–104]. Um eine möglichst große Vielfalt an Anwendungen in geeigneter Weise unterstützen zu können, sind Zuverlässigkeitsmechanismen daher nicht fest sondern möglichst flexibel in einen P2P-basierten Gruppenkommunikationsdienst zu integrieren. In der Tat sind die von der Overlay-Topologie getroffenen Entscheidungen bezüglich der Duplizierung und Weiterleitung von Datenpaketen als gänzlich unabhängig von der Art und Weise zu betrachten, aufgrund welcher innerhalb einer Overlay-Verbindung entstandene Paketlücken behoben werden. Wünschenswert ist hier eine strikte Trennung von Overlay-Topologie und Zuverlässigkeitsmechanismen, so dass diese lediglich anhand fest definierter Schnittstellen miteinander interagieren. Auf diese Weise können einerseits Zuverlässigkeitsmechanismen über unterschiedlichen Overlay-Topologien betrieben werden. Andererseits kann eine Overlay-Topologie im Kontext unterschiedlicher Anwendungen mit jeweils spezifischen Zuverlässigkeitsmechanismen eingesetzt werden.

Darüber hinaus wird an dieser Stelle gefordert, dass die für eine Anwendung konzipierten Zuverlässigkeitsmechanismen auch nur auf die von der *Anwendung* tatsächlich ausgesendeten *Datenpakete* angewendet werden. Jegliche Form von Kontrollpaketen, wie sie etwa von den Instandhaltungsmechanismen der Overlay-Topologie versendet werden, sind getrennt davon zu betrachten: Kontrollpakete sind somit nicht in die Anwendungsspezifischen Zuverlässigkeitsmechanismen mit einzubeziehen, auch wenn sie über die gleichen Overlay-Verbindungen gesendet werden. Diese Aussage beruht auf der Annahme, dass das Versenden von Kontrollpaketen bestimmten (zeitlichen) Mustern folgt,

welche deutlich von dem Datenstrom der Anwendung abweichen können. Zuverlässigkeitsmechanismen, welche stark auf spezifische Muster eines empfangenen Datenstroms achten (z.B. die Paketzweischenankunftszeit) können bei Einbezug von Anwendungsfremden Kontrollpaketen mit ungewolltem Verhalten reagieren. Dies hat insbesondere die Konsequenz, dass Overlay-Topologien, welche eine zuverlässige Übertragung ihrer Kontrollpakete wünschen, selbst für die Sicherung dieser zuständig sind.

3.3.3 Staukontrolle

Die Konzeption einer Ende-zu-Ende-basierten Staukontrolle gilt in drahtlosen Ad-hoc-Netzen als kaum zu bewältigende Herausforderung. Dies geht auf die Tatsache zurück, dass den Endgeräten keine ausreichende Sicht auf die innerhalb des Netzes stattfindenden Vorgänge vorliegt [70]. Ereignisse, welche *innerhalb* einer Ende-zu-Ende-Verbindung mit mehreren Übertragungsabschnitten auftreten, bleiben unbemerkt und daher unberücksichtigt. Solche Ereignisse umfassen beispielsweise das Ansteigen des Füllstands der Paketwarteschlange in weiterleitenden Endgeräten aufgrund anhaltender Kollisionen beim Zugriff auf das geteilte Medium. Da bei einer vollständig gefüllten Paketwarteschlange weiterhin eintreffende Pakete verworfen werden, sollten Sendergeräte, welche die betroffenen Endgeräte in die Weiterleitung ihrer Datenpakete einbeziehen, ihre Sendegeschwindigkeit drosseln.

An dieser Stelle wird deutlich, dass in drahtlosen Ad-hoc-Netzen eine Staukontrolle nicht auf einer Ende-zu-Ende-Basis erfolgen *kann*: Tatsächlich ist das Wissen, welches etwa zur Drosselung von Senderaten erforderlich ist, einerseits innerhalb des Netzes verteilt. Andererseits wird es dort von unteren Protokollschichten, wie etwa der Vermittlungsschicht, verwaltet, deren Nutzung auf einer Ende-zu-Ende-Basis transparent erfolgt, so dass die darin verwaltete Information nicht zugänglich ist.

Neuartige Konzepte für Staukontrollen in drahtlosen Ad-hoc-Netzen beschäftigen sich daher damit, Protokollmechanismen zur Staukontrolle von der Transport- in die Vermittlungsschicht zu verlagern [35, 70, 91]. Während sich mit diesen Ansätzen teilweise auch eine Staukontrolle für Gruppenkommunikationsprotokolle realisieren lässt [92], übersteigen sie jedoch den Rahmen dieser Arbeit, so dass Staukontrollmechanismen hier nicht weiter betrachtet werden sollen.

3.4 Eigenschaften und Bewertung existierender P2P-Multicast-Protokolle

Nachdem in den vorangehenden Abschnitten eine Liste von Bewertungskriterien erarbeitet wurde, soll in diesem Abschnitt die Bewertung existierender P2P-basierter Gruppenkommunikationsprotokolle erfolgen. Die in Betracht kommenden und in dieser Arbeit zu

Vergleichszwecken herangezogenen Ansätze umfassen unterschiedliche Protokolle, welche für das Internet oder speziell für drahtlose Ad-hoc-Netze entwickelt wurden:

1. Das Protokoll *Narada* [38]: Narada ist eines der ersten, speziell zur Gruppenkommunikation auf Anwendungsschicht entwickelten P2P-Multicast-Protokolle. Es zielt auf den Einsatz im Festnetz. Da es Protokollmechanismen zur fortlaufenden Anpassung der aufgebauten Overlay-Topologie integriert, erscheint es allerdings auch für den Einsatz in drahtlosen Ad-hoc-Netzen interessant und wird daher in dieser Arbeit herangezogen.
2. Das *NICE*-Protokoll [6]: Ebenso wie Narada wurde auch das NICE-Protokoll zur P2P-basierten Gruppenkommunikation im Festnetz entwickelt. Da es ebenfalls Protokollmechanismen zur Topologieanpassung integriert und darüber hinaus eine Overlay-Topologie aufbaut, welche speziell im Hinblick auf die Skalierbarkeit gegenüber der Gruppenmitgliedszahl entworfen wurde, wird auch dieses Protokoll in dieser Arbeit näher betrachtet.
3. Das *PAST-DM*-Protokoll [51]: PAST-DM gilt als das erste Protokoll, welches speziell zur P2P-basierten Gruppenkommunikation in drahtlosen Ad-hoc-Netzen entwickelt wurde und daher in dieser Arbeit zu Vergleichszwecken herangezogen wird. Es wurde jedoch nur zur Unterstützung kleiner Gruppen entworfen, so dass im Laufe dieser Arbeit auch die Skalierbarkeitsaspekte des Protokolls näher betrachtet werden. Darüber hinaus wird bereits an dieser Stelle angemerkt, dass die Protokollbeschreibung in [51] mitunter sehr vage gehalten ist, so dass in dieser Arbeit an den jeweiligen Stellen Annahmen im Hinblick auf das Protokollverhalten getroffen werden müssen.
4. Das Protokoll *ALMA* [48]: Auch bei ALMA handelt es sich um ein für drahtlose Ad-hoc-Netze entworfenes Protokoll zur P2P-basierten Gruppenkommunikation. In seinem Entwurf berücksichtigt es die größten Schwächen des PAST-DM-Protokolls, wie etwa das hohe Volumen des Kontrollflusses. Da die mit ALMA erzielbaren Ergebnisse aus dessen Beschreibung und Eigenschaften abgeleitet werden können, soll dieses Protokoll in dieser Arbeit lediglich theoretisch diskutiert, später aber nicht näher evaluiert werden.

Zur übersichtlichen Bewertung der Protokolle sind die einzelnen Bewertungskriterien inhaltlich gruppiert. Die entstehende Gruppierung spiegelt sich sowohl in der weiteren Gliederung dieses Abschnitts wider als auch in Tabelle 3.1: So diskutiert Abschnitt 3.4.1 (Block ① in Tabelle 3.1) allgemeine Eigenschaften der jeweiligen Overlay-Topologien, Abschnitt 3.4.2 (Tabellenblock ②) die Mechanismen für den Gruppenbeitritt, Abschnitt

		Narada	NICE	PAST-DM	ALMA
	Topologie	Mesh	Cluster	Mesh	Baum
①	# Overlay-Verbindungen	⊕	⊖	○	⊕
	Flexibilität	⊕	⊖	⊕	○
	Optimierungsmetrik	RTT	RTT	Hops	RTT
	Netzbelastung	⊕	○	⊖⊖	⊕
②	Initiale Nachbarsuche	unspez.	RVP	ERS	unspez.
	Eingliederungsaufwand	○	⊖	⊕	⊕
③	Routing-Information	Explizit	Implizit	Explizit	Implizit
	Routing-Konsistenz	○	⊖	⊖	⊕
	Verteilbaum	Quelle	Quelle*	Quelle	Global
	Verteilbaumstabilität	○	⊕	⊖	○
④	Gruppenaustritt	⊕⊕	⊖	⊕	⊕
	Ausfälle	⊕	⊖⊖	unspez.	unspez.
⑤	Semi-Broadcast-Nutzung	⊖	⊖	○	⊖

Tabelle 3.1 Zusammenfassung der Bewertung existierender P2P-Multicast-Protokolle (*: siehe Text). Bewertungsmaß: ⊕⊕ = sehr gut, ⊕ = gut, ○ = befriedigend, ⊖ = ausreichend, ⊖⊖ = mangelhaft.

3.4.3 (Tabellenblock ③) die Vermittlung von Datenpaketen, Abschnitt 3.4.4 (Tabellenblock ④) die Ausgliederung von Gruppenmitgliedern aus den Overlay-Topologien und Abschnitt 3.4.5 (Tabellenblock ⑤) die Nutzung der Semi-Broadcast-Eigenschaft.

3.4.1 Aufbau und Anpassung der Overlay-Topologie

In diesem Abschnitt werden für die in dieser Arbeit betrachteten Protokolle die für den Aufbau und die Anpassung von Overlay-Topologien integrierten Mechanismen näher beschrieben und diskutiert.

– *Narada*: Das Protokoll Narada baut zwischen den Gruppenmitgliedern ein leichtgewichtiges und effizientes Mesh auf, welches beispielhaft in Abbildung 3.6 dargestellt ist. Innerhalb des Mesh können Overlay-Verbindungen prinzipiell zwischen beliebigen Gruppenmitgliedern aufgebaut werden, so dass sich die Overlay-Topologie als flexibel erweist und sich gut an die Eigenschaften von Kommunikationswegen anpassen lässt. Dynamische Anpassungen zur Laufzeit sind laut [38] entscheidend, um auf sich ändernde Gruppenmitgliedschaften zu reagieren und eine hohe Effizienz der Kommunikation zu gewährleisten. Der Begriff “Effizienz” ist dabei derart zu interpretieren, dass der Pfad zwischen zwei Gruppenmitgliedern entlang des aufgebauten Mesh eine Leistungsfähigkeit bieten soll, welche vergleichbar ist mit einer direkten Unicast-Verbindung zwischen den Gruppenmitgliedern. Während die tatsächliche Bewertung der Leistungsfähigkeit in Narada flexibel gehalten wird, werden in [38] Paketumlaufzeiten (RTTs) näher betrachtet und als Optimierungsmetrik verwendet. Ausgehend hiervon wird direkt ein

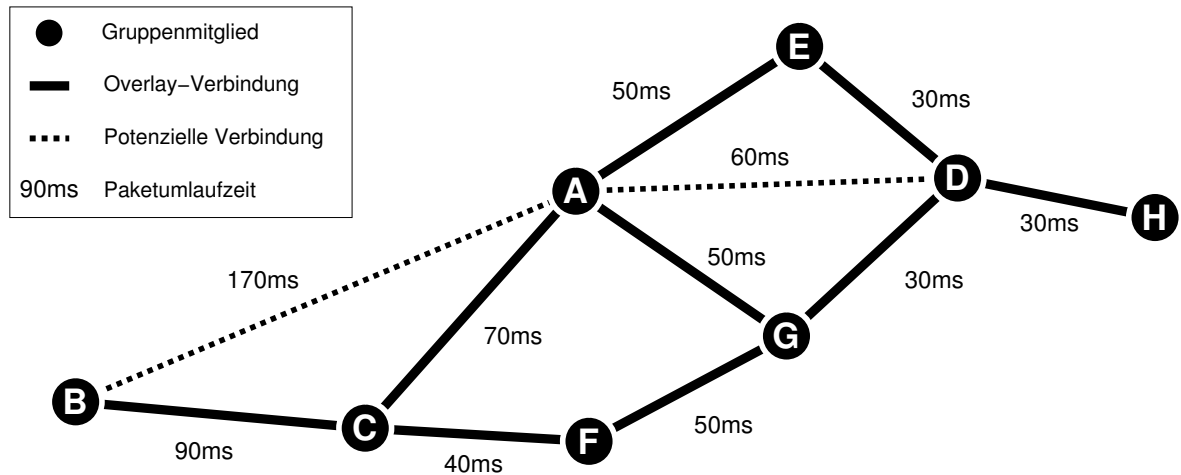


Abbildung 3.6 Das leichtgewichtige Mesh, wie es von Narada aufgebaut wird.

Kriterium zum Aufbau neuer oder zum Abbau bestehender Overlay-Verbindungen definiert, welches auf der Nützlichkeit der betrachteten Verbindung basiert: So wird eine Verbindung als nützlich gesehen, falls ihr Einbezug in (zumindest) einen Anteil α der existierenden Multicast-Verteilbäume die Latenz der jeweiligen Datenverteilung senkt. Analog wird die Verbindung als unnützlich gesehen, falls sie nicht in (zumindest) einem Anteil β aller Multicast-Bäume einbezogen wird. Die letztendliche Entscheidung, ob Verbindungen auf- oder abgebaut werden sollen, wird aufgrund des Vergleichs der (Un)Nützlichkeit mit entsprechenden Schwellenwerten getroffen, wobei die Anteile α und β in Abhängigkeit der vorherrschenden Gruppengröße gewählt werden. Durch geschickte Wahl von $\alpha > \beta$ vermeidet das Protokoll, dass soeben aufgebaute Verbindungen zeitnah wieder abgebaut werden und umgekehrt.

Um potenzielle Verbindungen bewerten zu können, bedient sich Narada eines einfachen Verfahrens. Dieses wird mit einer gewissen Periodizität ausgeführt und erfordert, dass jedes Gruppenmitglied g über eine Liste aller Gruppenmitglieder verfügt. Aus dieser Liste wählt g zufällig ein Gruppenmitglied g' aus, zu welchem noch keine Verbindung besteht. Durch Senden eines Ping-Pakets misst g die Paketumlaufzeit zu g' , wobei das von g' zurück gesendete Pong-Paket zusätzliche Information beinhaltet: Bei dieser handelt es sich um die Routing-Tabelle von g' . Obgleich auf diese erst in Abschnitt 3.4.3 näher eingegangen wird, wird hier jedoch bereits deutlich, dass die von Narada durchgeführte Optimierung der Overlay-Topologie eng an das auf dem Overlay aufgesetzte, explizite Routing-Verfahren gekoppelt ist. So wird die erhaltene Routing-Tabelle von g benötigt, um die Integration der neuen Overlay-Verbindung in Multicast-Bäume prüfen zu können. Daraufhin kann die Nützlichkeit der Verbindung zu g' berechnet und über ihren Aufbau entschieden werden.

Da auch während der Gruppenkommunikation beitretende Gruppenmitglieder in die Optimierung des Mesh mit einbezogen werden sollen, sieht Narada den fortlaufenden Austausch von Gruppeninformation anhand von **Refresh**-Paketen vor. Jedes Gruppenmitglied g hält daher eine entsprechende Tabelle, in welcher alle bekannten Gruppenmitglieder g_i aufgeführt sind. Mit jedem Gruppenmitglied g_i assoziiert g eine Sequenznummer s_{g_i} sowie einen Zeitstempel t_{g_i} . Während die von s_{g_i} angenommene Sequenznummer von g_i beim Versenden eines eigenen **Refresh**-Pakets vergeben wird, hält t_{g_i} den Zeitpunkt fest, zu welchem diese Sequenznummer von g erstmalig festgestellt wurde⁹. g versendet periodisch eigene **Refresh**-Pakete an alle im Overlay benachbarten Gruppenmitglieder. Für eine gleiche Periode enthalten die Pakete eine identische Sequenznummer, welche mit jeder Periode erhöht wird. Innerhalb der Pakete kodiert g die ihm bekannte Gruppeninformation als Tupel der Form $\langle g_i, s_{g_i} \rangle$. Erhält g selbst ein **Refresh**-Paket von einem benachbarten Gruppenmitglied g' , wird jedes darin enthaltene Tupel $\langle g'_i, s'_{g'_i} \rangle$ wie folgt verarbeitet: Ist g das Gruppenmitglied g'_i unbekannt oder gilt $s_{g'_i} < s'_{g'_i}$ so wird $s'_{g'_i}$ für g'_i übernommen und zusätzlich der mit g'_i assoziierte Zeitstempel $t_{g'_i}$ auf den aktuellen Zeitwert gesetzt. Andernfalls wird die Information verworfen. Hiermit wird sichergestellt, dass das Vorhandensein neuer Gruppenmitglieder propagiert wird und diese zur Optimierung des Mesh herangezogen werden.

Der besondere Entwurf der verwendeten Bewertungskriterien zum Auf- und Abbau von Overlay-Verbindungen hat zur Folge, dass die resultierende Overlay-Topologie lediglich eine dünne Vermaschung der Gruppenmitglieder aufweist. Dies kann dadurch erklärt werden, dass eine Overlay-Verbindung nur dann aufgebaut wird, falls sie einen nennenswerten Vorteil gegenüber dem Aneinanderhängen mehrerer, bereits vorhandener Overlay-Verbindungen bietet. Wie in Abbildung 3.6 verdeutlicht ist, kann hierdurch also insbesondere vermieden werden, dass eine Overlay-Verbindung zwischen zwei entfernten Gruppenmitgliedern aufgebaut wird, falls sich zwischen beiden Gruppenmitgliedern ein drittes Gruppenmitglied aufhält. So verzichtet Gruppenmitglied A beispielsweise auf den Aufbau einer Overlay-Verbindung zu B, da diese Verbindung keine Effizienzsteigerung hinsichtlich der Datenverteilung mit sich bringt: So zeigt sich das Erreichen von B über C als effizienter ($70ms + 90ms = 160ms < 170ms$). Dahingegen kann sich A sehr wohl für den Aufbau einer Overlay-Verbindung zu D entscheiden, da D direkt ($60ms$) besser zu erreichen ist als über E ($50ms + 30ms = 80ms > 60ms$).

Da das von Narada aufgebaute Overlay nur eine geringe Anzahl von Overlay-Verbindungen aufweist, entsteht durch deren Instandhaltung auf Vermittlungsschicht eine geringe Netzbelastung. Die tatsächliche von Narada induzierte Netzbelastung geht hauptsächlich auf das in Abschnitt 3.4.3 behandelte Routing-Verfahren zurück, sowie auf den Me-

⁹Die geführten Zeitstempel werden von Narada im Zusammenhang mit Gruppenaustritten und Ausfällen von Gruppenmitgliedern verwendet, so dass sie in Abschnitt 3.4.4 besprochen werden.

chanismus zur Propagation von Gruppenbeitritten. Dieser weist ein Kontrollverkehrsvolumen auf, welches quadratisch mit der Anzahl der Gruppenmitglieder wächst und daher lediglich zur Unterstützung kleiner Multicast-Gruppen geeignet erscheint.

– *NICE*: Das von NICE aufgebaute Overlay-Topologie wurde entworfen, um eine gute Skalierbarkeit hinsichtlich der Gruppenmitgliedszahl zu gewährleisten: So wird in [6] insbesondere im Kontext der in jedem Gruppenmitglied zu haltenden Zustandsinformation ein konstanter Aufwand für ein durchschnittliches Gruppenmitglied und ein logarithmischer Aufwand für den ungünstigsten Fall angegeben. Dies wird erreicht, indem die Gruppenmitglieder untereinander eine Cluster-Struktur auf mehreren Hierarchieebenen ausbilden, wie sie in Abbildung 3.7 angedeutet ist. Hier sind alle Gruppenmitglieder einer “Grundebene” (Ebene 0) beigetreten, auf welcher sie genau einem Cluster zugeordnet sind. Innerhalb eines Cluster wird ein Repräsentant bestimmt, welcher automatisch auch auf der nächst höheren Hierarchieebene vertreten ist. Auch hier werden die Gruppenmitglieder dem rekursiven Clustering-Prozess unterworfen, welcher erst dann endet, wenn auf einer Hierarchieebene lediglich ein einziges Gruppenmitglied vertreten ist. Dieses fungiert auf der höchsten Ebene ebenfalls als Cluster-Repräsentant. Somit ist jedes Gruppenmitglied g Mitglied genau eines Cluster auf einer (höchsten) Ebene g_{top} und Repräsentant genau eines Cluster auf allen Ebenen $i \in \mathbb{N}_0 : i < g_{top}$. Jedem Gruppenmitglied g sind genau diejenigen Gruppenmitglieder bekannt, die mit g in einem Cluster benachbart sind. Darüber hinaus kennt g die Mitglieder seines so genannten *Super-Cluster*: Bei diesem handelt es sich um denjenigen Cluster auf Ebene $g_{top} + 1$, in welchem dasjenige Gruppenmitglied vertreten ist, welches für g als Cluster-Repräsentant auf Ebene g_{top} fungiert. Betrachtet man Gruppenmitglied A in Abbildung 3.7, so gilt $A_{top} = 1$. In As Cluster auf dieser Ebene ist Gruppenmitglied B Repräsentant, so dass A über seinen Super-Cluster ebenso das Gruppenmitglied C kennt.

Jedes Gruppenmitglied hält zu *allen* bekannten Gruppenmitgliedern eine Overlay-Verbindung, so dass in jedem Cluster eine Vollvermaschung existiert. Über die bestehenden Overlay-Verbindungen erfolgen periodische Abstandsmessungen, wobei als Metrik RTTs verwendet werden. Mittels Heartbeat-Paketen signalisiert ein Cluster-Mitglied dem jeweiligen Repräsentanten und allen übrigen Cluster-Mitgliedern periodisch seine Cluster-Mitgliedschaft. Innerhalb der Heartbeats werden dabei alle Abstände kodiert, welche das Mitglied innerhalb des Cluster selbst gemessen hat. Analog signalisiert auch der Repräsentant eines Cluster den einzelnen Cluster-Mitgliedern periodisch seine Präsenz. Zusätzlich zu seinen Abstandsmessungen kodiert der Repräsentant in seinen Heartbeat-Paketen eine aktuelle Liste der Cluster-Mitglieder, sowie die Mitglieder des dem Cluster zugehörigen Super-Cluster.

Das NICE-Protokoll unterstützt Topologieanpassungen indem es einem Cluster-Mitglied g auf dessen Ebene g_{top} einen Cluster-Wechsel ermöglicht: So ordnet sich g auf

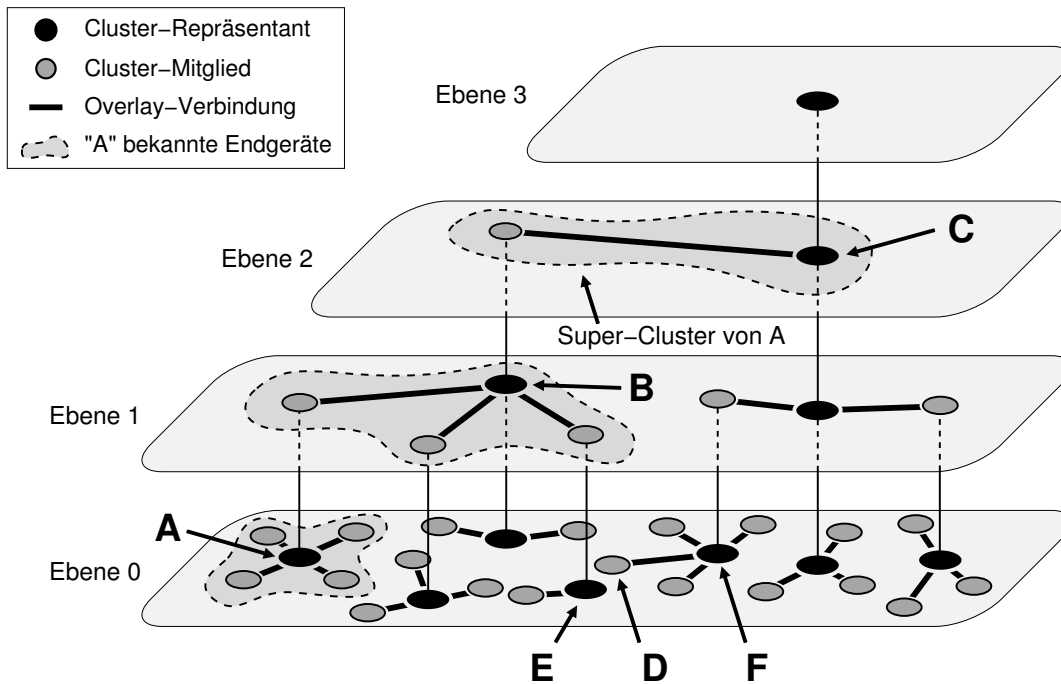


Abbildung 3.7 Die von NICE aufgebaute, hierarchische Cluster-Struktur. Auf die Darstellung der Vollvermaschung innerhalb eines Cluster wurde zur besseren Übersichtlichkeit verzichtet.

dieser Ebene stets demjenigen Cluster zu, dessen Repräsentant in gs Super-Cluster vertreten ist und einen minimalen Abstand aufweist. Übersteigt die Anzahl der in einem Cluster vertretenen Mitglieder einen festgesetzten Maximalwert, vollzieht der Cluster-Repräsentant eine Aufteilung des Cluster (engl. *Cluster-Split*): Hierfür teilt er die Cluster-Mitglieder (inklusive sich selbst) derart in zwei Teilmengen auf, dass der in beiden Teilmengen jeweils zwischen den Gruppenmitgliedern herrschende maximale Abstand annähernd minimiert wird¹⁰. Für beide Teilmengen wird anschließend dasjenige Gruppenmitglied bestimmt, welches den minimalen durchschnittlichen Abstand zu den übrigen Mitgliedern der gleichen Teilmenge aufweist: Die zwei so bestimmten Gruppenmitglieder übernehmen in den Teilmengen jeweils die Repräsentantenfunktion und treten somit der nächst höheren Ebene bei. Hierbei ist zu beachten, dass sich der bisherige Cluster-Repräsentant, sofern er sich selbst nicht erneut als Repräsentant bestimmt hat, von höheren Ebenen zurückziehen muss: Hierfür überträgt er in jedem betroffenen Cluster seine Repräsentantenfunktion an ein anderes Cluster-Mitglied. Analog zur Cluster-Aufteilung definiert das NICE-Protokoll die Verschmelzung (engl. *Cluster-Merge*) zweier Cluster. Diese wird von einem Cluster-Repräsentanten initiiert, sobald die Anzahl der Mitglieder eines Cluster unter einen bestimmten Minimalwert fällt. Als Folge dessen wird der betroffene Cluster mit einem nahe gelegenen Cluster verschmolzen.

¹⁰Es erfolgt lediglich eine Annäherung, da die optimale Lösung des Problems NP-hart ist [6].

Auch hier ist zu beachten, dass sich durch den Wegfall eines Cluster dessen ehemaliger Repräsentant von höheren Ebenen zurückziehen muss.

Obgleich sich das NICE-Protokoll im Kontext großer Gruppen im Festnetz als sehr performant erweist, zeigen sich im Zusammenhang kleinerer Gruppen in drahtlosen Ad-hoc-Netzen deutliche Ineffizienzen. Diese ergeben sich unmittelbar aus der aufwendigen Struktur der Overlay-Topologie. So kann beobachtet werden, dass jedes Gruppenmitglied insgesamt eine verhältnismäßig hohe Anzahl von Overlay-Verbindungen unterhält: So unterhält Gruppenmitglied A 8 Verbindungen (4 Verbindungen auf Ebene 0, 3 Verbindungen auf Ebene 1, 1 Verbindung zu Ebene 2) und Gruppenmitglied B 7 Verbindungen (2 Verbindungen auf Ebene 0, 3 Verbindungen auf Ebene 1, 1 Verbindung auf Ebene 2). Für diese Verbindungen sind auf Vermittlungsschicht Routen aufrecht zu halten, womit eine entsprechende Netzbelastung einhergeht. Im Hinblick auf die Topologieoptimierung erweist sich bei NICE die Overlay-Topologie als relativ unflexibel, da ein Gruppenmitglied g Anpassungen lediglich auf höchster Ebene g_{top} über seinen Super-Cluster erzielen kann. Wie in Abbildung 3.7 dargestellt ist, enthält insbesondere im Fall mobiler Szenarien der Super-Cluster aber nicht immer den tatsächlich nächstgelegenen Repräsentanten: So würde Gruppenmitglied D in E einen näheren Cluster-Repräsentanten finden als in F, doch verfügt D aufgrund disjunkter Super-Cluster über keinerlei Kenntnis von E. Darüber hinaus sind die von NICE verwendeten Mechanismen zur Pflege der Overlay-Topologie, wie etwa Cluster-Split- und -Merge-Operationen, als kritisch anzusehen: Während in deren Zusammenhang auch in [6] mit Inkonsistenzen gerechnet wird, ist in drahtlosen Ad-hoc-Netzen zu erwarten, dass diese Inkonsistenzen weitaus öfter auftreten und ihre Behebung entsprechend länger andauert.

– *PAST-DM*: PAST-DM baut ein Mesh-förmiges Overlay auf, bei welchem jedes Gruppenmitglied mit einer festgelegten, minimalen Anzahl anderer Gruppenmitglieder verbunden ist. Während im Laufe der Gruppenmitgliedschaft die Minimalzahl durchaus überschritten werden kann, wird dennoch versucht, die vorgegebene Nachbarzahl zu halten. Ein Beispiel einer von PAST-DM aufgebauten Overlay-Topologie ist in Abbildung 3.8 zu sehen. Das Protokoll versucht, in jedem Gruppenmitglied eine globale Sicht auf die Overlay-Topologie herzustellen. Hierzu führt jedes Gruppenmitglied eine Tabelle mit Topologie-Information. Die Information, wie sie ausgehend von Abbildung 3.8 von den Gruppenmitgliedern D und G geführt wird, ist in Tabelle 3.2 aufgetragen. Es ist zu erkennen, dass ein Gruppenmitglied g für jedes Gruppenmitglied g' (also auch für $g' = g$) eine *Link-State*-Menge $\mathcal{L}_{g'}$ zusammen mit einer dazugehörigen Sequenznummer und der Anzahl der Overlay-Hops zum Erreichen von g' führt. Ein Link-State $l \in \mathcal{L}_{g'}$ ist der Form $l = (x, y)$, wobei x ein Gruppenmitglied darstellt, welches im Overlay als Nachbar von g' fungiert. y wiederum nimmt den Abstand zwischen g' und x auf, welcher für ein Link-State in Übertragungsabschnitten auf Vermittlungsschicht gemessen

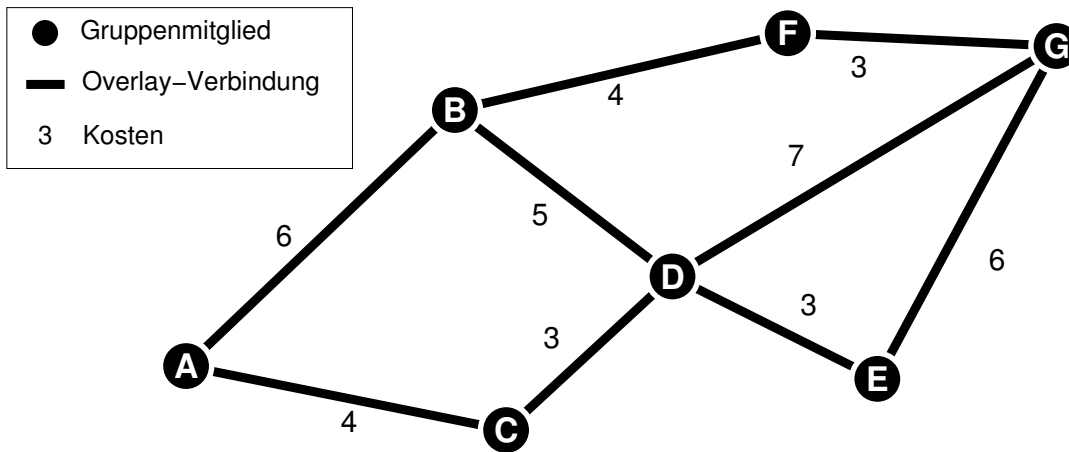


Abbildung 3.8 Das Mesh-förmige, von PAST-DM aufgebaute Overlay.

wird. Mit jeder Menge $\mathcal{L}_{g'}$ ist eine Sequenznummer assoziiert, welche von g' vergeben wird und der Identifikation aktueller Link-State-Information dient. Der in der letzten Tabellenspalte geführte Overlay-Hop-Abstand zu g' hat unter anderem einen Einfluss auf die durch g vorgenommene Weiterleitung der für g' gehaltenen Link-States.

Für ein Gruppenmitglied g erschließt sich die Menge \mathcal{L}_g direkt aus den im Overlay benachbarten Gruppenmitgliedern und den zu ihnen gemessenen Abständen. Mit jeder Abstandsmessung erhöht g die Sequenznummer für \mathcal{L}_g , da die Messung einer Aktualisierung der Topologieinformation gleicht. g versendet periodisch alle geführten Link-States $\mathcal{L}_{g'}$ und die damit jeweils assoziierten Sequenznummern an die im Overlay benachbarten Gruppenmitglieder. Hierdurch verteilt sich die Link-State-Information zwischen den Gruppenmitgliedern und stellt dort jeweils eine Gesamtsicht auf die Overlay-Topologie her. Um den quadratisch mit der Gruppengröße wachsenden Kontrollverkehr zu reduzieren, nutzt PAST-DM ein Verfahren ähnlich dem *Fish-Eye-State-Routing* [74]: So erschließt sich die Periodizität, mit welcher g die Link-State-Menge $\mathcal{L}_{g'}$ zu seinen Nachbarn propagiert, direkt aus dem zu g' bekannten Overlay-Hop-Abstand. Auf Basis einer Grundperiodizität P , mit welcher g die eigenen Link-States verschickt, ist die Periodizität P_o , mit welcher g die Link-States eines o Overlay-Hops entfernten Gruppenmitglieds versendet, definiert zu

$$P_o := \begin{cases} P, & o = 0, \\ o \cdot P, & o \in \{1, 2, 3\}, \\ 3 \cdot P, & \text{sonst.} \end{cases}$$

Empfängt g von einem im Overlay benachbarten Gruppenmitglied die Link-State-Information $\mathcal{L}_{g'}$ eines Gruppenmitglieds g' , so übernimmt g diese Information genau dann,

Von D gehaltene Topologieinformation:

	Link-States \mathcal{L}	Sequenznummer	Overlay-Hops
A	{(B, 6), (C, 4)}	15	2
B	{(A, 6), (D, 5), (F, 4)}	17	1
C	{(A, 4), (D, 3)}	9	1
D	{(B, 5), (C, 3), (E, 3), (G, 7)}	13	0
E	{(D, 3), (G, 6)}	25	1
F	{(B, 4), (G, 3)}	19	2
G	{(D, 7), (E, 6), (F, 3)}	8	1

Von G gehaltene Topologieinformation:

	Link-States \mathcal{L}	Sequenznummer	Overlay-Hops
A	{(B, 6), (C, 4)}	16	3
B	{(A, 6), (D, 5), (F, 4)}	18	2
C	{(A, 4), (D, 3)}	10	2
D	{(B, 5), (C, 3), (E, 3), (G, 7)}	11	1
E	{(D, 3), (G, 6)}	23	1
F	{(B, 4), (G, 3)}	20	1
G	{(D, 7), (E, 6), (F, 3)}	8	0

Tabelle 3.2 Bei PAST-DM gehaltene Topologieinformation auf Basis der Topologie aus Abbildung 3.8.

wenn sie eine höhere Sequenznummer als die bislang für g' geführte Link-State-Information aufweist.

Zur Berücksichtigung der Mobilität sucht PAST-DM anhand exponentieller Ringsuchen in periodischen Abständen nach nahe gelegenen Gruppenmitgliedern. Das genaue für die Topologieanpassung verwendete Verfahren ist in [51] jedoch nur unzureichend beschrieben. In dieser Arbeit wird daher angenommen, dass sich ein Gruppenmitglied g bevorzugt mit denjenigen Gruppenmitgliedern verbindet, welche einen geringen Hop-Abstand (im Sinne von Übertragungsabschnitten auf Vermittlungsschicht) aufweisen.

Damit g im Laufe der von PAST-DM vorgenommenen Topologieoptimierung zu einem nahe gelegenen Gruppenmitglied g' eine Overlay-Verbindung aufbauen kann, ist die Erfüllung zweier Anforderungen erforderlich. Einerseits muss ein Gruppenmitglied g^* existieren, zu welchem g derzeit eine Overlay-Verbindung hält, welche höhere Kosten als die neu aufzubauende Overlay-Verbindung aufweist. Andererseits darf das Abbauen der Overlay-Verbindung zwischen g und g^* nicht zu einer Partitionierung der Overlay-Topologie führen. Der Aufbau der Overlay-Verbindung zwischen g und g' erfolgt genau dann, wenn beide Anforderungen erfüllt sind. In diesem Falle wird anschließend die

Overlay-Verbindung zwischen g und g^* abgebaut. Kommt es durch den Abbau der Overlay-Verbindung bei g^* zu einem Unterschreiten der angestrebten Nachbarzahl, so sucht g^* anhand exponentieller Ringsuchen nach neuen Nachbarn und baut entsprechend neue Verbindungen auf.

Durch die Festlegung der von jedem Gruppenmitglied anzustrebenden Anzahl von Nachbarn kann bei PAST-DM der Vermaschungsgrad der Overlay-Topologie kontrolliert und damit die Anzahl der Overlay-Verbindungen gering gehalten werden. Topologieanpassungen lassen sich flexibel durchführen, da Overlay-Verbindungen prinzipiell zwischen beliebigen Gruppenmitgliedern aufgebaut werden können. Auch im Hinblick auf die später diskutierte Vermittlung von Datenpaketen sei an dieser Stelle bereits erwähnt, dass sich Anpassungen der Overlay-Topologie durch den Link-State-Austausch jedoch erst nach und nach zwischen den Gruppenmitgliedern verteilen. In Abhängigkeit der hierbei gewählten Periodizitäten kann sich die Wiederherstellung einer gruppenweiten, konsistenten Sicht auf die Overlay-Topologie entsprechend verzögern. Erfolgen derartige Topologieanpassungen aufgrund von Mobilität fortlaufend, ist deshalb zu erwarten, dass eine gruppenweite, konsistente Sicht auf die Overlay-Topologie nur sehr selten vorhanden ist. Ein zusätzlicher Nachteil des PAST-DM-Protokolls liegt in dem involvierten Kontrollverkehr. Trotz der oben beschriebenen Maßnahmen zu dessen Reduktion erfordert der Austausch von Link-State-Information im Vergleich zu anderen Protokollen einen erhöhten Bedarf an Bandbreite, welcher mit zunehmender Größe der Multicast-Gruppe rasch ansteigt.

– *ALMA*: Das Protokoll ALMA versucht insbesondere, den für den Aufbau und die Pflege der Overlay-Topologie anfallenden Kontrollverkehr zu reduzieren. Hierfür zielt es vor allem auf eine Reduktion der zu verwaltenden Overlay-Verbindungen und verwendet als Topologie zur Verbindung der Gruppenmitglieder daher eine Baumstruktur. So verfügt, abgesehen von der Wurzel des Baums, jedes Gruppenmitglied g über genau einen Elternknoten, welcher aus Sicht von g das nächste Gruppenmitglied auf dem Overlay-Pfad zur Wurzel ist. Darüber hinaus verfügt g über eine beschränkte Menge von Kindknoten, welche innerhalb des Baums die Menge der Nachfolger von g bilden.

Innerhalb des Baums ist jedes Gruppenmitglied dafür verantwortlich, die Verbindung zu seinem Elternknoten aufrecht zu halten und, zu Zwecken der Topologieanpassung, nach einem eventuell besseren Elternknoten Ausschau zu halten. Hierfür sendet ein Gruppenmitglied g periodisch Hello-Pakete zu seinem Elternknoten e . Das Eintreffen der von e gesendeten Antwort wird von g einerseits als Bestätigung der Verbindung zu e gesehen und andererseits zur Berechnung der Paketumlaufzeit (RTT) zwischen sich und e verwendet. Eine gemessene Paketumlaufzeit t wird zur Anpassung der Overlay-Topologie herangezogen, indem sie mit insgesamt drei fest definierten Schwellenwerten s_1 , s_2 und s_3 verglichen wird. Es wird der Schwellenwertindex $i \in 0, 1, 2, 3$ derart be-

stimmt, dass $s_i < t \leq s_{i+1}$ gilt (mit $s_0 = 0$ und $s_4 = \infty$). Der so bestimmte Index gibt an, in welchem Umkreis um den bestehenden Elternknoten e das Gruppenmitglied g innerhalb der Overlay-Topologie nach einem besseren Elternknoten sucht. Dies ist in Abbildung 3.9 beispielhaft skizziert, in welcher ein Gruppenmitglied **A** derzeit eine Verbindung zu seinem Elternknoten **E** hält. Die Abbildung veranschaulicht den Zusammenhang zwischen dem bestimmten Schwellenwertindex und der daraufhin erfolgenden Paketumlaufzeitmessung: Ist der kleinste Schwellenwert s_1 nicht überschritten, d.h. gilt $i = 0$ erfolgt die Messung lediglich zu dem aktuellen Elternknoten **E**. Bei fortlaufender Überschreitung der Schwellenwerte werden zusätzlich die Gruppenmitgliedsgruppen $\{F, G\}$ für $i \geq 1$, $\{H\}$ für $i \geq 2$ und $\{R\}$ für $i = 3$ in die Messungen mit einbezogen. Wichtig ist hierbei die Tatsache, dass das suchende Gruppenmitglied nur über eine lokale Sicht auf die Overlay-Topologie verfügt, d.h. lediglich seinen Eltern- und seine Kindknoten kennt. Umliegende Gruppenmitglieder, zu welchen Abstandsmessungen vorgenommen werden sollen, müssen daher zunächst in Erfahrung gebracht werden. [48] schlägt vor, die Struktur des Baums periodisch an weitergeleitete Datenpakete anzuhängen. Wird durch die erfolgenden Messungen ein Gruppenmitglied ausfindig gemacht, zu welchem eine geringere Paketumlaufzeit als zu dem bisherigen Elternknoten besteht, wird zu diesem eine Verbindung hergestellt und anschließend die Verbindung zum bisherigen Elternknoten getrennt.

Obleich der Aufwand, welcher mit diesem Verfahren zur Topologieanpassung verbunden ist, gering ist, ist die Effektivität der Anpassung fragwürdig. So ist einerseits zu erwarten, dass auch bei Überschreitung des höchsten Schwellenwertes ein vorhandener, besserer Elternknoten nicht notwendigerweise gefunden wird. Dies ist in Abbildung 3.9 verdeutlicht, in welcher es sich bei dem für **A** besten Elternknoten um das (nächstgelegene) Gruppenmitglied **I** handelt: Dieses kann jedoch auch mit dem Schwellenwertindex $i = 3$ nicht lokalisiert werden. Andererseits kann keine Garantie gegeben werden, dass ein besserer Elternknoten tatsächlich existiert. Entfernt sich in Abbildung 3.9 das Gruppenmitglied **X** beispielsweise zunehmend (nach rechts) von der übrigen Gruppe, so wird es seine Suche nach einem Elternknoten nach und nach ausweiten. Letztendlich wird **X** alle Gruppenmitglieder (**I**, **H**, **G**, **E**) in einem 3-Hop-Abstand um **R** in die fortlaufend durchgeführten Messungen mit einbeziehen, wobei diese sich wiederum störend auf diese Gruppenmitglieder auswirken.

3.4.2 Der Gruppenbeitritt

In diesem Abschnitt werden die in den einzelnen Protokollen umgesetzten Mechanismen zur Bewerkstelligung des Gruppenbeitritts untersucht.

- *Narada*: Die initiale Suche nach Gruppenmitgliedern ist für das Protokoll *Narada* nicht spezifiziert: Es wird lediglich davon ausgegangen, dass in Abhängigkeit der

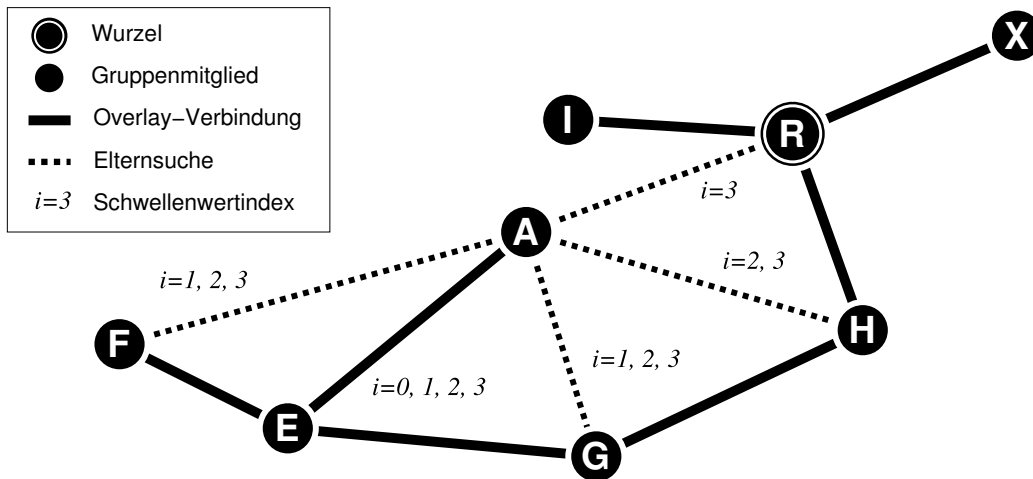


Abbildung 3.9 Die baumförmige Overlay-Topologie von ALMA und ihre Anpassung durch Bestimmung eines besseren Elternknotens.

letztendlich betriebenen Anwendung ein oder mehrere bereits beigetretene Gruppenmitglieder vorgegeben werden.

Zu seiner Eingliederung wählt ein beitretendes Gruppenmitglied g zufällig einige der bekannten Gruppenmitglieder aus und baut eine Verbindung zu ihnen auf. Als Folge dessen nehmen diese mit g den Austausch von Refresh-Paketen vor, so dass g Wissen über zusätzliche Gruppenmitglieder erlangt. Da der Aufbau der initialen Verbindungen willkürlich erfolgt, ist die Eingliederung von g anfangs nur mit geringer Wahrscheinlichkeit effizient. Durch Propagation des Beitritts zu den übrigen Gruppenmitgliedern anhand von Refresh-Paketen fangen Gruppenmitglieder jedoch zunehmend an, die Nützlichkeit von Verbindungen zu g zu bewerten und Overlay-Verbindungen gegebenenfalls aufzubauen. Da demnach mit der Zeit bessere Overlay-Verbindungen zur Verfügung stehen, kann es dazu kommen, dass die initial aufgebauten Overlay-Verbindungen kaum noch zur Datenverteilung verwendet und daher wieder abgebaut werden.

Ogleich sich die Eingliederung von g in die Overlay-Topologie durch den Austausch von nur wenigen Kontrollpaketen bewerkstelligen lässt und der Aufwand sich daher als sehr gering zeigt, ist zu beachten, dass sich an die initiale (ineffiziente) Eingliederung eine Phase anschließen kann, in welcher eine Anpassung der Eingliederung erfolgt. Die Länge dieser Phase hängt nicht nur von der Periodizität der Refresh-Pakete ab, sondern zusätzlich von der Periodizität, mit welcher Gruppenmitglieder die Verbindungsbewertung zu anderen Gruppenmitgliedern vornehmen. Als Folge dessen, kann die Anpassungsphase eine Dauer im Minutenbereich beanspruchen. Insbesondere in drahtlosen Ad-hoc-Netzen ist die Ineffizienz, mit welcher die Datenverteilung zu g in dieser Phase erfolgt, jedoch zu vermeiden, so dass Narada bezüglich des Eingliederungsaufwands lediglich mit *befriedigend* bewertet wird.

– *NICE*: Bei NICE erfolgt die initiale Nachbarsuche anhand eines Rendez-vous-Punkts (RVP), über welchen dasjenige Gruppenmitglied ermittelt wird, das sich derzeit (allein) auf der höchsten Hierarchieebene befindet und dort als Cluster-Repräsentant fungiert. Von diesem Gruppenmitglied ausgehend wird ein einfacher rekursiver Prozess gestartet, welcher in Pseudo-Code als Algorithmus 1 aufgeführt ist. Der dargestellten Prozedur *findeRepräsentanten*(r, i) werden zwei Parameter übergeben: Das vom RVP auf höchster Ebene ermittelte Gruppenmitglied r sowie dessen höchste Ebene i . Als Rückgabe liefert die Prozedur denjenigen Cluster-Repräsentanten, zu dessen Cluster auf Ebene 0 der Beitritt erfolgen soll. Zu Beginn (Zeile 1) wird geprüft, ob die übergebene Ebene bereits die Ebene 0 oder 1 ist: Ist dies der Fall (Zeile 2), wird das als Parameter übergebene Gruppenmitglied r zurückgegeben, da es sich dabei um den nächsten bekannten Cluster-Repräsentanten auf Ebene 0 handelt¹¹. Andernfalls (Zeile 4) wird von r eine Liste dessen Cluster-Mitglieder auf Ebene $i - 1$ erfragt, von welchen anschließend (Zeilen 5-10) dasjenige Cluster-Mitglied n mit dem geringsten Abstand ermittelt wird. Der Rückgabewert der Prozedur (Zeile 11) ist schließlich das Ergebnis des rekursiven Aufrufs der Prozedur, welcher nun das soeben ermittelte, nächste Cluster-Mitglied n und die Ebene $i - 1$ als Parameter übergeben werden. Ist ein Cluster-Repräsentant auf Ebene 0 gefunden, so kann der Beitritt zu dessen Cluster anhand eines entsprechenden Austauschs von Kontrollpaketen erfolgen. Übersteigt durch den Beitritt die Anzahl der Cluster-Mitglieder einen festgesetzten Maximalwert, vollzieht der Cluster-Repräsentant gemäß der zuvorigen Beschreibung eine Aufteilung des Cluster.

¹¹Dies gilt auch für einen Cluster-Repräsentanten auf Ebene 1, da dessen Cluster-Mitglieder auf Ebene 0 selbst keine Cluster-Repräsentanten auf tieferen Ebenen sein können.

procedure *findeRepräsentanten*(r, i)

Eingabe: Cluster-Repräsentant r , Ebene $i \in \mathbb{N}_0$

Ausgabe: Cluster-Repräsentant e

```

1: if ( $i \leq 1$ ) then
2:   return  $r$ 
3: else
4:    $M := \text{erfrageClusterMitglieder}(r, i - 1)$ 
5:    $n := r$ 
6:   for all  $m \in M$  do
7:     if ( $\text{bestimmeAbstandZu}(m) < \text{bestimmeAbstandZu}(n)$ ) then
8:        $n := m$ 
9:     end if
10:  end for
11:  return findeRepräsentanten( $n, i - 1$ )
12: end if

```

Algorithmus 1 Rekursion zum Auffinden eines Cluster-Repräsentanten auf Ebene 0.

Es wird schnell deutlich, dass in drahtlosen Ad-hoc-Netzen das Auffinden des nächstgelegenen Cluster-Repräsentanten auf Ebene 0 mittels des dargestellten Verfahrens ineffizient ist: So ist insbesondere mit einer hohen Netzbelastung einerseits und mit einer entsprechenden Verzögerung des Gruppenbeitritts andererseits zu rechnen. Dies ergibt sich aufgrund der Tatsache, dass die innerhalb der Cluster durchgeführten Abstandsmessungen die Existenz einer Route zu den jeweiligen Gruppenmitgliedern voraussetzen und einer Glättung bedürfen, um aussagekräftig zu sein. An dieser Stelle soll ebenfalls erwähnt sein, dass der Beitritt zu einem Cluster eine Kettenreaktion auslösen kann: Eine solche tritt ein, falls auf den Beitritt eine Aufteilung des Cluster folgt, und der dabei hinzukommende Repräsentant ebenfalls eine Aufteilung des Cluster auf der nächst höheren Ebene verursacht. Insbesondere in drahtlosen Ad-hoc-Netzen ist in solchen Situationen mit lang andauernden Inkonsistenzen zu rechnen. Problematisch erweist sich ebenfalls der rasche Beitritt zahlreicher Gruppenmitglieder zu einem neu gegründeten Overlay: In einem solchen existiert zunächst nur ein einziger Cluster auf Ebene 0, in welchen die Mitglieder beitreten. Übersteigt die Anzahl der Cluster-Mitglieder den oben genannten Maximalwert, ist der Cluster aufzuteilen. Eine *zielgerichtete* Aufteilung ist allerdings erst möglich, wenn dem Cluster-Repräsentanten entsprechende Abstandsmessungen aller Cluster-Mitglieder vorliegen. Während sich dies durch den fortlaufenden Beitritt neuer Mitglieder hinauszögern kann, erfolgen innerhalb des Cluster Abstandsmessungen mit quadratischem Aufwand, da jedes Cluster-Mitglied Messungen zu allen anderen Cluster-Mitgliedern vornimmt. In drahtlosen Ad-hoc-Netzen kann die damit verbundene Netzbelastung schnell kritische Ausmaße annehmen und die Kommunikation innerhalb des Ad-hoc-Netzes zeitweise zum Erliegen bringen.

– *PAST-DM*: Die initiale Nachbarsuche erfolgt bei *PAST-DM* vollständig dezentral anhand einer exponentiell wachsenden Ringsuche (ERS), während welcher *GroupRequest*-Pakete mit einer entsprechenden Lebenszeit geflutet werden. Bereits beigetretene Gruppenmitglieder, welche ein solches Anfragepaket empfangen, vermerken das beitretende Gruppenmitglied als neuen Nachbarn und beantworten das Anfragepaket anhand eines *GroupReply*-Pakets. Um dem beitretenden Mitglied schnell zu einer Sicht auf die Overlay-Topologie zu verhelfen, sind in der versendeten Antwort bereits Link-State-Informationen kodiert. Das beitretende Gruppenmitglied wiederholt die Suche mit wachsenden Lebenszeiten bis entweder die geforderte Minimalzahl von Nachbarn oder die vorgegebene, maximale Lebenszeit der *GroupRequest*-Pakete erreicht ist. In diesem Fall wird erst später, im Laufe der periodisch durchgeführten Nachbarsuche, versucht, die Nachbarzahl zu vervollständigen.

Nach einem Gruppenbeitritt kann es analog zu obigem Fall der Detektion neuer Nachbarn für einige Gruppenmitglieder zu dem Überschreiten der angestrebten Nachbarzahl

kommen. Diesem wird mit der Identifikation und dem Abbau überschüssiger Overlay-Verbindungen unter Vermeidung einer Partitionierung entgegengewirkt.

Durch den beschriebenen Mechanismus gestaltet sich der Beitritt von neuen Gruppenmitgliedern bei PAST-DM als effizient: So genügt nach der Lokalisierung bereits beigetretener Gruppenmitglieder der Austausch eines einzigen *GroupReply*-Pakets, um das beitretende Gruppenmitglied in die Overlay-Topologie einzugliedern. Allein die Tatsache, dass *alle* Gruppenmitglieder, welche ein geflutetes *GroupRequest*-Paket erhalten, das beitretende Gruppenmitglied als neuen Nachbarn ansehen, verursacht eine gewisse Ineffizienz: So werden bei einer erhöhten Anzahl umliegender Gruppenmitglieder entsprechend zahlreiche und oftmals unnötige Overlay-Verbindungen zu dem beitretenden Gruppenmitglied aufgebaut, so dass für dieses die angestrebte Nachbarzahl bereits überschritten wird und überschüssige Verbindungen zügig wieder abgebaut werden.

– *ALMA*: Die initiale Suche nach Gruppenmitgliedern, über welche die Eingliederung eines neuen Gruppenmitglieds erfolgen kann, ist für *ALMA* nicht genau spezifiziert. Stattdessen wird für den Erhalt einer Liste existierender Gruppenmitglieder entweder das Vorhandensein eines Rendez-vous-Punkts (RVP) oder die Durchführung einer lokalen Suche angenommen. Aus dieser Liste werden mehrere Gruppenmitglieder ausgewählt und jeweils ein *Join*-Paket an sie gerichtet. Letztendlich wird eine Overlay-Verbindung zu demjenigen Gruppenmitglied aufgebaut, von welchem die erste Antwort auf die versendeten *Join*-Pakete erhalten wird.

Während sich die Eingliederung damit sehr einfach und mit wenig Aufwand bewerkstelligen lässt, ergibt sich, analog zu *Narada*, das Problem, dass die Auswahl des Gruppenmitglieds, zu welchem eine Overlay-Verbindung aufgebaut wird, zufällig erfolgt. Damit ist die Eingliederung des neuen Gruppenmitglieds keineswegs als effizient anzusehen. Daher wird sich oftmals eine Anpassungsphase direkt an die Eingliederung anschließen, während welcher die Verbindung des neuen Gruppenmitglieds zum Overlay verbessert wird. Ausgehend von dem bei *ALMA* eingesetzten Anpassungsverfahren sind jedoch wiederum viele Situationen denkbar, in welchen der letztendlich effizienteste Elternknoten nicht gefunden werden kann.

3.4.3 Die Vermittlung von Datenpaketen

In diesem Abschnitt werden Mechanismen zur Weiterleitung von Datenpaketen zwischen den einzelnen Gruppenmitgliedern diskutiert.

– *Narada*: Um die Overlay-Topologie zur Weiterleitung von Daten nutzen zu können, setzt *Narada* auf dem aufgebauten Mesh ein getrenntes Routing-Protokoll ein. Dieses basiert ähnlich wie das *Distance Vector Multicast Routing Protocol (DVMRP [42, 110])* auf einem *Shortest-Path*-Algorithmus, für welchen ein expliziter Austausch zusätzlicher Routing-Information erforderlich ist: So tauscht jedes Gruppenmitglied mit seinen im

Overlay benachbarten Gruppenmitgliedern periodisch Routing-Pakete aus. In diesen sind die Kosten kodiert, welche jeweils zum Erreichen aller übrigen Gruppenmitglieder erforderlich sind. Als Kosten werden hierbei die für die Anpassung der Overlay-Topologie gemessenen Paketumlaufzeiten verwendet, wobei diese für Overlay-Pfade über mehrere Gruppenmitglieder entsprechend aufsummiert werden (vgl. Diskussion von Abbildung 3.6). Durch den periodischen Austausch propagieren sich die Kosten allmählich zwischen allen Gruppenmitgliedern, so dass diese für jedes Gruppenmitglied den zugehörigen Shortest-Path bestimmen können. Um dabei das bekannte *Count-to-infinity*-Problem zu vermeiden, werden in den Routing-Paketen, ähnlich zu BGP [86], nicht nur die tatsächlichen Kosten zum Erreichen der Gruppenmitglieder versendet, sondern auch den dazugehörigen Shortest-Path entlang des Overlay. Die Weiterleitung von Datenpaketen auf Basis der Shortest-Path-Information erfolgt dabei wie folgt: Erhält ein Gruppenmitglied g von einem im Overlay benachbarten Gruppenmitglied g' ein von der Multicast-Quelle s stammendes Datenpaket, so leitet g dieses Datenpaket nur dann weiter, falls g' das erste Gruppenmitglied auf dem Shortest-Path zu s ist. Darüber hinaus leitet g das Datenpaket nur dann an einen Nachbarn n weiter, falls g das erste Gruppenmitglied auf dem Shortest-Path von n zu s ist.

Ein Beispiel einer Datenverteilung bei Narada ist in Abbildung 3.10 gezeigt, in welcher das Gruppenmitglied F als Multicast-Quelle agiert. Zusätzlich ist in Tabelle 3.3 die in den Gruppenmitgliedern A und D gehaltene Routing-Information aufgeführt, wobei die für F verwendete Routing-Information grau hinterlegt ist. In der ersten Spalte geben die Routing-Tabellen das Gruppenmitglied an, welches erreicht werden soll. Der dafür verwendete Pfad entlang des Mesh ist in der zweiten Spalte eingetragen, gefolgt von der zugehörigen Umlaufzeit. Die letzten beiden Spalten bilden die Information, welche direkt zur Weiterleitung von Datenpaketen verwendet wird: So leitet beispielsweise Gruppenmitglied D nur Datenpakete einer Multicast-Quelle F weiter, welche von G empfangen werden, da G aus Sicht von D den Vorgänger auf dem Pfad von F nach D bildet. In diesem Fall leitet D das Datenpaket an eine Menge von Nachfolger $\mathcal{N} = \{E, H\}$ weiter: Diese Menge ergibt sich aus denjenigen Gruppenmitgliedern, welche mit D im Overlay benachbart sind und D als Vorgänger für F gewählt haben.

Aufgrund des obigen Routing-Verfahrens bilden sich bei Narada Quellen-spezifische Verteilbäume aus. Die Routing-Konsistenz hängt maßgeblich von der Periodizität der Routing-Pakete ab. So kann es dazu kommen, dass eine Overlay-Verbindung zwischen zwei Gruppenmitgliedern g und g' abgebaut wird, obwohl sie von g' noch zur Weiterleitung von Datenpaketen einer Quelle s an g verwendet wird. Durch den Verbindungsabbau muss g von seinen Overlay-Nachbarn ein neues Gruppenmitglied g^* bestimmen, über welches der Shortest-Path zu s gebildet wird. Bis g^* über seine Wahl informiert ist, ist das Routing inkonsistent, da sich g^* seiner Aufgabe der Weiterleitung von Datenpaketen

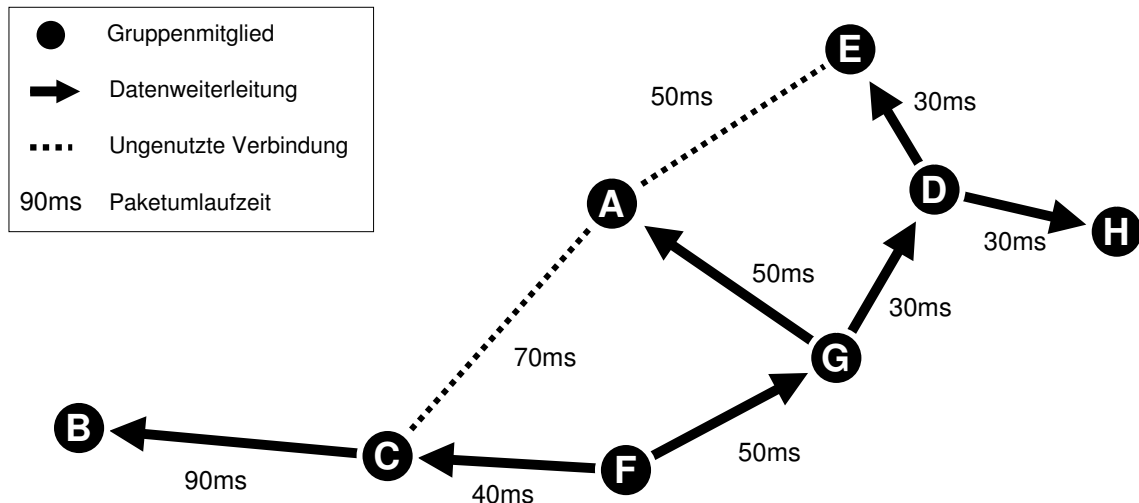


Abbildung 3.10 Die Weiterleitung von Datenpaketen erfolgt bei Narada aufgrund eines *Shortest-Path*-Algorithmus.

von s an g noch nicht bewusst ist. Während ein häufiger Austausch von Routing-Paketen für eine höhere Konsistenz sorgt, ist damit jedoch auch eine stärkere Netzbelastung verbunden. Dies gilt insbesondere für den von Narada verfolgten Ansatz, bei welchem Routing-Pakete vollständige Pfade zu allen Gruppenmitgliedern umfassen.

Die Stabilität der Verteilbäume hängt wiederum von der durchgeführten Glättung der Paketumlaufzeiten ab. Da in [38] nicht näher darauf eingegangen wird, soll an dieser Stelle lediglich erwähnt werden, dass eine stärkere Glättung eine höhere Stabilität der Verteilbäume mit einer langsameren Anpassung der Overlay-Topologie erkauft. Dahingegen kann eine schwächere Glättung häufigere Topologieänderungen und somit instabilere Verteilbäume bedingen.

– *NICE*: Durch die Struktur der Overlay-Topologie ist bei NICE die Wegewahl zur Vermittlung von Datenpaketen implizit vorgegeben: Ein sendendes Gruppenmitglied g leitet ein Datenpaket an alle Gruppenmitglieder weiter, die mit g in einem Cluster benachbart sind. Empfängt ein Gruppenmitglied g' ein Datenpaket von einem Gruppenmitglied g , leitet g' das Datenpaket wiederum innerhalb aller Cluster weiter, in denen g' Mitglied ist, mit Ausnahme jedoch desjenigen Cluster, in welchem auch g Mitglied ist. Auf diese Weise prägt sich für jedes Gruppenmitglied ein Verteilbaum aus, welcher sich auf oberster Ebene als Quellen-spezifisch erweist. In Abhängigkeit der Position, an welcher als Quellen agierende Gruppenmitglieder in die NICE-Hierarchie eingegliedert sind, weisen die verwendeten Multicast-Bäume deutliche Ähnlichkeit auf. Dies ist in Abbildung 3.11 skizziert, in welcher eine NICE-Hierarchie vereinfacht von oben dargestellt ist, indem die Hierarchieebenen als konzentrische Kreise um Gruppenmitglieder kodiert sind. In Teilbild 3.11.a) agiert ein Gruppenmitglied A, welches lediglich auf Hierarchie-

Teil der von A gehaltenen Routing-Information:

Ziel	Pfad	Umlaufzeit	Vorgänger	Nachfolger \mathcal{N}
H	→ E → D → H	110ms	E	\emptyset
F	→ G → F	100ms	G	\emptyset
B	→ C → B	160ms	C	{E}
E	→ E	50ms	E	{C}
...

Teil der von D gehaltenen Routing-Information:

Ziel	Pfad	Umlaufzeit	Vorgänger	Nachfolger \mathcal{N}
A	→ E → A	80ms	E	{H}
B	→ G → F → C → B	210ms	G	{H}
F	→ G → F	80ms	G	{E, H}
E	→ E	30ms	E	{G, H}
...

Tabelle 3.3 Die von Narada gehaltene Routing-Information.

ebene 0 als Cluster-Mitglied vertreten ist, als Quelle. Hierzu leitet A seine Datenpakete an alle Gruppenmitglieder weiter, welche in dem gleichen Cluster auf Ebene 0 vertreten sind. Der Repräsentant R dieses Cluster ist zusätzlich auf Ebene 1 vertreten und leitet die Datenpakete an die dort im gleichen Cluster vertretenen Gruppenmitglieder S und T. Diese übernehmen wiederum die Weiterleitung in jedem Cluster, in dem sie auf Ebene 0 als Repräsentant fungieren. In Teilbild 3.11.b) agiert nun ein anderes Gruppenmitglied B, welches auf Ebene 0 jedoch dem gleichen Cluster angehört wie A, als Quelle. Analog zum vorherigen Fall, erfolgt die Datenverteilung zunächst innerhalb des Cluster durch B selbst und anschließend über den Cluster-Repräsentanten R in Richtung der übrigen Gruppenmitglieder. Dabei fällt auf, dass sich ab der durch R vorgenommenen Weiterleitung der verwendete Multicast-Baum mit dem aus dem vorherigen Fall (Quelle A) deckt. Da sich also die Multicast-Bäume mehrerer Gruppenmitglieder sehr ähnlich sind und jeweils aus den gleichen Overlay-Verbindungen bestehen, weisen sie eher einen globalen als Quellen-spezifischen Charakter auf (* in Tabelle 3.1). Die Verteilbäume selbst erweisen sich bei NICE als stabil. Dies ergibt sich aus der mangelnden Flexibilität des NICE-Overlay, welche die Anzahl der möglichen Topologieanpassungen deutlich einschränkt. Darüber hinaus bewirken die erforderlichen Glättungen der zwischen Gruppenmitgliedern durchgeführten Abstandsmessungen lediglich sehr selten eine Topologieanpassung.

Aufgrund der mitunter sehr komplexen Vorgänge zur Umstrukturierung der Overlay-Topologie im Falle von Topologieanpassungen weist das NICE-Overlay lang anhaltende

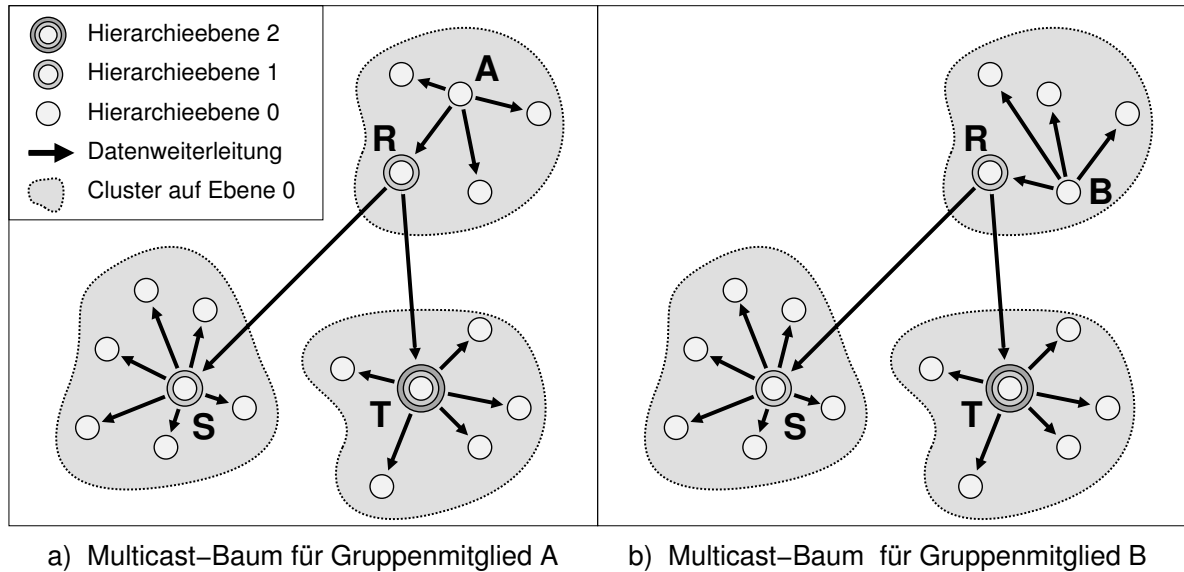


Abbildung 3.11 Ähnlichkeit der durch das NICE Protokoll implizit vorgegebenen Wegewahl für die Vermittlung von Datenpaketen.

Inkonsistenzen bei der Weiterleitung von Datenpaketen auf [11]. Diese äußern sich für einige Gruppenmitglieder durch große Lücken während des Empfangs von Datenpaketen. Da die betroffenen Gruppenmitglieder folglich keine Weiterleitung der Datenpakete vornehmen können, sind oftmals ganze Teilbäume von den Inkonsistenzen betroffen, so dass die gesamte Gruppenkommunikation entsprechend in Mitleidenschaft gezogen wird.

– *PAST-DM*: Die globale, von PAST-DM in jedem Gruppenmitglied aufgebaute Sicht auf die Overlay-Topologie wird zur Weiterleitung von Datenpaketen verwendet. Für die letztendliche Wegewahl verfolgt PAST-DM dabei einen expliziten, losen Source-Routing-Ansatz: Wünscht ein Gruppenmitglied g Datenpakete zu versenden, berechnet es ausgehend von der vorliegenden Link-State-Information einen Verteilbaum mit minimalen Kosten. Für dessen Berechnung wird ein heuristisches Verfahren zur Berechnung von Steiner-Bäumen [100] verwendet. Während es sich bei g um die Wurzel des Verteilbaums handelt, wird die erste Ebene von Gruppenmitgliedern g_i gebildet, welche im Overlay direkte Nachbarn von g sind. Jedes dieser Gruppenmitglieder ist aus Sicht von g verantwortlich für die Weiterleitung eines Datenpakets an eine bestimmte Menge von Gruppenmitgliedern S_i , welche als Subgruppe von g_i bezeichnet wird. Da die Link-State-Information, welche g für mehrere Overlay-Hops entfernte Gruppenmitglieder vorliegt, aufgrund der Austauschperiodizität nicht aktuell ist, wird die Struktur des Verteilbaums nicht vollständig von g vorgegeben sondern im Zuge der eigentlichen Weiterleitung verteilt berechnet. So wird g_i anhand eines entsprechenden Paketkopfs in den Datenpaketen lediglich über seine Subgruppe S_i und nicht über die vorzunehmenden Weiterleitungen

informiert. Vielmehr berechnet g_i aus der erhaltenen Subgruppe sowie der vorliegenden Link-State-Information selbst einen Verteilbaum, welcher genau die Gruppenmitglieder \mathcal{S}_i umfasst. Durch Bildung und Weiterleitung der Subgruppeninformation an die entsprechenden Nachbarn wiederholt sich das Verfahren, bis für ein Gruppenmitglied die erhaltene Subgruppe der leeren Menge entspricht. Das Verfahren ist exemplarisch in Abbildung 3.12 skizziert, in welcher ein Gruppenmitglied D als Quelle agiert.

Die für den Link-State-Austausch gewählte Periodizität hat bei PAST-DM einen direkten Einfluss auf die Konsistenz des Routing, da dieses Wegewahlentscheidungen auf Basis der vorliegenden Topologieinformation trifft. Allgemein kann für PAST-DM festgehalten werden, dass sich Änderungen in der Overlay-Topologie nur langsam propagieren, so dass von einer Multicast-Quelle berechnete Verteilbäume sowie Subgruppen einen ineffizienten und sogar inkonsistenten Charakter aufweisen. So kann es vorkommen, dass einer Multicast-Quelle q Wissen über eine Overlay-Verbindung zwischen zwei Gruppenmitgliedern g und g' vorliegt, und q aufgrund dieser Verbindung g' der gleichen Subgruppe \mathcal{S}_i zuordnet wie g . Wurde die Verbindung zwischen g und g' inzwischen jedoch abgebaut, kann es zu einer Partitionierung des Verteilbaums für die Subgruppe \mathcal{S}_i kommen. Diese tritt genau dann ein, wenn

$$\forall g'' \in \mathcal{S}_i : \neg (\exists \text{ Overlay-Verbindung zwischen } g' \text{ und } g'')$$

In einem solchen Fall fühlt sich kein Gruppenmitglied aus \mathcal{S}_i für die Weiterleitung des Datenpakets an g' zuständig, so dass für g' Lücken im Datenempfang entstehen. Diese Lücke hält an bis die Topologieänderung zur Quelle propagiert wurde.

Aufgrund des Source-Routing-Ansatzes bildet PAST-DM Quellen-spezifische Verteilbäume aus, welche für jede Multicast-Quelle getrennt berechnet werden. In Abhängigkeit der Mobilität erweisen sich die Verteilbäume als mitunter sehr instabil. Tatsächlich ergeben sich wegen des kontinuierlichen Austauschs von Link-State-Information und der verteilten Berechnung der Verteilbäume und Subgruppen fortlaufend Änderungen bei der Weiterleitung von Datenpaketen.

– *ALMA*: Aufgrund der Baumstruktur der Overlay-Topologie ist die Wegewahl zur Weiterleitung von Datenpaketen bei ALMA implizit definiert, so dass keine zusätzliche Routing-Information ausgetauscht wird. Als direkte Konsequenz der baumförmigen Overlay-Topologie ergibt sich auch der globale Verteilbaum, da für alle Multicast-Quellen die gleichen Overlay-Verbindungen genutzt werden.

Das Overlay-Routing zeigt sich bei ALMA im Wesentlichen als konsistent, da gemäß dem in Abschnitt 3.4.1 vorgestellten Verfahren ein Gruppenmitglied auch im Zuge von Topologieanpassungen immer eine Verbindung zu der Overlay-Topologie hält. Dahingehen sind über die Verteilbaumstabilität nur bedingt Aussagen möglich: Diese hängt

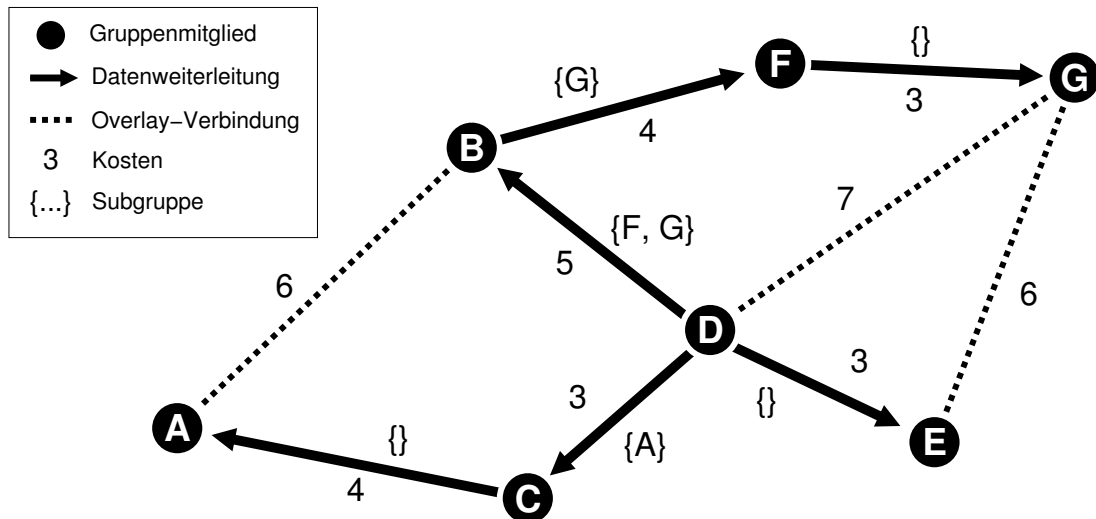


Abbildung 3.12 Subgruppen, wie sie bei PAST-DM zur Weiterleitung von Daten berechnet werden, und der daraus entstehende Verteilbaum.

bei ALMA maßgeblich von den gewählten Schwellenwerten s_1, \dots, s_3 und der durchgeführten Glättung der Paketumlaufzeiten ab. Da in [48] jedoch nicht näher auf diese Aspekte eingegangen wird, soll an dieser Stelle lediglich angemerkt werden, dass die Wahl der Schwellenwerte keineswegs trivial ist. Werden sie zu gering angesetzt, können aufgrund von Schwankungen der gemessenen Paketumlaufzeiten häufige Topologieänderungen erfolgen und zeitnah wieder rückgängig gemacht werden. Dies hätte demnach instabile Verteilbäume zur Folge. Werden die Schwellenwerte hingegen zu hoch angesetzt, wird die Topologie starr, da sie Änderungen der Kommunikationswegeigenschaften kaum noch berücksichtigt. Hier wiederum wären die Verteilbäume äußerst stabil, wenn auch ineffizient.

3.4.4 Die Ausgliederung von Gruppenmitgliedern

Die Ausgliederung eines Gruppenmitglieds G aus einer Overlay-Topologie kann aufgrund dessen Wunsches erfolgen, oder da umliegende Gruppenmitglieder den Ausfall von G bemerkt haben. Im Folgenden wird die Behandlung beider Fälle für die untersuchten Protokolle beschrieben.

- *Narada:* Zur Bewerkstelligung des Austritts eines Gruppenmitglieds g verzichtet Narada auf das sofortige Ausgliedern von g aus der Overlay-Topologie durch Abbau der entsprechenden Overlay-Verbindungen. Stattdessen erfolgt die Ausgliederung in direktem Zusammenspiel mit dem eingesetzten Routing-Protokoll: So gibt g für die Overlay-Verbindungen, die es zu anderen Gruppenmitgliedern hält, künstlich überhöhte Kosten bekannt. Hierdurch wird erreicht, dass die entsprechenden Overlay-Verbindungen nach und nach aus den Multicast-Bäumen ausgeschlossen und letztendlich durch den Prozess

der Topologieanpassung abgebaut werden. Während sich die Ausgliederung von g demnach über einen längeren Zeitraum erstrecken kann, wird jedoch sichergestellt, dass die Ausgliederung konsistent erfolgt und es demnach zu keinen Lücken in der Datenverteilung kommt.

Zur Detektion eines ausgefallenen Gruppenmitglieds g werden die durch Refresh-Pakete erhaltenen Sequenznummern und Zeitstempel herangezogen. So ergibt sich nach dem Ausfall von g keine weitere Erhöhung der mit ihm assoziierten Sequenznummer. Liegt deren letzte Aktualisierung eine gewisse Zeitspanne zurück, senden die im Overlay benachbarten Gruppenmitglieder dem ausgefallenen Gruppenmitglied bis zu einer bestimmten Anzahl von Probe-Paketen zu. Bleiben diese unbeantwortet, wird der Ausfall erkannt und innerhalb des Overlay propagiert.

Sowohl im Zuge von Gruppenaustritten als auch bei Ausfällen von Gruppenmitgliedern kann es zur Partitionierung des Mesh kommen. Analog zu Ausfällen von Gruppenmitgliedern lässt sich in einem solchen Fall für die unerreichbaren Gruppenmitglieder keine kontinuierliche Erhöhung der mit ihnen assoziierten Sequenznummern mehr feststellen. Anders als bei tatsächlichen Ausfällen werden die versendeten Probe-Pakete nun jedoch beantwortet, so dass die Partitionierung entdeckt und durch den Aufbau entsprechender Overlay-Verbindungen behoben werden kann.

– *NICE*: In Abhängigkeit der Position, an welcher ein Gruppenmitglied g innerhalb des NICE-Overlay eingegliedert ist, erweist sich dessen selbst eingeleiteter Gruppenaustritt mitunter als sehr aufwendig. Gilt $g_{top} = 0$ ist der Austritt einfach zu bewerkstelligen, da sich g lediglich bei seinem Repräsentanten auf Ebene 0 abmelden muss: Innerhalb seiner nachfolgend ausgesendeten Heartbeat-Pakete kodiert der Repräsentant die aktuelle Liste der Cluster-Mitglieder und propagiert somit den Gruppenaustritt von g . Gilt hingegen $g_{top} > 0$, ist der Gruppenaustritt nicht ohne weiteres möglich. Tatsächlich agiert g in diesem Fall als Cluster-Repräsentant auf mindestens einer Ebene und muss hier vor seinem Austritt seine Repräsentantenaufgabe zunächst an ein anderes Cluster-Mitglied delegieren. Hierzu bestimmt g innerhalb der Cluster aufgrund der vorliegenden Abstandsinformation dasjenige Gruppenmitglied mit dem minimalen durchschnittlichen Abstand und ernennt dieses Gruppenmitglied zum neuen Cluster-Repräsentanten. Im schlimmsten Fall handelt es sich bei g um dasjenige Gruppenmitglied, welches allein auf höchster Hierarchieebene vertreten ist. In diesem Fall ist auf jeder Ebene ein Cluster von dem Gruppenaustritt betroffen, so dass in drahtlosen Ad-hoc-Netzen die damit verbundene, aufwendige Umstrukturierung des NICE-Overlay oft von Inkonsistenzen begleitet ist.

Ausfälle von Gruppenmitgliedern werden bei NICE durch das Ausbleiben der periodisch versendeten Heartbeats erkannt: Gruppenmitglieder, welche für eine bestimmte Zeitspanne kein Heartbeat von einem innerhalb eines Cluster benachbarten Gruppen-

mitglied erhalten, entfernen dieses aus der Liste der Cluster-Mitglieder. Während sich der Ausfall eines Gruppenmitglieds g mit $g_{top} = 0$ als unkompliziert erweist, zeigt sich jedoch der Ausfall eines Cluster-Repräsentanten auf einer höheren Ebene als problematisch. Hier ist in jedem betroffenen Cluster eine Neuwahl des Repräsentanten erforderlich. Dies erfolgt, indem jedes Cluster-Mitglied aufgrund der vorliegenden Abstandsinformation dasjenige Cluster-Mitglied mit dem minimalen durchschnittlichen Abstand bestimmt und dieses anhand eines Pakets als neuen Repräsentant wählt. Aufgrund von ungleichen Abstandsinformationen kann es zu der Wahl mehrerer Repräsentanten kommen. Dieser Zustand wird erst durch das Versenden der nächsten Heartbeat-Pakete erkannt und aufgelöst, indem die überflüssigen Repräsentanten ihre Funktion abgeben. Während die hierfür eingesetzten Protokollmechanismen in [6] nicht genau beschrieben sind, soll an dieser Stelle nur auf die lang andauernden Inkonsistenzen hingewiesen werden, welche als Folge des Ausfalls eines Gruppenmitglieds auftreten können.

– *PAST-DM*: Verlässt ein Gruppenmitglied g die Multicast-Gruppe, so trennt es die Verbindungen zu den im Overlay benachbarten Gruppenmitgliedern anhand von Paketen des Typs **Group_LV**. Diese entfernen daraufhin g sowie den zugehörigen Link-State aus ihrer Topologieinformation, so dass der Gruppenaustritt durch den periodisch erfolgenden Link-State-Austausch zu den übrigen Gruppenmitgliedern propagiert wird. Obgleich g deshalb fälschlicherweise noch eine Weile in den von Multicast-Quellen berechneten Verteilbäumen und Subgruppen erscheint, entstehen hierdurch kaum Inkonsistenzen bei der Weiterleitung von Datenpaketen. Der Grund hierfür ist, dass nur diejenigen Gruppenmitglieder das Gruppenmitglied g in die Datenverteilung einbeziehen würden, welche ehemals eine Overlay-Verbindung zu g gehalten haben. Genau diese Gruppenmitglieder wurden jedoch von g direkt über den Gruppenaustritt informiert, so dass sich Inkonsistenzen erst im Falle des Verlustes von **Group_LV**-Paketen ergeben. Solche Inkonsistenzen wiederum ließen sich durch Quittierungen und etwaige Übertragungswiederholungen der **Group_LV**-Pakete vermeiden, wobei derartige Mechanismen in [51] jedoch nicht vorgesehen sind.

Ebenso wenig wird in [51] auf die Behandlung von Ausfällen einzelner Gruppenmitglieder eingegangen. Durch das wiederholte Ausbleiben der periodisch erwarteten Link-State-Information kann jedoch ein ausgefallenes Gruppenmitglied nach einer gewissen Zeitspanne erkannt und aus der Liste der benachbarten Gruppenmitglieder gestrichen werden. Analog zu obigem Fall des Gruppenaustritts, wird auch hier der Ausfall des Gruppenmitglieds erst nach und nach zu den übrigen Gruppenmitgliedern propagiert.

– *ALMA*: Die Ausgliederung eines Gruppenmitglieds g erfolgt bei ALMA indem g ein entsprechendes **Leave**-Paket an seinen Eltern- und an seine Kindknoten sendet und hierdurch die jeweiligen Overlay-Verbindungen abbaut. Somit verlieren insbesondere die Kindknoten kurzzeitig die Anbindung an die Overlay-Topologie. Die Anbindung wird

erst wieder hergestellt indem die Kindknoten der Multicast-Gruppe erneut beitreten: In diesem Zusammenhang sind daher deutliche Lücken in der Datenverteilung zu erwarten. Auf das Protokollverhalten, im Fall des Austritts des Wurzelknotens wird in [48] nicht näher eingegangen.

Ausfälle von Gruppenmitgliedern werden bei ALMA durch das Ausbleiben der periodisch zwischen Eltern- und Kindknoten ausgetauschten Kontrollpakete detektiert. Aus Sicht eines Elternknotens kann der betroffene Kindknoten einfach aus der Liste der Nachfolger gestrichen werden. Aus Sicht eines Kindknotens dagegen wird bei Verlust des Elternknotens ein neuer Gruppenbeitritt initiiert.

3.4.5 Die Nutzung der Semi-Broadcast-Eigenschaft

Dieser Abschnitt gibt einen Überblick über die Nutzung der Semi-Broadcast-Eigenschaft in den in dieser Arbeit betrachteten Protokollen.

- *Narada und NICE*: Da diese Protokolle für den Einsatz im Internet entworfen wurden, integrieren sie keinerlei Mechanismen, um das für Ad-hoc-Netze typische geteilte Medium und dessen Semi-Broadcast-Eigenschaft vorteilhaft zu nutzen.

- *PAST-DM*: Obgleich PAST-DM speziell für den Einsatz in drahtlosen Ad-hoc-Netzen entwickelt wurde, berücksichtigt es kaum das Vorhandensein eines geteilten Mediums sowie dessen Semi-Broadcast-Eigenschaft. In der Tat beschränkt sich dessen Nutzung auf die Lokalisierung umliegender Gruppenmitglieder zum Aufbau des Mesh-förmigen Overlay. Alle übrigen Kontrollinformationen werden anhand gewöhnlicher Unicast-Übertragungen ausgetauscht.

- *ALMA*: Das Protokoll ALMA verzichtet bewusst auf Optimierungen, welche aus dem Einsatz des geteilten Mediums hervorgehen, um faire Vergleiche zu Multicast-Routing-Protokollen auf Vermittlungsschicht zu erlauben [48].

3.4.6 Zuverlässigkeit und Staukontrolle

In diesem Abschnitt wird diskutiert inwieweit Zuverlässigkeitsaspekte in den untersuchten Arbeiten berücksichtigt werden.

- *Narada, NICE und PAST-DM*: Diese Protokolle gehen in keiner Weise auf die Integration Anwendungs-spezifischer Zuverlässigkeitsmechanismen ein. Zwar werden in [51] Größen wie die erreichte Paketzustellrate (vgl. Kapitel 7) untersucht, doch erfolgt dies im Kontext einer vollständig unzuverlässigen Datenverteilung.

- *ALMA*: Während das Protokoll die Integration Anwendungs-spezifischer Zuverlässigkeitsmechanismen nicht explizit vorsieht, wird das Protokoll in [48] sowohl im Kontext unzuverlässiger Overlay-Verbindungen als auch im Zusammenhang mit einer Absicherung über TCP untersucht. Darüber hinaus werden zusätzliche Zuverlässigkeitsmechanismen eingeführt, aufgrund welcher im Falle von Strukturänderungen innerhalb der

Overlay-Topologie Verluste von Datenpaketen (wie sie in Abbildung 3.3 verdeutlicht sind) teilweise ausgeglichen werden können. Die in der Arbeit aufgeführten Messungen lassen jedoch keine Rückschlüsse über die dabei auftretenden Verzögerungen zu, so dass der erzielte Nutzen im Kontext unterschiedlicher Anwendungen unklar bleibt.

3.5 Zusammenfassung

In diesem Kapitel wurden die bislang wichtigsten Ansätze zur P2P-basierten Gruppenkommunikation, welche in direktem Zusammenhang zu dieser Arbeit stehen, vorgestellt und bewertet. Hierfür wurde zunächst eine Liste von Anforderungen erarbeitet, welche an eine in drahtlosen Ad-hoc-Netzen betriebene Overlay-Topologie gestellt werden, um einen möglichst effizienten Ablauf der Gruppenkommunikation zu gewährleisten. Bei der anschließend erfolgten Bewertung wurden die in den betrachteten Ansätzen verwendeten Protokollmechanismen näher beleuchtet und, darauf aufbauend, argumentiert inwieweit die jeweiligen Mechanismen in drahtlosen Ad-hoc-Netzen versagen oder nicht den reibungslosen Ablauf der Gruppenkommunikation ermöglichen.

In diesem Kapitel wurde mitunter festgestellt, dass die hier betrachteten Ansätze keine akzeptable Unterstützung einer Gruppenkommunikation in drahtlosen Ad-hoc-Netzen ermöglichen. Aus diesem Grund sollen in den folgenden Kapiteln Protokolle und Mechanismen erarbeitet werden, welche einerseits eine solche Unterstützung erlauben und andererseits den vorgestellten Anforderungen gerecht werden.

4. TrAM: Ein leichtgewichtiges Overlay für drahtlose Ad-hoc-Netze

Bei dem Protokoll *TrAM* (*Tree-based Overlay Architecture for MANETs*) handelt es sich um ein P2P-basiertes Gruppenkommunikationsprotokoll, welches speziell für den Einsatz in drahtlosen Ad-hoc-Netzen entwickelt wurde [13, 63]. Es baut eine Overlay-Topologie auf, welche darauf ausgelegt ist,

- mit ihrem leichtgewichtigen Charakter ein hohes Maß an Effizienz zu erzielen,
- die geringe verfügbare Bandbreite nicht durch Vermittlung von Kontrollinformation übermäßig zu beanspruchen und gleichzeitig
- ein hohes Maß an Flexibilität zu wahren, um eine gute Anpassung an die Topologie des drahtlosen Ad-hoc-Netzes bewerkstelligen zu können.

TrAM erreicht dies indem es auf Redundanz verzichtet und die Gruppenmitglieder anhand einer baumförmigen Overlay-Topologie verbindet. Das Protokoll sieht vor, dass Gruppenmitglieder die Qualität ihrer Verbindungen innerhalb des Overlay fortlaufend überwachen, um Änderungen in der Topologie des drahtlosen Ad-hoc-Netzes entsprechend berücksichtigen und die Overlay-Topologie daraufhin anpassen zu können. In den folgenden Abschnitten werden die Mechanismen für den Aufbau der Overlay-Topologie erläutert und bewertet.

4.1 Aufbau und Erhalt der Overlay-Topologie

Aus logischer Sicht entsteht TrAMs baumförmiges Overlay, im Folgenden auch als TrAM-Baum bezeichnet, indem jedes darin eingegliederte Gruppenmitglied G eine Menge von Kindknoten G_{children} verwaltet und über einen Elternknoten G_{parent} verfügt. Sowohl die Kindknoten als auch der Elternknoten besitzen die Eigenschaft, dass es sich bei ihnen ebenfalls um Mitglieder der Multicast-Gruppe handelt, welche daher ebenfalls in die Overlay-Topologie eingegliedert sind. G_{children} entspricht G s Menge von Nachfolgern innerhalb des TrAM-Baums, während G_{parent} den Vorgänger im Bezug auf die Wurzel des TrAM-Baums darstellt. Bei der Initialisierung des Protokolls gilt $G_{\text{children}} = \emptyset$ und $G_{\text{parent}} = 0$, d.h. G verfügt weder über einen Kindknoten noch über einen gültigen Elternknoten. Für die Aufnahme von Elementen in die Menge G_{children} sowie zur Lokalisierung und Auswahl eines Elternknotens verwendet das Protokoll unterschiedliche Mechanismen, welche in den folgenden Abschnitten näher betrachtet werden. Zur Verdeutlichung der Abläufe und der dabei ausgetauschten Informationen sind die vom TrAM-Protokoll verwendeten Kontrollpakete in Abbildung 4.1 spezifiziert.

4.1.1 Das Entstehen des TrAM-Baums

Das von TrAM aufgebaute baumförmige Overlay fordert die Existenz eines Wurzelknotens W , an welchem sich weitere Gruppenmitglieder beim Aufbau des Overlay ausrichten. Die Wurzelfunktion wird von dem ersten Gruppenmitglied ausgeübt, welches der Multicast-Gruppe beiträgt: Es versucht zunächst anhand des in Abschnitt 4.1.3 beschriebenen Verfahrens zur Lokalisierung von Elternknoten eine Verbindung zu einem bestehenden TrAM-Baum aufzubauen. Da dieser initial nicht vorhanden ist, schlägt der Verbindungsaufbau fehl, so dass sich das Gruppenmitglied selbst als Wurzelknoten deklariert und somit eine neue Multicast-Gruppe sowie den dazugehörigen TrAM-Baum gründet¹². Die so ausgebildete Wurzel fungiert fortan als Anknüpfungspunkt für künftig beitretende Gruppenmitglieder. Sie ist das einzige Gruppenmitglied, das über keinen Elternknoten verfügt.

Um den Beitritt zu einem TrAM-Baum zu bewerkstelligen, sucht ein beitretendes Gruppenmitglied G nach einem bereits vorhandenen Gruppenmitglied E , welches in der Lage ist, als Elternknoten zu fungieren. Bei diesem kann es sich entweder um den Wurzelknoten selbst oder um ein beliebiges anderes Gruppenmitglied handeln, welches bereits erfolgreich in das Overlay eingegliedert ist: Mit Ausnahme des Wurzelknotens gilt ein Gruppenmitglied dabei als genau dann erfolgreich in das Overlay eingegliedert, wenn es über einen gültigen Elternknoten verfügt. Gelingt G der Verbindungsaufbau zu E , so verfügt G von nun an ebenfalls über einen gültigen Elternknoten ($G_{\text{parent}} = E$) und ist

¹²Kann die Wurzel eines TrAM-Baums vorübergehend nicht erreicht werden, können im Laufe des Gruppenbeitritts anderer Gruppenmitglieder folglich mehrere Overlays parallel aufgebaut werden.

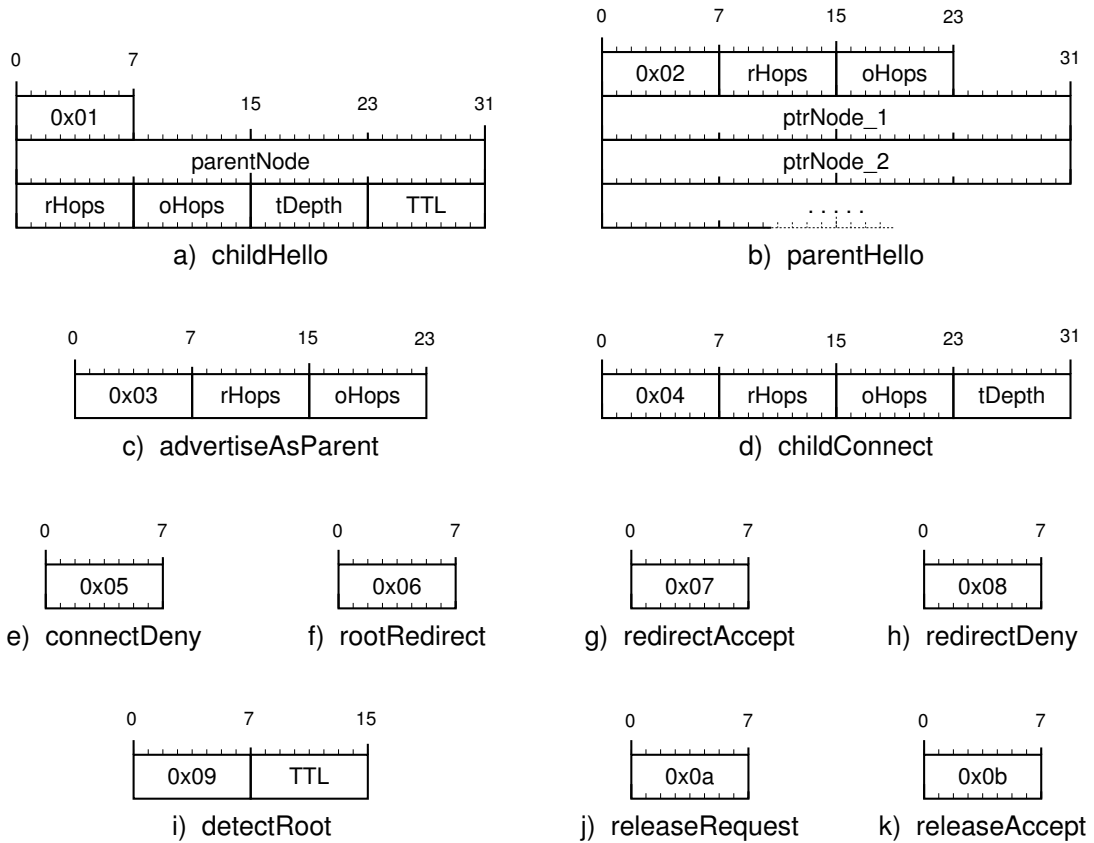


Abbildung 4.1 Kontrollpakete zum Aufbau und Erhalt des TrAM-Overlay.

somit einerseits in das Overlay eingegliedert und andererseits in der Lage, ebenfalls als Elternknoten zu fungieren.

Durch die Eingliederung von G in das Overlay entsteht aus Sicht von G der so genannte *Wurzelpfad*. Dieser entspricht einer Kette von Overlay-Hops

$$G = K^0 \rightarrow K^1 \rightarrow K^2 \rightarrow \dots \rightarrow K^n = W,$$

wobei

$$\forall i = 0, \dots, n - 1 : K_{parent}^i = K^{i+1}.$$

Die Länge des Wurzelpfades, d.h. die Anzahl der Overlay-Hops, die aus Sicht von G zum Erreichen der Wurzel erforderlich sind, spielt eine maßgebende Rolle bei der Auswahl des Elternknotens bzw. bei der damit verbundenen Bewertung von Gruppenmitgliedern.

4.1.2 Die Bewertung umliegender Gruppenmitglieder

Für den Aufbau einer Verbindung zum TrAM-Overlay, bzw. zu dessen Wahrung und eventuellen Verbesserung, nimmt jedes Gruppenmitglied eine Bewertung umliegender

Gruppenmitglieder vor, da diese potenzielle Elternknoten darstellen. Die Bewertung erfolgt aufgrund zweier Größen: Einerseits fließt die Anzahl der Übertragungsabschnitte $rHops$ (*Routing-Hops*) zu dem betrachteten Gruppenmitglied mit ein, andererseits wird dessen Wurzelpfadlänge $oHops$ (*Overlay-Hops*) verwendet. Beide Größen bilden zusammen die so genannte *Verbindungsbewertung* des entsprechenden Gruppenmitglieds. Um eine gute Differenzierbarkeit von Gruppenmitgliedern zu gewährleisten, werden die Größen nicht zu einem Wert verrechnet, sondern immer getrennt voneinander betrachtet.

Beispiele von Verbindungsbewertungen sind in Abbildung 4.2 dargestellt, in welcher die Bewertungen unterschiedlicher potenzieller Elternknoten E_i ($i = 1, \dots, 4$) sowie des Wurzelknotens W aus Sicht eines Gruppenmitglieds G verdeutlicht sind. Aus Sicht von G ist das Gruppenmitglied E_1 4 Übertragungsabschnitte (Spalte $rHops$) entfernt, und es sind 2 Overlay-Hops (Spalte $oHops$) zum Erreichen des Wurzelknotens erforderlich ($E_1 \rightarrow E_2 \rightarrow W$). Gruppenmitglied E_2 kann ebenfalls über 4 Übertragungsabschnitte erreicht werden, und es ist ein einziger Overlay-Hop zum Erreichen der Wurzel notwendig. An dieser Stelle ist noch einmal die Funktionalität eines jeden Endgeräts in drahtlosen Ad-hoc-Netzen als weiterleitende Instanz verdeutlicht: So ist der Abbildung zu entnehmen, dass E_3 auch als in das Overlay eingegliederte Gruppenmitglied eine weiterleitende Funktion bei der Kommunikation zwischen G und E_2 einnehmen kann. E_3 selbst ist über einen Übertragungsabschnitt weniger erreichbar als E_2 und bietet die Erreichbarkeit der Wurzel in 2 Overlay-Hops ($E_3 \rightarrow E_2 \rightarrow W$). Während das Gruppenmitglied E_4 anhand von genauso vielen Übertragungsabschnitten erreichbar ist wie E_3 , ist die Anzahl von Overlay-Hops zum Erreichen von W jedoch um 1 höher ($E_4 \rightarrow E_3 \rightarrow E_2 \rightarrow W$). Der Wurzelknoten W selbst weist einen Abstand von 7 Übertragungsabschnitten und einen Overlay-Hop-Abstand zur Wurzel von 0 auf.

Die Bewertung von Elternknoten ist abhängig von dem aktuellen Beitrittsstatus eines Gruppenmitglieds G . Handelt es sich um ein beitretendes Gruppenmitglied, verwendet es zur Bewertung von Elternknoten den Algorithmus 2, wobei bei den Berechnungen der Operator “.” verwendet wird, um den Kontext zu dem vorangestellten Elternknoten herzustellen. Der Algorithmus gibt für zwei als Parameter übergebene Elternknoten A und B den jeweils besseren zurück, wobei sowohl die Wurzelpfadlänge als auch die Anzahl der Übertragungsabschnitte zum Elternknoten minimiert werden. Durch die Gewichtung α des Unterschieds der Wurzelpfadlänge in Zeile 3 kann vorwiegend die Anzahl der Übertragungsabschnitte oder die Wurzelpfadlänge minimiert werden. Da in drahtlosen Ad-hoc-Netzen eine zuverlässige Datenübertragung zu nahen Endgeräten einfacher zu bewerkstelligen ist als zu entfernten Endgeräten, ist eine Minimierung der Anzahl der Übertragungsabschnitte durch eine Wahl von $\alpha < 1$ zu bevorzugen.

Ist das Gruppenmitglied G beigetreten, so dass ein gültiger Elternknoten vorliegt, wird zur Bewertung potenzieller Elternknoten der Algorithmus 3 verwendet: Dieser gibt als

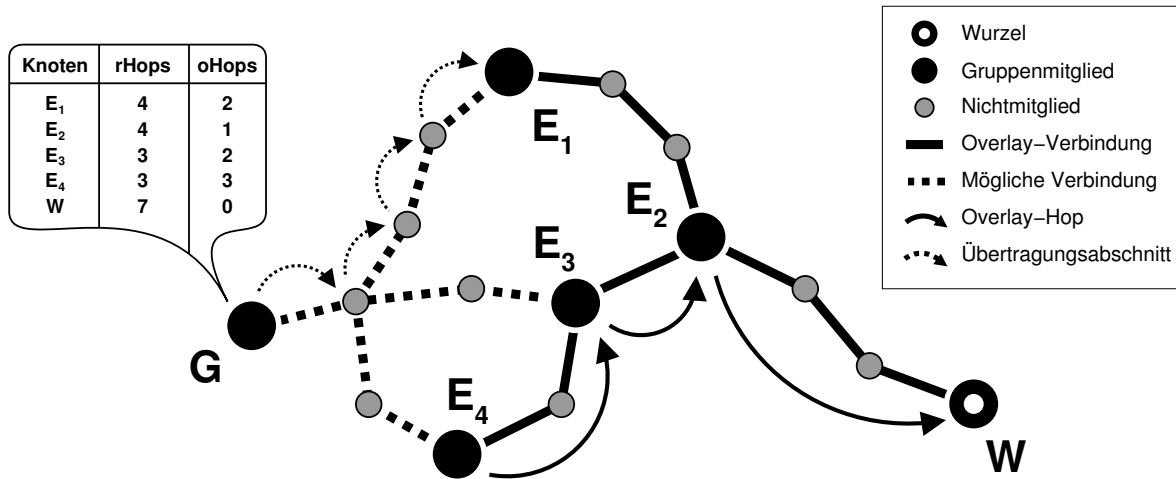


Abbildung 4.2 Bewertung potenzieller Elternknoten E_1, \dots, E_4 und W aus Sicht eines Gruppenmitglieds G .

Ergebnis einen Wahrheitswert zurück, welcher angibt, ob der Wechsel zu dem als zweiten Parameter übergebenen Elternknoten P stattfinden oder ob das Gruppenmitglied bei seinem bestehenden Elternknoten B verbleiben soll. Während der eigentliche Vergleich der Elternknoten in Zeile 4 analog zu dem Vergleich aus Algorithmus 2 erfolgt, wird für beigetretene Gruppenmitglieder jedoch die zusätzliche Bedingung ($\Delta o + 2 \geq 0$) in Zeile 3 benötigt, um die Bildung von Schleifen zu verhindern: So kommt für ein Gruppenmitglied G ein potenzieller Elternknoten nur dann infrage, wenn er eine Wurzelpfadlänge aufweist, welche maximal 2 Overlay-Hops länger als die Wurzelpfadlänge des bestehenden Elternknotens ist. Wie im Folgenden erläutert liegt dies darin begründet, dass für potenzielle Elternknoten, deren Wurzelpfadlänge die des bisherigen Elternknotens um 3 (oder mehr) übersteigt, die Bildung von Schleifen innerhalb der Overlay-Topologie möglich wird:

procedure *besterElter*(A, B)

Eingabe: Potenzieller Elternknoten A, B

Ausgabe: Gewählter Elternknoten E

- 1: $\Delta o := A.oHops - B.oHops$;
- 2: $\Delta r := A.rHops - B.rHops$;
- 3: **if** ($\Delta r + \alpha \cdot \Delta o \geq 0$) **then**
- 4: **return** B ;
- 5: **end if**
- 6: **return** A ;

Algorithmus 2 Algorithmus zum Treffen der Entscheidung, welcher von beiden übergebenen Elternknoten für den initialen Beitritt zum TrAM-Overlay verwendet werden soll.

- Mit $\Delta o \geq 0$ weist P eine kleinere Wurzelfadlänge als G auf. Beim Wechsel von Elternknoten B zu P kann folglich keine Schleife entstehen, da sich P im TrAM-Baum oberhalb von G befindet.
- Auch für Werte $\Delta o = -1$ kann keine Schleife entstehen, da P die gleiche Wurzelfadlänge wie G aufweist, so dass sich demzufolge P und G auf der gleichen Ebene innerhalb des TrAM-Baums befinden.
- Mit $\Delta o = -2$ entsteht eine Schleife genau dann, wenn es sich bei P im TrAM-Baum um einen Nachfolger von G handelt. Dieser Fall kann jedoch abgefangen werden, da G seine Nachfolger kennt und diese nicht als potenzielle Elternknoten in Betracht zieht.
- Mit $\Delta o \leq -3$ tritt eine Schleife genau dann auf, wenn es sich bei P um ein Gruppenmitglied handelt, welches in G s Teilbaum unterhalb von G liegt. Eine solche Schleife kann nicht detektiert werden, da G innerhalb des TrAM-Baums nur seinen Vorgänger und seine direkten Nachfolger kennt. Durch die Fallunterscheidung in Zeile 3 von Algorithmus 3 bleiben derartige Elternknoten unberücksichtigt, so dass die Schleifenbildung verhindert werden.

Die maßgebende Aufgabe des TrAM-Protokolls besteht in der Lokalisierung eines Elternknotens, über welchen eine möglichst effiziente Verbindung zum TrAM-Baum hergestellt werden kann. Der im folgenden Abschnitt vorgestellte Mechanismus bestimmt mit möglichst geringem Aufwand eine Menge solcher infrage kommenden Elternknoten sowie deren jeweilige Verbindungsbewertung und wählt daraus den, gemäß obiger Metrik, besten zur Verbindung mit dem TrAM-Baum aus.

procedure elternWechsel(B, P)

Eingabe: Bestehender Elternknoten B , Potenzieller Elternknoten P

Ausgabe: Boolean e

```
1:  $\Delta o := B.oHops - P.oHops;$ 
2:  $\Delta r := B.rHops - P.rHops;$ 
3: if ( $\Delta o + 2 \geq 0$ ) then
4:   if ( $\Delta r + \alpha \cdot \Delta o > 0$ ) then
5:     return true;
6:   end if
7: end if
8: return false;
```

Algorithmus 3 Algorithmus zum Treffen der Entscheidung, ob ausgehend von einem bestehenden Elternknoten B zu einem potenziellen Elternknoten P gewechselt werden soll.

4.1.3 Die Suche und Auswahl des Elternknotens

Der Mechanismus zur Lokalisierung und Auswahl von Elternknoten ist für neu beitretende Gruppenmitglieder weitgehend identisch wie für bereits beigetretene: Es ergeben sich lediglich Unterschiede bzgl. des Inhalts einzelner Paketfelder. Der Mechanismus wird daher für beide Beitrittsarten gleichzeitig beschrieben und an den entsprechenden Stellen eine Differenzierung vorgenommen. Zum besseren Verständnis der Vorgänge ist in Abbildung 4.3 ein Zustandsübergangsdiagramm dargestellt. Bei diesem befindet sich ein Endgerät vor einem erfolgten Gruppenbeitritt in dem Zustand **Unverbunden**. Grau hinterlegte Zustände symbolisieren Zustände, in welchen weiterführende Protokollmechanismen greifen, die aus Gründen der Übersichtlichkeit erst später diskutiert werden.

Zur Lokalisierung von potenziellen Elternknoten sendet ein Gruppenmitglied G mit einer bestimmten Periodizität $t_{refresh}$ Pakete des Typs **childHello** aus. Derartige Pakete beinhalten Information über den bestehenden Elternknoten G_{parent} , nämlich dessen Adresse im Feld **parentNode**, sowie dessen Verbindungsbewertung in den Feldern **rHops** und **oHops**. Zusätzlich beinhalten **childHello**-Pakete ein Feld **tDepth**, dessen genaue Bedeutung in Abschnitt 4.1.7 erläutert wird. Handelt es sich bei G um ein neu beitretendes und nicht um einen bereits beigetretenes Gruppenmitglied, nehmen die einzelnen Felder durchgehend den Wert 0 an.

Die ausgesendeten **childHello**-Pakete werden innerhalb des drahtlosen Ad-hoc-Netztes geflutet, wobei die Lebenszeit derart gewählt wird, dass sie bis zu dem eventuell bestehenden Elternknoten G_{parent} reicht. Zusätzlich wird für die Wahl der Paketlebenszeit ebenfalls berücksichtigt, ob eventuell vorangegangene Versuche zur Lokalisierung eines Elternknotens erfolglos geblieben sind. Die für ein **childHello**-Paket verwendete Lebenszeit wird in dem Paketfeld **TTL** eingetragen und aus

$$TTL_i := \begin{cases} 2^i & , \text{ falls Gruppenmitglied neu beitretend,} \\ 2^i + rHops & , \text{ falls Gruppenmitglied bereits beigetreten,} \end{cases}$$

berechnet, wobei $rHops$ die Anzahl der Übertragungsabschnitte zu dem eventuell bestehenden Elternknoten G_{parent} angibt. i nimmt initial den Wert 0 an und wird im Falle von erfolglosen Suchen jeweils um 1 erhöht. Als Folge dessen wird bei ausbleibenden Antworten der Suchbereich exponentiell vergrößert, um eine zügige Lokalisierung von Elternknoten zu gewährleisten. Im Fall einer erfolgreichen Suche wird i auf den Wert 0 zurückgesetzt. An dieser Stelle sei angemerkt, dass bereits beigetretene Gruppenmitglieder initial ($i = 0$) einen Suchbereich wählen, welcher einen Übertragungsabschnitt über den tatsächlichen Abstand des aktuellen Elternknotens hinausreicht. Hierdurch wird eine gewisse Mobilitätsunterstützung gewährleistet: So kann ein sich räumlich entfernter Elternknoten, von welchem lediglich veraltete Abstandsinformation vorliegt,

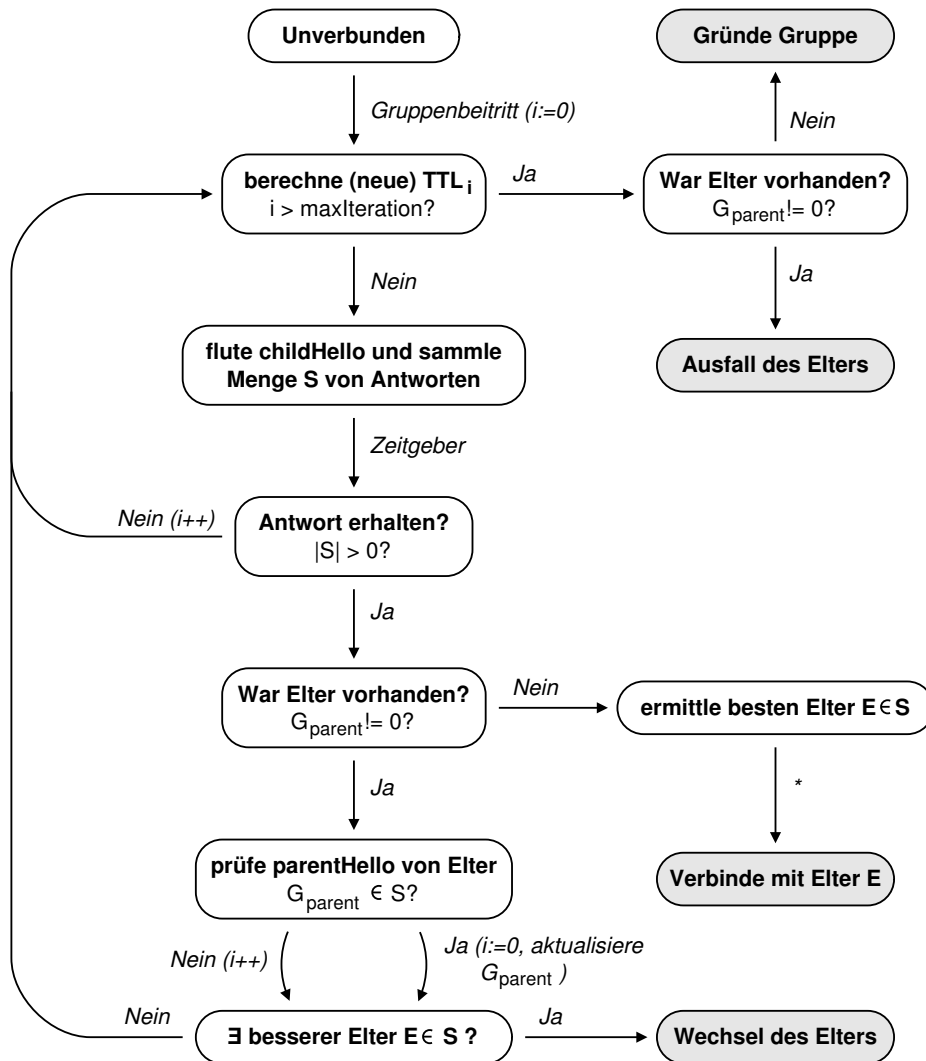


Abbildung 4.3 Zustandsautomat für die Suche nach einem Elternknoten.

noch immer erreicht werden. Steigt die Anzahl der Übertragungsabschnitte zu dem Elternknoten sprunghaft an, so bleibt die initiale Elternsuche erfolglos. Jedoch kann der Suchbereich durch seine anschließende exponentielle Vergrößerung schnell an den neuen Abstand angepasst werden.

Empfängt ein potenzieller Elternknoten P ein `childHello`-Paket (kurz `CH`) von einem Gruppenmitglied G , so verläuft die in Abbildung 4.4 veranschaulichte Verarbeitung des Pakets durch eine Verkettung von drei Fallunterscheidungen:

- Ist G ein bei P registriertes Kind? Bei Empfang des `childHello`-Pakets überprüft P zunächst, ob es sich bei G um einen bereits bei ihm registrierten Kindknoten handelt. Ist dies der Fall, bestätigt P die Registrierung von G anhand eines `parentHello`-Pakets, dessen genauer Aufbau in Abschnitt 4.1.4 erläutert wird.

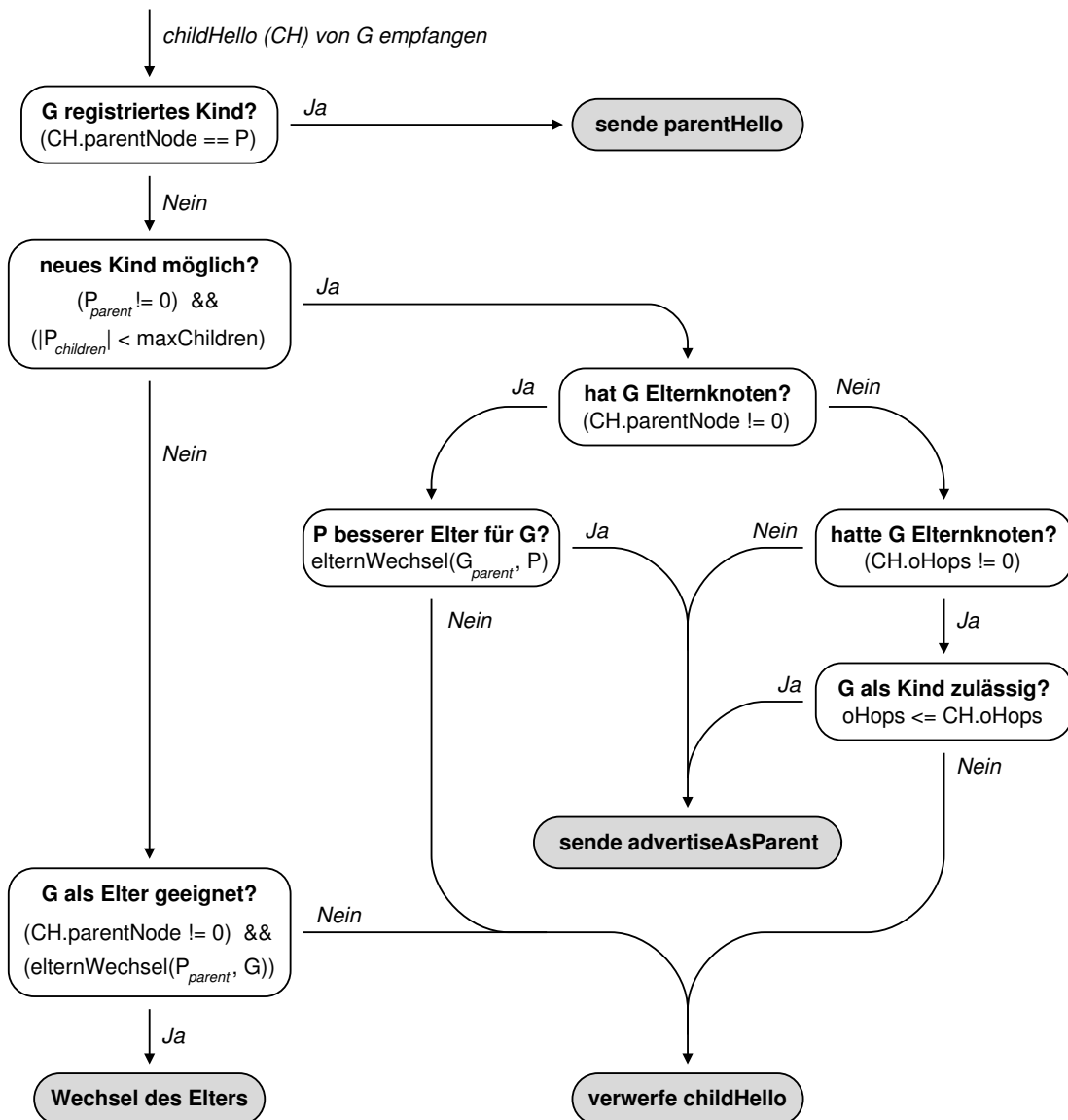


Abbildung 4.4 Verarbeitung eines empfangenen childHello-Pakets.

– Soll sich P als Elternknoten bei G bewerben? Ist G nicht als Kindknoten bei P angemeldet überprüft P , ob es in der Lage ist, einen zusätzlichen Kindknoten aufzunehmen. Ein Grund, welcher gegen die Aufnahme eines neuen Kindknotens sprechen kann, ist eine Beschränkung der pro Elternknoten erlaubten Kindknoten auf eine bestimmte Anzahl maxChildren , um den Aufwand zur Duplizierung und Weiterleitung von Datenpaketen zu begrenzen. Ein weiterer Grund, keinen weiteren Kindknoten aufzunehmen, kann die Tatsache sein, dass P derzeit selbst über keinen Elternknoten (evtl. aufgrund dessen Ausfalls) verfügt, d.h. $G_{\text{parent}} = 0$ gilt. Kann P einen zusätzlichen Kindknoten aufnehmen, so prüft P ob es sich bei G um ein bereits beigetretenes Gruppenmitglied handelt, d.h. ob

$CH.parentNode \neq 0$ gilt. Ist dies der Fall, so verfügt G über einen gültigen Elternknoten G_{parent} . Anhand dessen in dem `childHello`-Paket enthaltenen Verbindungsbewertung kann P mittels Algorithmus 3 bestimmen, ob P aus Sicht von G einen besseren Elternknoten als G_{parent} darstellt, d.h. ob $elternWechsel(G_{parent}, P) == true$ gilt¹³. Ist dies der Fall, bewirbt sich P anhand eines `advertiseAsParent`-Pakets als Elternknoten bei G . Hierfür bestimmt P aus dem Feld `TTL` des empfangenen `childHello`-Pakets und dessen verbleibender Lebenszeit zunächst die Anzahl der Übertragungsabschnitte zu G . Diese wird zusammen mit P s Wurzelpfadlänge, welche sich aus der um 1 erhöhten Wurzelpfadlänge des Elternknotens P_{parent} ergibt, in den entsprechenden Feldern des `advertiseAsParent`-Pakets eingetragen. Gilt jedoch $elternWechsel(G_{parent}, P) == false$, so kann P keinen besseren Zugang zu dem Overlay bieten, so dass das `childHello`-Paket verworfen wird. Hat P in G ein noch nicht beigetretenes Gruppenmitglied erkannt, d.h. $CH.parentNode = 0$ festgestellt, so prüft P ob G zuvor bereits über einen gültigen Elternknoten verfügt hat. Derartige Situationen ergeben sich im Zusammenhang mit dem Ausfall von Gruppenmitgliedern und den damit verbundenen Reparaturarbeiten innerhalb der Overlay-Topologie. Hat G bereits über einen Elternknoten verfügt, so kann dies von P durch das Prüfen von $CH.oHops \neq 0$ festgestellt werden. Schlägt der Vergleich fehl, handelt es sich bei G um ein gänzlich neu beitretendes Gruppenmitglied, so dass sich P bei diesem anhand eines `advertiseAsParent`-Pakets als Elternknoten bewirbt. War G dem TrAM-Baum bereits beigetreten, ist, wie in Abschnitt 4.1.8 verdeutlicht werden wird, in dem Paketfeld `oHops` die von G s ehemaligem Elternknoten zuletzt bekannte Wurzelpfadlänge vermerkt. Um bei der Wiedereingliederung von G in den TrAM-Baum die Bildung von Zyklen zu verhindern, bewirbt sich P nur dann als Elternknoten bei G , falls P s Wurzelpfad höchstens so viele Overlay-Hops umfasst wie G s ehemaliger Elternknoten.

– *Soll sich P als Kindknoten bei G bewerben?* Kann P keine weiteren Kindknoten aufnehmen, wird in einem letzten Schritt geprüft, ob G einen besseren Anknüpfungspunkt an das Overlay bietet, als P s derzeitiger Elternknoten P_{parent} es tut, d.h. ob $elternWechsel(P_{parent}, G) == true$ gilt. Diese Betrachtung ist möglich, da durch den Empfang des `childHello`-Pakets von G P über alle erforderlichen Abstandsmessungen verfügt: So ist, analog zu oben, die Anzahl der Übertragungsabschnitte zu G aus dem Paketfeld `CH.TTL` und der verbleibenden Paketlebenszeit bestimmbar, während sich die Länge des Wurzelpfades über G aus $CH.oHops + 1$ ergibt. Besteht für P über G eine bessere Overlay-Verbindung als über P_{parent} , so kann P durch den Wechsel seines Elternknotens eine Verbesserung seiner Anbindung an den TrAM-Baum erreichen. Diese wird gemäß dem in Abschnitt 4.1.6 beschriebenen Verfahren umgesetzt. Birgt jedoch G

¹³Bei der Berechnung von $elternWechsel(G_{parent}, P)$ ist darauf zu achten, dass P sich aus Sicht von G selbst als Elternknoten betrachtet. Das heißt $P.rHops$ nimmt die Anzahl der Übertragungsabschnitte zwischen P und G an, welche bei Empfang des `childHello`-Pakets aus dessen verbleibender Lebenszeit und dem Paketfeld `TTL` bestimmt wird.

aus Sicht von P kein Verbesserungspotenzial, wird das `childHello`-Paket verworfen und bleibt somit unbeantwortet.

In Abhängigkeit der Lebenszeit des versendeten `childHello`-Pakets bzw. der Größe des gewählten Suchbereichs, erhält das suchende Gruppenmitglied G eine bestimmte Menge \mathcal{S} von Antworten. Diese Menge umfasst in der Regel einerseits das von dem aktuellen Elternknoten als Antwort auf das `childHello` versendete `parentHello`-Paket. Andererseits können `advertiseAsParent`-Pakete umliegender, potenzieller Elternknoten eingetroffen sein. Gemäß dem Zustandsübergangsdiagramm aus Abbildung 4.3 vollzieht G bis zu drei Fallunterscheidungen:

- *Wurden Antworten erhalten?* Zunächst wird überprüft, ob tatsächlich Antworten eingetroffen sind ($|\mathcal{S}| > 0$). Ist dies nicht der Fall, wird der Suchbereich für Elternknoten vergrößert ($i++$) und bei der nächsten Suche eine entsprechend höhere Lebenszeit des `childHello`-Pakets verwendet: Handelt es sich bei G um ein neu beitretendes Gruppenmitglied, so erfolgt die erneute Suche sofort, andernfalls erst nach dem vorgegebenen Zeitintervall $t_{refresh}$.

- *Wurde eine Antwort von dem aktuellen Elternknoten erhalten?* Sind Antworten eingetroffen wird überprüft, ob sich darunter das von dem eventuell bestehenden Elternknoten erwartete `parentHello`-Paket befindet. Ist dies nicht der Fall, wird der Suchbereich vergrößert ($i++$), um den Elternknoten künftig erneut zu erreichen. Ist das `parentHello`-Paket eingetroffen, wird die eventuelle Suchbereichsvergrößerung zurückgesetzt ($i := 0$) und die aktuellen Abstandsdaten ($rHops, oHops$) für G_{parent} übernommen.

- *Soll ein Wechsel des Elternknotens erfolgen?* Letztendlich wird anhand von Algorithmus 3 für jeden potenziellen Elternknoten $P \in \mathcal{S}$ ermittelt, ob dieser einen besseren Elternknoten als der bestehende darstellt. Wird ein solcher Elternknoten gefunden, kommt es gemäß dem in Abschnitt 4.1.6 beschriebenen Verfahren zu einem Wechsel des Elternknotens.

Für den Fall, dass das Gruppenmitglied G der Multicast-Gruppe beitrifft, wird anhand von Algorithmus 2 durch Vergleich aller Elternknoten $E \in \mathcal{S}$ der beste Elternknoten ermittelt und gemäß Abschnitt 4.1.4 eine Verbindung zu diesem hergestellt.

Bei der Vergrößerung des Suchbereichs im Falle ausbleibender Antworten kann für das Gruppenmitglied G der Fall eintreten, dass eine zuvor festgelegte, maximale Anzahl $maxIteration$ von Vergrößerungsiterationen erreicht wird. Tritt dies ein, so überprüft G ob zuletzt ein gültiger Elternknoten bestand ($G_{parent} \neq 0$). Ist dies der Fall, wird der Elternknoten als ausgefallen angenommen und der in Abschnitt 4.1.8 beschriebene Mechanismen zur Lokalisierung eines neuen Elternknotens angestoßen. Bestand für G hingegen kein Elternknoten, so wird angenommen, dass die Gruppe, zu welcher der Beitritt bewerkstelligt werden sollte, nicht existent ist. Als Folge dessen gründet G die

Gruppe und übernimmt hierbei eine Wurzelfunktion. Insbesondere im Zusammenhang mit einer Partitionierung des drahtlosen Ad-hoc-Netzes kann es dabei zum Aufbau mehrerer TrAM-Bäume kommen.

4.1.4 Die Anmeldung bei einem Elternknoten

Hat ein suchendes Gruppenmitglied G von einem Gruppenmitglied P eine Bewerbung als Elternknoten (d.h. ein `advertiseAsParent`-Paket) erhalten und diesen daraufhin als besten Elternknoten erkannt, so versucht G eine Verbindung zu P aufzubauen. Der hierfür eingesetzte Prozess ist unabhängig davon, ob es sich bei G um ein beitretendes oder ein bereits beigetretenes Gruppenmitglied handelt. Der Verbindungsaufbau erfolgt, indem G ein Paket des Typs `childConnect` an P sendet. Dieses enthält die über das `advertiseAsParent`-Paket zuvor erhaltene Verbindungsbewertung von P (Paketfelder `rHops` und `oHops`), sowie das in Abschnitt 4.1.7 erläuterte Feld `tDepth`. Trifft das Paket bei Gruppenmitglied P ein, prüft es zunächst, ob es ein weiteres Gruppenmitglied in die Menge seiner Kindknoten aufnehmen kann. Obwohl P diese Überprüfung bereits vor seiner Bewerbung als Elternknoten bei G durchgeführt hat, ist sie an dieser Stelle erneut erforderlich, da es im Zusammenhang mit zügig aufeinander folgenden Gruppenbeitritten zu konkurrierenden Bewerbungen bei der Aufnahme als Kindknoten kommen kann. Kann P keine weiteren Kindknoten aufnehmen, wird dies durch das Versenden eines `connectDeny`-Pakets an G signalisiert. Ist P hingegen in der Lage, einen zusätzlichen Kindknoten aufzunehmen, wird G in die Menge P_{children} der Kindknoten eingefügt und die erfolgreiche Registrierung durch Versenden eines `parentHello`-Pakets an G bekannt gegeben. Das Paket umfasst P s aktuelle Verbindungsbewertung aus Sicht von G , wobei der Wert für das Paketfeld `rHops` aus dem zuvor erhaltenen `childConnect`-Paket übernommen wird. Bei Erhalt der Registrierungsbestätigung erkennt G das Gruppenmitglied P als aktuellen Elternknoten und setzt daraufhin sowohl $G_{\text{parent}} = P$ als auch den Suchbereich für `childHello`-Pakete zurück. Bleibt aus Sicht von G eine Antwort von Gruppenmitglied P aus, d.h. es wird innerhalb einer festgelegten Zeitspanne weder ein Paket des Typs `connectDeny` noch des Typs `parentHello` empfangen, so erfolgen bis zu 5 Sendewiederholungen des `childConnect`-Pakets. Trifft auch nach der letzten Sendewiederholung keine Antwort ein, geht G von einem Ausfall bzw. einer Unerreichbarkeit von P aus und stellt den Verbindungsaufbau ein. Sofern vorhanden, verharrt G bei seinem bisherigen Elternknoten G_{parent} . Besteht kein solcher, wird der Gruppenbeitritt mit der Suche nach Elternknoten fortgesetzt.

Nachdem sich G bei Gruppenmitglied P als Kindknoten angemeldet hat, erfolgt eine periodische Aktualisierung der Verbindungsbewertung. Durch das Fluten der von G ausgesendeten `childHello`-Pakete wird die aktuelle Anzahl von Übertragungsabschnitten zwischen G und P ermittelt. Der so bestimmte Abstand wird von P innerhalb des `parentHello`-Pakets zusammen mit einem möglicherweise aktualisierten Overlay-Hop-Ab-

stand von P zum Wurzelknoten¹⁴ an G zurück gesendet. Bei Erhalt des Pakets passt Gruppenmitglied G gegebenenfalls seinen Suchbereich für das nächste geflutete **childHello**-Paket an. Die Verbindung zwischen beiden Gruppenmitgliedern bleibt bis zur Abmeldung von G bei P bestehen oder bis eines der Gruppenmitglieder den Ausfall des anderen feststellt. Die beiden in diesem Zusammenhang eingesetzten Verfahren sind Gegenstand der nächsten Abschnitte.

Sowohl die Anmeldungen von Kindknoten bei Elternknoten als auch das Vorhandensein eines gültigen Elternknotens werden in den jeweiligen Gruppenmitgliedern als *Soft-State* gespeichert. Sie bedürfen deshalb einer periodischen Auffrischung. Aus Sicht eines Elternknotens E erfolgt diese Auffrischung durch den Empfang eines **childHello**-Pakets des entsprechenden Kindknotens, bei welchem das Paketfeld **parentNode** die Adresse von E trägt. Aus Sicht eines Kindknotens K wiederum erfolgt die Auffrischung durch den Empfang des nach einem ausgesendeten **childHello** von dem aktuellen Elternknoten K_{parent} erwarteten **parentHello**-Pakets. Bleibt eine Auffrischung für ein bestimmtes, festgesetztes Zeitintervall aus, so geht TrAM von einem Geräteausfall aus. Auf deren Behandlung wird in Abschnitt 4.1.8 eingegangen.

4.1.5 Die Abmeldung bei einem Elternknoten

Im Zuge seines Gruppenaustritts oder einer Umstrukturierung der Overlay-Topologie, kann sich ein Gruppenmitglied G von seinem Elternknoten G_{parent} abmelden. Dabei richtet G an diesen ein Paket des Typs **childDisconnect** und entfernt G_{parent} als Elternknoten. Empfängt G_{parent} das Paket, wird G aus der Menge der Kindknoten entfernt. Die Abmeldung wird von G_{parent} nicht bestätigt. Geht das **disconnectRequest**-Paket bei seiner Vermittlung innerhalb des Ad-hoc-Netzes verloren, bleibt G also aus Sicht von G_{parent} zunächst als Kindknoten bestehen. Jedoch erfolgt von G keine weitere Auffrischung seiner Anmeldung als Kindknoten bei G_{parent} , so dass dieser G automatisch aus der Liste seiner Kindknoten entfernt.

4.1.6 Der Wechsel des Elternknotens

Besteht eine Verbindung zwischen einem Gruppenmitglied G und dessen Elternknoten G_{parent} , so kann G nach dem Fluten eines **childHello**-Pakets die Bewerbung eines potenziellen Elternknotens P erhalten. In einem solchen Fall vergleicht G die über die Bewerbung erhaltene Verbindungsbewertung von P mit der von G_{parent} . Stellt sich dabei heraus, dass P eine bessere Verbindung bietet als G_{parent} , so kann G durch den Wechsel seines Elternknotens eine Verbesserung der Overlay-Topologie erreichen. Das hierfür eingesetzte Verfahren entspricht im Wesentlichen einer Kombination der in Abschnitt

¹⁴Die Länge des Wurzelfades von P kann sich ändern, falls P selbst seinen Elternknoten wechselt.

4.1.4 beschriebenen An- und der in Abschnitt 4.1.5 erläuterten Abmeldung bei Elternknoten. Der Gesamtprozess des Elternknotenwechsels ist in Abbildung 4.5 dargestellt und kann wie folgt zusammengefasst werden:

1. G flutet eine Suchanfrage in Form von `childHello`-Paketen im gewählten Suchbereich. Die Lebenszeit des Pakets ergibt sich aus $rHops + 2^i$, mit $rHops$ als der zuletzt bekannten Anzahl an Übertragungsabschnitten zwischen G und G_{parent} und i als der Anzahl zuvor ausgebliebener `parentHello`-Pakete von G_{parent} . In dem gezeigten Beispiel gilt $i = 0$, so dass der Suchbereich einen Übertragungsabschnitt über G_{parent} hinausreicht.
2. Gruppenmitglied P, welches im Vergleich zu G_{parent} die gleiche Wurzelpfadlänge aufweist aber über weniger Übertragungsabschnitte erreichbar ist, erkennt, dass es G einen besseren Anknüpfungspunkt an das Overlay bereitstellen kann als G_{parent} . Daraufhin bewirbt sich P mittels eines `advertiseAsParent`-Pakets, welches Ps aktuelle Verbindungsbewertung enthält, bei G.
3. G empfängt die Bewerbung von P, vergleicht die enthaltene Verbindungsbewertung mit der von G_{parent} und erkennt P als besseren Elternknoten. Als Folge dessen bewirbt sich G bei P anhand eines `childConnect`-Pakets als Kindknoten.
4. P nimmt die Bewerbung von G als Kindknoten an, fügt G zu der Menge $P_{children}$ hinzu und bestätigt die Anmeldung anhand eines `parentHello`-Pakets.
5. Nach Bestätigung der neuen Verbindung durch P meldet sich G bei seinem bisherigen Elternknoten G_{parent} mittels eines `childDisconnect`-Pakets ab und setzt $G_{parent} = P$. Diese Abmeldung wird von G_{parent} nicht quittiert.

Gelingt G der Verbindungsaufbau zu dem neuen Elternknoten P nicht, so verharrt G bei seinem zuletzt bekannten Elternknoten G_{parent} . Während des gesamten Prozesses des Elternwechsels bleibt also, auch im Falle eines fehlgeschlagenen Elternwechsels, die Verbindung von G zum TrAM-Baum bestehen.

Im Fall von Paketverlusten kann es im Zusammenhang mit dem Wechsel des Elternknotens zu einer Zyklenbildung kommen. Diese Situation tritt genau dann ein, wenn 1. die Anmeldung aus Sicht des neuen Elternknotens gelingt, dessen Anmeldebestätigung aber verloren geht *oder* 2. die Anmeldung bei dem neuen Elternknoten gelingt, die Abmeldung bei dem bisherigen Elternknoten aber nicht erfolgreich ist. Während in beiden Fällen eine Verbindung zu sowohl dem bisherigen als auch dem neuen Elternknoten besteht, wird die Situation automatisch aufgelöst: In der Tat frischen die ausgesendeten `childHello`-Pakete die Anmeldung als Kindknoten nur bei einem der beiden Elternknoten auf. Wie in Abschnitt 4.1.8 verdeutlicht, wird als Folge dessen der übrige Elternknoten den betroffenen Kindknoten automatisch aus seiner Kindknotenliste streichen.

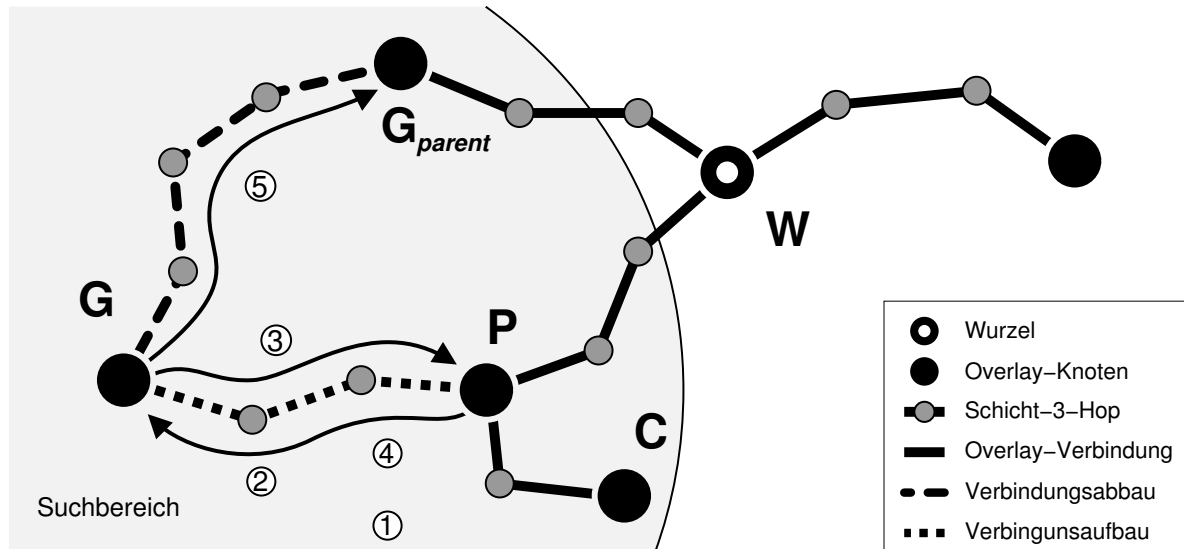


Abbildung 4.5 Anpassung der Overlay-Topologie durch Wechsel des Elternknotens aufgrund einer besseren Verbindungsbewertung.

4.1.7 Das Verschieben der Wurzel des TrAM-Baums

Je nach Aufenthaltsort der Gruppenmitglieder innerhalb des Netzes sowie der Reihenfolge des Gruppenbeitritts kann im Laufe seines Bestehens der TrAM-Baum eine kettenförmige Struktur annehmen. Um eine derartige Degenerierung der Topologie zu vermeiden, erlaubt TrAM das Verschieben des ursprünglichen Wurzelknotens.

Um dies zu bewerkstelligen, benötigt der Wurzelknoten Kenntnis über die Tiefe der einzelnen Teilbäume. Diese Information wird daher in den zwischen Kind- und Elternknoten periodisch ausgetauschten **childHello**-Paketen innerhalb des Felds **tDepth** (*Tree-Depth, Baumtiefe*) kodiert. Handelt es sich bei einem sendenden Gruppenmitglied um einen Blattknoten (d.h. ein Gruppenmitglied ohne eigene Kindknoten), vermerkt dieser innerhalb des Felds die Baumtiefe 0. Gruppenmitglieder, welche selbst als Elternknoten fungieren, sammeln die von all ihren Kindknoten zuletzt erhaltenen Baumtiefen, bilden hieraus das Maximum und vermerken das um eins inkrementierte Ergebnis in den eigens ausgesendeten **childHello**-Paketen. Hierdurch propagieren sich die Baumtiefen zur Wurzel, welche somit das Verschieben der Wurzelfunktion veranlassen kann.

Das Verschieben der Wurzelfunktion wird angestoßen, sobald ein an der Wurzel hängender Teilbaum um mindestens zwei Stufen tiefer als alle übrigen Teilbäume ist. In diesem Fall wird die Wurzelfunktion an jenen Kindknoten übertragen, von welchem die größte Baumtiefe empfangen wurde. Die Funktionsübertragung erfolgt durch Versenden eines **rootRedirect**-Pakets an den ausgewählten Kindknoten. Dieser signalisiert dem Wurzelknoten sein Einverständnis zur Funktionsübernahme durch Bestätigung anhand eines Pakets des Typs **rootRedirectAccept** bzw. seine Ablehnung der Funktionsübernah-

me anhand eines Pakets des Typs `rootRedirectDeny`. Gründe, die für die Ablehnung der Wurzelfunktion sprechen, können beispielsweise die Tatsache sein, dass sich das Gruppenmitglied aufgrund eines Gruppenaustritts aus dem Overlay-Netz zurückzieht.

4.1.8 Ausfälle von Gruppenmitgliedern

Bleiben aus Sicht eines Elternknotens E $n_{failure}$ aufeinander folgende, erwartete `childHello`-Pakete von Seiten eines Kindknotens $K \in E_{children}$ aus, geht E von einem Ausfall von K aus. K wird daraufhin aus der Menge $E_{children}$ der Kindknoten gestrichen.

Bleiben aus Sicht eines Kindknotens K die `parentHello`-Pakete des Elternknotens K_{parent} aus, wird gemäß Abschnitt 4.1.3 zunächst die Lebenszeit der `childHello`-Pakete bzw. der Suchbereich zur Kontaktierung des Elternknotens vergrößert. Ist jedoch auch nach insgesamt $n_{failure}$ versendeten `childHello`-Paketen keine Antwort von K_{parent} eingetroffen, geht K von einem Ausfall seines Elternknotens aus. Zu diesem Zeitpunkt verfügt K über keinen gültigen Elternknoten mehr ($K_{parent} = 0$) und stößt daher die Suche nach einem neuen an. Hierfür wird eine Abwandlung des aus Abschnitt 4.1.3 bekannten Verfahrens zur Bestimmung von Elternknoten verwendet: Im so versendeten `childHello`-Paket wird das Adressfeld des (ausgefallenen) Elternknotens ebenso wie das Feld `rHops` auf 0 gesetzt. Das Feld `oHops` nimmt jedoch weiterhin die zuletzt bekannte Länge des Wurzelfpades des bisherigen Elternknotens an. Auf diese Weise kann ein das `childHello` empfangende Gruppenmitglied G feststellen, dass es sich bei K um ein Gruppenmitglied handelt, welches aufgrund eines Geräteausfalls einen neuen Elternknoten zu bestimmen versucht. Gemäß Abbildung 4.4 bewirbt sich G in einem solchen Fall nur dann als Elternknoten bei K , falls die eigene Wurzelfadlänge die von K s ehemaligem Elternknoten nicht übersteigt. Hierdurch kann eine Bildung von Zyklen effektiv vermieden werden: Ein Gruppenmitglied, welches einen neuen Elternknoten zu bestimmen versucht, wird nie eine Bewerbung von einem im gleichen Teilbaum sich tiefer befindenden Gruppenmitglied erhalten, da ein solches offensichtlich eine höhere Wurzelfadlänge besitzt und es daher unterlässt, sich bei K als Elternknoten zu bewerben.

Die in Form von `advertiseAsParent`-Paketen versendeten Bewerbungen potenzieller Elternknoten umfassen analog zu den in Abschnitt 4.1.3 verwendeten Antworten eine eigene Verbindungsbewertung (Felder `rHops` und `oHops`). Sie werden von dem suchenden Kindknoten über einen bestimmten Zeitraum hinweg gesammelt, um so aus einer Auswahl infrage kommender Elternknoten den besten ermitteln zu können. Bei diesem erfolgt anschließend die in Abschnitt 4.1.4 beschriebene Anmeldung.

Der Ausfall des Wurzelknotens W stellt einen Sonderfall dar, welcher durch eine Abwandlung des Mechanismus zur Elternknotenbestimmung aufgelöst wird. Erkennt ein Gruppenmitglied $G \in W_{children}$ aufgrund ausbleibender `parentHello`-Pakete den Ausfall

des Wurzelknotens W^{15} , so unterbricht G das Versenden von `childHello`-Paketen. Stattdessen flutet G das Netz periodisch mit `detectRoot`-Paketen, wobei die Periodizität derjenigen der `childHello`-Pakete entspricht. Die Lebenszeit der Pakete wird von dem zuletzt versendeten `childHello` übernommen und in dem Paketfeld `TTL` vermerkt.

Gruppenmitglieder, deren Elternknoten eine Wurzelfadlänge > 0 aufweisen, verwerfen empfangene `detectRoot`-Pakete. Empfängt jedoch ein Gruppenmitglied G' , dessen Elternknoten eine Wurzelfadlänge von 0 aufweist (und welches demnach ein Kindknoten des Wurzelknotens ist), ein `detectRoot`-Paket, so sind vier Fälle möglich:

1. Hat G' ebenfalls den Ausfall des Wurzelknotens *bemerkt*, bewirbt sich G' mittels eines `advertiseAsParent`-Pakets bei G als Elternknoten genau dann wenn die IP-Adresse von G' numerisch *größer* als die von G ist. Der Wert des Paketfelds `rHops` nimmt dabei die Anzahl der Übertragungsabschnitte zu G an, welche durch das `TTL`-Feld des empfangenen `detectRoot`-Pakets sowie dessen verbleibender Lebenszeit berechenbar ist. Das Paketfeld `oHops` wird auf 0 gesetzt.
2. Hat G' ebenfalls den Ausfall des Wurzelknotens *bemerkt*, sieht G' in G einen neuen Elternknoten genau dann wenn die IP-Adresse von G' numerisch *kleiner* als die von G ist. In diesem Fall versucht G' gemäß des in Abschnitt 4.1.4 beschriebenen Verfahrens eine Verbindung zu G herzustellen. Der Wert des Paketfelds `rHops` nimmt dabei die Anzahl der Übertragungsabschnitte zu G an. Das Paketfeld `oHops` wird auf 0 gesetzt, während die zuletzt bekannte, eigene Baumtiefe im Paketfeld `tDepth` vermerkt wird.
3. Hat G' den Ausfall des Wurzelknotens *nicht bemerkt* und das zuletzt erwartete Paket des Typs `parentHello` von der Wurzel *erhalten*, bewirbt sich G' analog zu dem in Abschnitt 4.1.3 beschriebenen Verfahren als Elternknoten bei G .
4. Hat G' den Ausfall des Wurzelknotens noch *nicht bemerkt* aber bereits ein oder mehrere zuletzt erwartete Pakete des Typs `parentHello` von der Wurzel *nicht erhalten*, verwirft G' das von G erhaltene `detectRoot`-Paket, da keine Gewissheit darüber besteht, dass der Wurzelknoten noch in Betrieb ist.

Für G ergeben sich an dieser Stelle zwei Möglichkeiten:

1. Empfängt G von einem anderen Gruppenmitglied G' eine Bewerbung als *Elternknoten*, so verbindet sich G zu diesem mittels des Verfahrens aus Abschnitt 4.1.4. Gelingt der Verbindungsaufbau so verfügt G über einen gültigen Elternknoten und

¹⁵Obwohl die Wurzel W des TrAM-Baums den Gruppenmitgliedern nicht explizit bekannt ist, ist sie ihren Kindknoten W_{children} jedoch aufgrund ihrer Wurzelfadlänge von 0 implizit bekannt.

nimmt das Fluten von `childHello`-Paketen wieder auf. Schlägt der Verbindungsaufbau fehl fährt G mit dem Fluten von `detectRoot`-Paketen fort.

2. Empfängt G von einem anderen Gruppenmitglied G' eine Bewerbung als *Kindknoten*, so nimmt G diese in jedem Falle an, auch wenn dabei die maximal zulässige Anzahl an Kindknoten überschritten wird. Im Anschluss fährt G mit dem Fluten von `detectRoot`-Paketen fort.

Aufgrund des beschriebenen Verfahrens bildet sich unter der Menge der Gruppenmitglieder $G \in W_{\text{children}}$ eine neue Baumstruktur aus. An dessen Spitze liegt ein neuer, eindeutiger Wurzelknoten W' , für welchen gilt:

$$\left((W' \in W_{\text{children}}) \wedge (\forall G \in W_{\text{children}} : (G \neq W') \Rightarrow (|G| < |W'|)) \right)$$

W' ist daher das einzige Gruppenmitglied in W_{children} , welches Bewerbungen nur von Kind- aber nicht von potenziellen Elternknoten empfängt. Die Bestimmung eines neuen Elternknotens schlägt bei W' also fehl, so dass sich W' nach insgesamt n_{failure} ausgesendeten, aber ohne Antwort gebliebenen `detectRoot`-Paketen selbst als neue Wurzel deklariert und deren Funktion übernimmt.

4.1.9 Gruppenaustritt

Wünscht ein Gruppenmitglied G der Gruppe auszutreten bzw. sich aus dem TrAM-Baum auszugliedern so bedeutet dies den Verbindungsabbau zu dem Elternknoten G_{parent} einerseits und zu den Kindknoten $K \in G_{\text{children}}$ andererseits.

Damit die Kindknoten auch während G s Ausgliederung eine Verbindung zu dem TrAM-Baum halten können, erfolgt der Abbau der Verbindungen zu ihnen als erstes. Hierfür wählt G einen Kindknoten $K \in G_{\text{children}}$ und sendet diesem ein Paket des Typs `releaseRequest` zu. Bei Erhalt des Pakets startet K die Suche nach einem neuen Elternknoten. Hierfür wird das in Abschnitt 4.1.8 verwendete Verfahren zur Wiederherstellung der Verbindung mit dem TrAM-Baum nach Ausfall des bisherigen Elternknotens eingesetzt. Ist K die Verbindung zu einem neuen Elternknoten gelungen, bestätigt es den von G gewünschten Verbindungsabbau: Dies erfolgt indem K ein Paket des Typs `releaseAccept` an G sendet. Bei dessen Empfang entfernt G zunächst das Gruppenmitglied K aus der Menge der Kindknoten. Dann prüft G das Vorhandensein weiterer Kindknoten und wiederholt gegebenenfalls das soeben beschriebene Verfahren.

Der Verbindungsabbau zu Kindknoten erfolgt demnach nicht parallel, sondern der Reihe nach. Dies ergibt sich aufgrund der Möglichkeit, dass nach dem Abbau der Verbindung zu einem Kindknoten dieser als potenzieller Elternknoten für weitere, räumlich nahe gelegene Kindknoten fungieren kann, deren Verbindungen noch abzubauen sind. Wurde

die Verbindung zu allen Kindknoten erfolgreich abgebaut, meldet sich G zuletzt bei seinem Elternknoten G_{parent} ab. Dies erfolgt mittels des in Abschnitt 4.1.5 beschriebenen Verfahrens.

Im Zusammenhang mit dem Austritt eines Gruppenmitglieds G ergeben sich einige Besonderheiten bzw. Sonderfälle:

- Während seiner Ausgliederung verarbeitet G Pakete des Typs `childHello` auf vereinfachte Weise: Handelt es sich bei dem Sender des `childHello`-Pakets um einen Kindknoten, zu welchem eine Verbindung noch besteht und nicht gerade abgebaut wird, so wird dessen Anmeldung wie gewohnt anhand eines `parentHello`-Pakets bestätigt. Ansonsten wird das `childHello`-Paket jedoch verworfen. Einerseits kann hierdurch vermieden werden, dass sich G als Elternknoten bei Gruppenmitgliedern bewirbt, zu welchen eine eventuell im Anschluss aufgebaute Verbindung wenig später wieder abgebaut werden müsste. Ebenso wird verhindert, dass sich G als Elternknoten bei eigenen Kindknoten bewirbt, welche gerade nach einem neuen Elternknoten suchen.

- Handelt es sich bei G um die Wurzel des TrAM-Baums, so kann G seine Ausgliederung nicht auf Anrieb bewerkstelligen. Stattdessen muss G zuvor seine Wurzelfunktion gemäß Abschnitt 4.1.7 an einen Kindknoten delegieren. Erst dann kann G mit seiner Ausgliederung aus dem TrAM-Baum beginnen.

4.2 Das Routing

Aufgrund der baumförmigen Overlay-Topologie ist das Overlay-Routing bei TrAM implizit vorgegeben und bedarf keines weiteren Austauschs von Routing-Information. Agiert ein Gruppenmitglied G selbst als Sender, so leitet es ein zu versendendes Datenpaket an alle im TrAM-Baum benachbarten Gruppenmitglieder

$$E \in G_{\text{children}} \cup \{G_{\text{parent}}\}$$

weiter. Diese leiten das empfangene Datenpaket wiederum an alle im TrAM-Baum benachbarten Gruppenmitglieder

$$E' \in E_{\text{children}} \cup \{E_{\text{parent}}\} \setminus \{G\}$$

mit Ausnahme desjenigen Gruppenmitglieds G , von welchem das Paket empfangen wurde, weiter.

4.2.1 Sonderfälle

Im Zusammenhang mit der Umstrukturierung des TrAM-Baums im Laufe der Gruppenkommunikation ergeben sich einige Sonderfälle bzw. nennenswerte Situationen bei der Weiterleitung von Datenpaketen innerhalb des Overlay.

– Hat ein Gruppenmitglied G mit seiner Ausgliederung aus dem Overlay begonnen, so leitet es währenddessen Datenpakete innerhalb des Overlay in gewohnter Weise weiter. G leitet Datenpakete demnach auch an einen Kindknoten K weiter, zu welchem die Verbindung gerade abgebaut wird, d.h. zu welchem gerade ein `releaseRequest`-Paket gesendet wurde. Erst bei Empfang des zugehörigen `releaseAccept`-Pakets wird K aus G_{children} entfernt und nicht länger als Nachbar betrachtet.

– Fällt ein Gruppenmitglied, welches nicht der Wurzelknoten ist, aus, so zerfällt der TrAM-Baum vorübergehend in eventuell mehrere Teilbäume. In dieser Zeit werden die in einem Teilbaum ausgesendeten Datenpakete auch nur in diesem Teilbaum vermittelt. Nach der Reparatur des TrAM-Baums ist es Aufgabe eines Transportprotokolls, diese Datenpakete auch den übrigen Gruppenmitgliedern zuzustellen.

– Bei einem Ausfall des Wurzelknotens W bestimmen die Gruppenmitglieder $G \in W_{\text{children}}$ unter sich einen neuen Wurzelknoten W' . Obgleich sich W' erst nach n_{failure} versendeten `detectRoot`-Paketen als neue Wurzel deklariert und dessen Funktion übernimmt, ist das Routing von Datenpaketen innerhalb des TrAM-Baums bereits zu dem Zeitpunkt wiederhergestellt, zu welchem die Gruppenmitglieder $G \in W_{\text{children}}$ erneut untereinander verbunden sind. Im besten Fall ist hierfür lediglich ein einziges Fluten eines `detectRoot`-Pakets erforderlich. Dieser Fall tritt genau dann ein, wenn der künftige Wurzelknoten W' den Ausfall von W als erstes bemerkt und entsprechend ein `detectRoot`-Paket versendet: In diesem Fall bewerben sich alle übrigen Kindknoten des ehemaligen Wurzelknotens bei W' , so dass nach deren erfolgreichen Verbindungsaufbauten zu W' der TrAM-Baum wiederhergestellt ist.

4.3 Bewertung und Diskussion

In diesem Abschnitt erfolgt eine Bewertung des TrAM-Protokolls sowie die Diskussion dessen wichtigster Eigenschaften. Analog zu Abschnitt 3.4 spiegelt sich die vorgenommene Gliederung in Tabelle 4.1 wider: Sie erlaubt den Vergleich von TrAMs Protokollmechanismen für den Overlay-Aufbau bzw. dessen -Erhalt (①), den Gruppenbeitritt (②), die Vermittlung von Datenpaketen (③), die Ausgliederung von Gruppenmitgliedern aus dem Overlay (④) sowie die Nutzung der Semi-Broadcast-Eigenschaft (⑤) mit den in anderen Protokollen integrierten Mechanismen.

4.3.1 Aufbau und Erhalt der Overlay-Topologie

Die Overlay-Topologie entspricht bei TrAM einer Baumstruktur. Sie verbindet die Gruppenmitglieder anhand einer geringen Anzahl von Overlay-Verbindungen. Die Topologie selbst wird fortlaufend aufgrund einer speziellen Metrik zur Verbindungsbewertung angepasst. Die Baumstruktur erweist sich dabei als äußerst flexibel, da sie anders als

		TrAM	Narada	NICE	PAST-DM	ALMA
①	Topologie	Baum	Mesh	Cluster	Mesh	Baum
	# Overlay-Verbindungen	⊕⊕	⊕	⊖	○	⊕
	Flexibilität	⊕	⊕	⊖	⊕	○
	Optimierungsmetrik	Hops	RTT	RTT	Hops	RTT
	Netzbelastung	⊕⊕	⊕	○	⊖⊖	⊕
②	Initiale Nachbarsuche	ERS	unspez.	RVP	ERS	unspez.
	Eingliederungsaufwand	⊕⊕	○	⊖	⊕	⊕
③	Routing-Information	Implizit	Explizit	Implizit	Explizit	Implizit
	Routing-Konsistenz	⊕	○	⊖	⊖	⊕
	Verteilbaum	Global	Quelle	Quelle*	Quelle	Global
	Verteilbaumstabilität	⊕	○	⊕	⊖	○
④	Gruppenaustritt	⊕⊕	⊕⊕	⊖	⊕	⊕
	Ausfälle	⊕	⊕	⊖⊖	unspez.	unspez.
⑤	Semi-Broadcast-Nutzung	⊕	⊖	⊖	○	⊖

Tabelle 4.1 Bewertung von TrAM im Vergleich zu anderen Protokollen.

etwa bei NICE keinem komplexen und starren Clustering von Gruppenmitgliedern unterliegt. Vielmehr können sich bei TrAM Gruppenmitglieder mit beliebigen anderen Gruppenmitgliedern verbinden, welche gemäß der Verbindungsbewertung eine bessere Anbindung an das Overlay ermöglichen.

Die Bewertung von Verbindungen ist derart entworfen, dass sie einerseits die Bildung von Zyklen innerhalb des TrAM-Baums verhindert. Andererseits kann durch den Gewichtungsfaktor α eine Minimierung der Wurzelpfadlänge oder der Anzahl der Übertragungsabschnitte zum Elternknoten bevorzugt werden: So erfolgt bei der Wahl kleiner α -Werte eine Minimierung der Übertragungsabschnittszahl zum Elternknoten vorrangig gegenüber der Wurzelpfadlänge. Dies ist in Abbildung 4.6 dargestellt, in welcher über der gleichen Gruppenmitgliedsverteilung TrAM-Bäume für verschiedene α -Werte aufgebaut wurden. Es ist erkennbar, dass mit steigendem α auch Elternknoten gewählt werden, welche größere räumliche Abstände aufweisen. Dass in solchen Fällen die Wurzel des TrAM-Baums über weniger Overlay-Hops erreicht werden kann, zeigen die in der Abbildung notierten durchschnittlichen $oHops$ -Werte. Analog hierzu steigt mit wachsendem α die Übertragungsabschnittszahl $rHops$ zum Elternknoten.

Ein Wechsel des Elternknotens kann von einem Gruppenmitglied mit jeder Periode $t_{refresh}$ durchgeführt werden. Die Overlay-Topologie zeigt sich daher als sehr reaktiv gegenüber Veränderungen innerhalb des zugrunde liegenden drahtlosen Ad-hoc-Netzes. Dagegen weisen die zusätzlich in TrAM integrierten Mechanismen zur Verschiebung der Wurzelfunktion eine gewisse Trägheit auf. Diese Trägheit ist dadurch bedingt, dass Änderungen der Baumtiefen zunächst zum Wurzelknoten propagiert werden müssen. Sie steht daher in direktem Zusammenhang zu der Periodizität der `childHello`-Pakete, wel-

4. TrAM: Ein leichtgewichtiges Overlay für drahtlose Ad-hoc-Netze

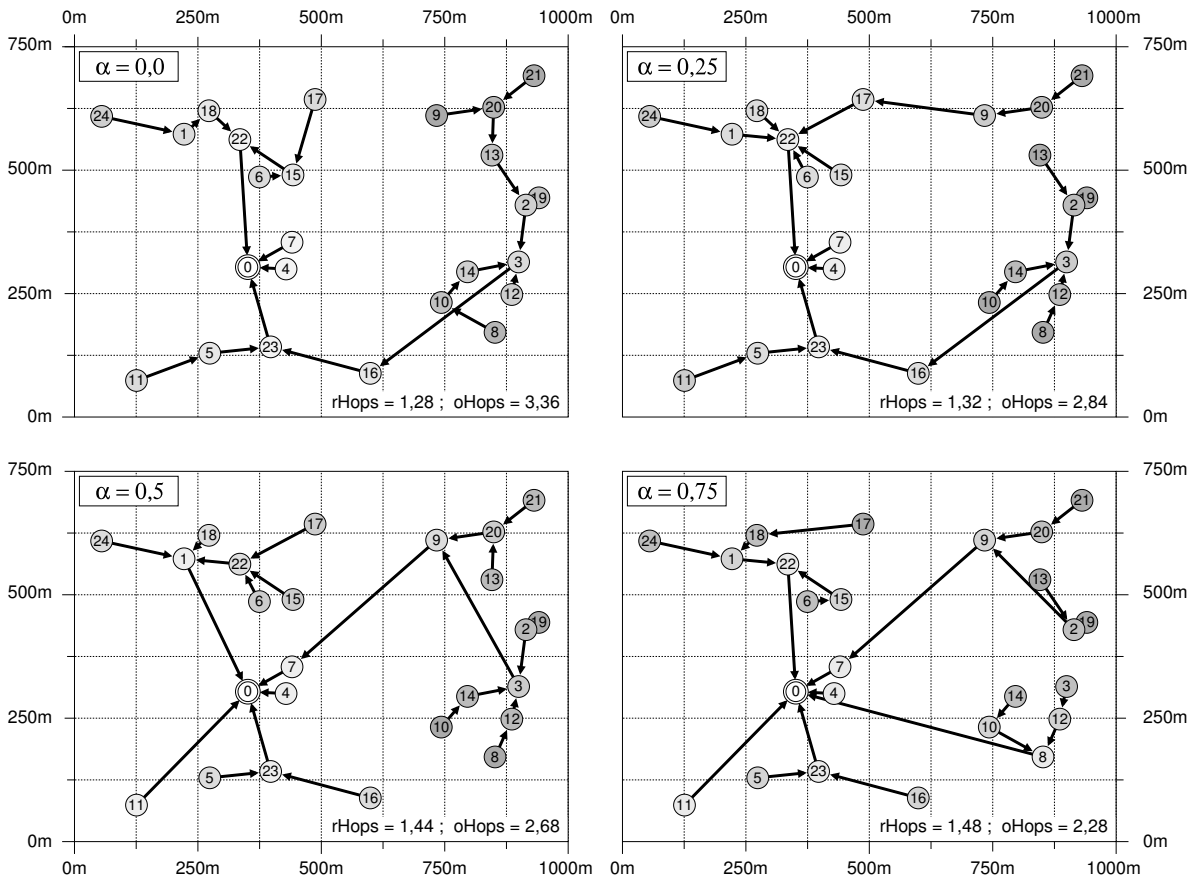


Abbildung 4.6 Auswirkung unterschiedlicher α -Werte auf die Overlay-Topologie bei einer Übertragungsreichweite von $125m$.

che zwischen Kind- und Elternknoten ausgetauscht werden. Werden die Pakete häufig ausgetauscht, können Änderungen der Baumtiefen schnell zu dem Wurzelknoten propagiert und Verschiebungen der Wurzelfunktion entsprechend zügig angestoßen werden. Da *childHello*-Pakete durch ein lokales Fluten des drahtlosen Ad-hoc-Netzes vermittelt werden, bedeutet ein geringerer zeitlicher Abstand jedoch ebenfalls eine höhere Belastung. An dieser Stelle sei angemerkt, dass das Verschieben der Wurzelfunktion kein für die Effizienz kritischer Mechanismus ist, und eine Trägheit daher durchaus in Kauf genommen werden kann.

Der in Abschnitt 4.1.3 beschriebene Mechanismus zum Erhalt der von TrAM aufgebauten Overlay-Topologie zielt auf eine Reduktion des Kontrollflusses unter Berücksichtigung Schichten-übergreifender Aspekte. Die Vorteile der eingesetzten Mechanismen werden dabei insbesondere im Kontext reaktiver Routing-Protokolle deutlich. Dies ergibt sich aufgrund der Tatsache, dass TrAM zur Anpassung der Overlay-Topologie nicht wahllos eine auf dem Austausch von Unicast-Paketen basierende Verbindungsbeurteilung zu anderen Gruppenmitgliedern vornimmt: Bei gängigen Overlay-Topologien

verursacht dieses Vorgehen im Zusammenspiel mit reaktiven Routing-Protokollen den Aufbau zahlreicher Routen bzw. das oftmalige Fluten des Netzes zur Routen-Bestimmung. Dabei kann keine Garantie gegeben werden, dass der Routen-Aufbau bzw. die Bewertung der potenziellen Verbindung im Anschluss auch tatsächlich zum Aufbau der Verbindung und somit zu einer Verbesserung der Overlay-Topologie führt. Folglich erweisen sich Routen-Aufbauten oftmals als nutzlose Belastung des Netzes. TrAM umgeht diesen Nachteil indem es selbst Kontrollpakete zur Lokalisierung von Elternknoten innerhalb des Netzes mit begrenzter Lebenszeit flutet. Durch dieses einzige Fluten kann eine potenziell beliebige Anzahl von Gruppenmitgliedern erreicht und so die Qualität der Verbindung zu ihnen gemessen werden. Aufgrund der in den gefluteten `childHello`-Paketen enthaltenen Information können Gruppenmitglieder, welche die Pakete erhalten, Rückschlüsse ziehen, ob sie als Elternknoten für das suchenden Gruppenmitglied überhaupt infrage kommen bzw. ihm eine bessere Verbindung zu dem TrAM-Overlay als dessen bestehender Elternknoten bieten können. Erst in einem solchen Fall findet ein Austausch von Unicast-Paketen (`advertiseAsParent`) statt, welcher eventuell einen Routen-Aufbau erfordert. Dieser erfolgt dann jedoch nicht umsonst, da über den sich anbietenden Elternknoten auf jeden Fall eine Verbesserung der Topologie erzielt wird.

4.3.2 Der Gruppenbeitritt

Die Eingliederung neuer Gruppenmitglieder in das TrAM-Overlay erweist sich als äußerst einfach zu bewerkstelligen. Dies beruht auf der Tatsache, dass für die Eingliederung lediglich ein einziges bereits beigetretenes Gruppenmitglied ausfindig gemacht werden muss. Dieses erfolgt durch Fluten von Paketen mit exponentiell wachsender Lebenszeit. Hiermit kann einerseits garantiert werden, dass im Falle nahe gelegener Gruppenmitglieder nur ein geringer Netzbereich für deren Bestimmung belastet wird. Andererseits kann sichergestellt werden, dass auch bei Entfernungen von mehreren Übertragungsabschnitten ein entsprechendes Gruppenmitglied zügig lokalisiert werden kann. Um zu diesem eine Verbindung herzustellen und damit die Eingliederung in den TrAM-Baum zu vollziehen, ist lediglich der Austausch von drei Kontrollpaketen (`advertiseAsParent`, `childConnect` und `parentHello`) erforderlich. Im Vergleich zu anderen Overlay-Topologien wird hier also ein bedeutender Gewinn erzielt, da bei diesen üblicherweise die Registrierung des neuen Gruppenmitglieds bei einer größeren Anzahl von bereits beigetretenen Gruppenmitgliedern erfolgen muss.

4.3.3 Die Vermittlung von Datenpaketen

Als Folge seiner baumförmigen Topologie verfügt das TrAM-Overlay über einen globalen Baum zur Vermittlung von Datenpaketen und nicht über Quellen-spezifische Bäume. Einerseits erweist sich dies als vorteilhaft, da aufgrund der Topologie das Routing innerhalb des Overlay implizit vorgegeben ist und demnach keines expliziten Austauschs

von Routing-Information bedarf. Während an dieser Stelle demnach eine Minderung des Kontrollflusses erzielt wird, sind mit der baumförmigen Topologie andererseits auch Nachteile verbunden:

- *Die Lastbündelung:* Bei TrAM erfolgt die Vermittlung der Datenpakete aller als Sender agierenden Gruppenmitglieder auf den gleichen Overlay-Verbindungen, was einer Bündelung der Verkehrslast anstatt deren Verteilung auf das Netz bedeutet. Bei der Paketvermittlung innerhalb der Overlay-Verbindungen ist bei steigender Verkehrslast daher mit Schwierigkeiten bzw. einer steigenden Synchronisationsdauer beim Medienzugriff zu rechnen. An dieser Stelle muss allerdings berücksichtigt werden, dass eine Lastverteilung in drahtlosen Ad-hoc-Netzen als außerordentlich schwierig umzusetzen ist [47, 60]. Dies ergibt sich als direkte Konsequenz aus dem bei IEEE 802.11 erheblichen Erkennungsbereich von Funksignalen, welcher parallele Medienzugriffe auch im Falle von räumlich deutlich entfernten ($\approx 500m$) Geräten oftmals verhindert. Demzufolge ist auch bei anderen Overlay-Topologien, welche über Quellen-spezifische Verteilbäume verfügen, nicht mit deutlichen Verbesserungen zu rechnen.

- *Die ineffizienten Verteilbäume:* Da sich alle Gruppenmitglieder an der Wurzel des Baums ausrichten, weist der von TrAM verwendete Verteilbaum lediglich für den Wurzelknoten einen effizienten Charakter bzw. eine minimale Baumtiefe auf. Agiert ein Gruppenmitglied, welches im TrAM-Baum einen Blattknoten darstellt, als Sender, hat dies daher eine Erhöhung der durchschnittlichen Latenzen zur Folge, da ein Paket im Laufe seiner Vermittlung erst zur Wurzel des TrAM-Baums geleitet werden muss, um dort in die übrigen Teilbäume propagiert werden zu können.

Auch im Laufe von Strukturänderungen innerhalb der Overlay-Topologie hält ein von der Änderung betroffenes Gruppenmitglied durchgehend eine Verbindung zum TrAM-Baum aufrecht. Da die Vermittlung von Datenpaketen implizit durch die Struktur der Overlay-Topologie vorgegeben ist, erfolgt die Aktualisierung der Routing-Information direkt durch die eigentliche Strukturänderung und bedarf keines zusätzlichen Austauschs von Kontrollinformation. Aus diesem Grund weist die Datenvermittlung auf Basis des TrAM-Baums im Vergleich zu Protokollen, die auf dem expliziten Austausch von Routing-Information basieren, eine höhere Konsistenz auf. Bei Datenpaketen, die *im Laufe* einer Strukturänderung über das Overlay vermittelt werden, kann es dennoch vorkommen, dass nicht alle Gruppenmitglieder diese erhalten. Dieser Fall tritt dann ein, wenn sich ein Gruppenmitglied G mit einem neuen Elternknoten N verbindet, welcher sich im Verteilbaum oberhalb des bisherigen Elternknotens B befindet. Erfolgt G s Anmeldung als Kindknoten bei N *nachdem* N ein Datenpaket p vermittelt hat, und erfolgt G s Abmeldung bei B *bevor* dieser das Paket p vermittelt hat, so erhält G das Paket p nicht. Es ist Aufgabe eines Transportprotokolls, derartige Datenpakete nach Abschluss einer Strukturänderung den betroffenen Gruppenmitgliedern zuzustellen.

4.3.4 Die Ausgliederung von Gruppenmitgliedern

Die Ausgliederung eines Gruppenmitglieds G aus der Overlay-Topologie aufgrund dessen gewünschten Gruppenaustritts ist bei TrAM ein zweistufiger Prozess. Während G in der ersten Phase die Verbindungen zu seinen Kindknoten G_{children} abbaut, wird in der zweiten Phase die Verbindung zu dem Elternknoten G_{parent} getrennt. Da G die Verbindungen zu seinen Kindknoten der Reihe nach abbaut, und ein solcher Verbindungsabbau jeweils mit der Bestimmung eines neuen Elternknotens für den betroffenen Kindknoten verbunden ist, kann die Ausgliederung von G aus der Overlay-Topologie eine gewisse Zeit (eventuell bis zu einigen Sekunden) in Anspruch nehmen. Während sich hier demnach eine Verzögerung ergeben kann, hat der Mechanismus im Vergleich zu anderen Overlay-Topologien den Vorteil, dass die Vermittlung von Daten während der Ausgliederung von G keine gravierenden Inkonsistenzen aufweist. Dies ergibt sich aufgrund der Tatsache, dass sich der Prozess der Ausgliederung aus Sicht der Kindknoten von G jeweils durch den Wechsel des Elternknotens beschreiben lässt. Die Behebung der durch die eventuell auftretenden Routing-Inkonsistenzen entstehenden Paketverluste wird einem übergeordneten Transportprotokoll überlassen.

Der Ausfall von Gruppenmitgliedern wird bei TrAM, analog zu dem von anderen Protokollen eingesetzten Verfahren, durch das Ausbleiben von periodisch zwischen Gruppenmitgliedern ausgetauschten Kontrollpaketen detektiert: Bei TrAM handelt es sich hierbei um die `childHello`- und `parentHello`-Pakete. Hinsichtlich der Trägheit zur Detektion von Ausfällen erzielt TrAM im Vergleich zu anderen Protokollen demnach keine Verbesserung. Die Reparatur der Overlay-Topologie im Anschluss an den erkannten Ausfall eines Gruppenmitglieds G gestaltet sich bei TrAM allerdings auf vergleichbar einfache Weise. Aus Sicht eines *Kindknotens* von G gilt es lediglich einen neuen Elternknoten zu lokalisieren, zu welchem eine Verbindung aufgebaut wird. Aus Sicht des Elternknotens von G wird G aufgrund des Soft-State-Ansatzes von TrAM bei ausbleibenden `childHello`-Paketen automatisch aus der Menge der Kindknoten gestrichen. Auch dem Ausfall des Wurzelknotens des TrAM-Baums kann durch einen Mechanismus zur Wahl eines neuen Wurzelknotens entgegen gewirkt werden.

4.3.5 Die Nutzung der Semi-Broadcast-Eigenschaft

Zur Vermittlung von Kontrollinformation zwischen einzelnen Gruppenmitgliedern setzen manche von TrAMs Protokollmechanismen die Semi-Broadcast-Eigenschaft des geteilten Mediums ein. So zum Beispiel der Mechanismus zur Lokalisation von Gruppenmitgliedern, welcher für den Gruppenbeitritt und die anschließende Pflege der Overlay-Topologie `childHello`-Pakete mit einer bestimmten Lebenszeit um ein Gruppenmitglied herum flutet. Ebenso nutzt nach der Detektion des Ausfalls des bisherigen Wurzelknotens der Mechanismus zur Wahl eines neuen Wurzelknotens die Semi-Broadcast-Eigenschaft, indem `detectRoot`-Pakete entsprechend um Gruppenmitglieder geflutet werden.

4. TrAM: Ein leichtgewichtiges Overlay für drahtlose Ad-hoc-Netze

Zur Vermittlung von Daten hingegen wird bei TrAM die Nutzung der Semi-Broadcast-Eigenschaft gezielt ausgelassen. Dies beruht auf der Tatsache, dass mit dem in Abschnitt 5.1.1 vorgestellten Mechanismus des lokalen Broadcast-Clustering ein sehr effizientes Verfahren zur Nutzung der Semi-Broadcast-Eigenschaft gefunden wurde, welches aufgrund seiner Generalität mit beliebigen Overlay-Topologien kombinierbar ist.

5. Verbesserungen von P2P-Multicast-Protokollen

In diesem Kapitel werden unterschiedliche Mechanismen vorgestellt, anhand derer sich die Leistungsfähigkeit von P2P-basierten Gruppenkommunikationsprotokollen in drahtlosen Ad-hoc-Netzen steigern lässt. Sie lassen sich in zwei Kategorien einteilen: Generische und Protokoll-spezifische Mechanismen. Während sich die generischen Mechanismen auf beliebige Overlay-Topologien anwenden lassen, handelt es sich bei den Protokoll-spezifischen Mechanismen um gezielte Änderungen bestehender Protokolle zum Erzielen einer höheren Leistungsfähigkeit.

5.1 Generische Verbesserungsmechanismen

In dieser Arbeit wurden Mechanismen entwickelt, welche die Leistungsfähigkeit und die Effizienz von P2P-basierten Gruppenkommunikationsprotokollen in drahtlosen Ad-hoc-Netzen steigern ohne sich direkt auf bestehende Protokollmechanismen einer Overlay-Topologie auszuwirken. Diese Mechanismen weisen einen generischen Charakter auf und sind daher mit beliebigen Overlay-Topologien kombinierbar.

5.1.1 Das lokale Broadcast-Clustering

Die Nutzung des geteilten Mediums ist von entscheidender Bedeutung, um auch in Netzarealen mit einer hohen Gruppenmitgliedsdichte eine effiziente Vermittlung von Daten zu gewährleisten. Im Hinblick auf die Frage nach einem möglichst geschickten Einsatz von Broadcast-Übertragungen ist in Betracht zu ziehen, dass der Aufbau einer Overlay-Topologie zwischen Gruppenmitgliedern, die sich in gegenseitiger Sende-reichweite befinden, nicht sinnvoll erscheint. In der Tat ist eine Verbindungsbewertung

durch Ende-zu-Ende-basierte Metriken auf engem Raum nicht möglich ist: Da sich keine ausreichende Differenzierung von Messwerten und somit keine Unterscheidbarkeit von Verbindungsqualitäten ergibt, kann eine Overlay-Topologie nicht zielgerichtet aufgebaut werden. In solchen Arealen bietet es sich an, auf den Aufbau einer Overlay-Topologie zu verzichten. Dieser Ansatz wird von unterschiedlichen, parallel entwickelten Arbeiten verfolgt [9, 28, 62]. Zu diesen zählt auch das in dieser Arbeit entwickelte Konzept des lokalen Broadcast-Clustering [9, 18, 19], welches im Folgenden kurz vorgestellt wird.

5.1.1.1 Die lokale Auflösung der Overlay-Topologie

Die Grundidee des lokalen Broadcast-Clustering besteht darin, die Overlay-Topologie lokal aufzulösen, d.h. auf den Aufbau von Overlay-Verbindungen zwischen Gruppenmitgliedern lokal zu verzichten. Diese sollen stattdessen nur noch zur Vermittlung von Daten zwischen Gruppenmitgliedern eingesetzt werden, die einen Abstand von mindestens zwei Übertragungsabschnitten aufweisen.

Um dies zu erreichen, spannt jedes dem Overlay beigetretene Gruppenmitglied einen so genannten *lokalen Broadcast-Cluster (LBC)* auf. Dieser ist definiert durch das aufspannende Overlay-Mitglied selbst sowie durch dessen Übertragungsreichweite. Zur Signalisierung der Existenz des Cluster versendet das Overlay-Mitglied periodische Kontrollpakete, so genannte *Heartbeats*, als Broadcast. Hierdurch wird es zum *Repräsentanten seines LBCs* und fungiert innerhalb der Multicast-Gruppe als *lokaler Repräsentant*. Gruppenmitglieder, welche sich innerhalb eines derartigen Cluster und somit in Übertragungsreichweite seines Repräsentanten befinden, erkennen aufgrund empfangener Heartbeats das umliegende Overlay-Mitglied und treten dem Overlay selbst nicht bei. Sie werden zu so genannten *lokal beigetretenen Gruppenmitgliedern*, oder kürzer *lokalen Gruppenmitgliedern*.

Die lokale Auflösung der Overlay-Topologie erfordert eine Erweiterung der Bootstrapping-Mechanismen und der Datenverteilung an die Gruppenmitglieder. Diese sowie generelle Konsequenzen und Eigenschaften des lokalen Broadcast-Clustering werden in den folgenden Abschnitten diskutiert.

5.1.1.2 Das Bootstrapping

Aufgrund der lokalen Auflösung der Overlay-Topologie sollen Gruppenmitglieder zu Beginn ihres Gruppenbeitritts nicht automatisch ihre Eingliederung in das Overlay-Netz initiieren. Stattdessen ist zunächst die eventuelle Verfügbarkeit eines umliegenden Overlay-Mitglieds zu prüfen, über welches der lokale Beitritt in einen LBC erfolgen könnte. Hierfür lauschen beitretende Gruppenmitglieder auf die periodisch von Overlay-Mitgliedern versendeten Heartbeats. Wird ein solches empfangen, befindet sich das beitretende Gruppenmitglied innerhalb eines LBCs und deklariert sich daraufhin als lokales Grup-

penmitglied. Hierdurch ist der Gruppenbeitritt bereits abgeschlossen, da sich das beitretende Mitglied einem LBC eindeutig zugeordnet hat. Wurde nach Ablauf einer gewissen Zeitspanne kein Heartbeat empfangen, wird davon ausgegangen, dass sich kein Overlay-Mitglied in Reichweite bzw. das beitretende Mitglied sich nicht innerhalb eines LBCs befindet. Um an der Gruppenkommunikation teilzunehmen, ist es daher erforderlich, das beitretende Gruppenmitglied in das Overlay einzugliedern. Dies geschieht anhand der gewöhnlichen Bootstrapping-Mechanismen des Overlay-Protokolls. Nach seiner erfolgreichen Eingliederung in das Overlay spannt das soeben beigetretene Gruppenmitglied wiederum einen eigenen LBC auf und fungiert als dessen Repräsentant.

5.1.1.3 Der Versand und die Vermittlung von Daten

Für das Versenden und das Vermitteln von Datenpaketen greifen dem Overlay beigetretene Gruppenmitglieder auf die gewöhnlichen, durch das Overlay-Protokoll spezifizierten Routing-Mechanismen zurück. Diese werden lediglich dadurch ergänzt, dass Overlay-Mitglieder Datenpakete zusätzlich zur Weiterleitung über das Overlay ebenfalls als Broadcast versenden. Wie in Abbildung 5.1 veranschaulicht ist, werden die Daten auf diese Weise den sich in Übertragungsbereich befindenden lokalen Gruppenmitgliedern zugestellt. Ein Vergleich zu der in Abschnitt 3.1.8 gezeigten Abbildung 3.4 verdeutlicht die Entlastung des drahtlosen Mediums: Anstatt der expliziten Zustellung des Pakets an jedes Gruppenmitglied können lokale Gruppenmitglieder nun anhand eines einzigen Medienzugriffs mit Daten versorgt werden.

Handelt es sich bei einem sendenden Gruppenmitglied um ein lokal beigetretenes Gruppenmitglied, wird ein zu versendendes Datenpaket per Unicast an den umliegenden lokalen Repräsentanten geleitet. Dieser wiederholt das Paket zur Verteilung an die übrigen lokalen Gruppenmitglieder zunächst mittels einer Broadcast-Übertragung und leitet es dann in gewohnter Weise über das Overlay weiter. Zu bemerken ist an dieser Stelle, dass die Broadcast-Übertragung des Pakets nur durch den Repräsentanten erfolgen kann. Dies beruht auf der Tatsache, dass wegen des beschränkten Übertragungsbereichs nicht davon ausgegangen werden kann, dass das sendende, lokale Gruppenmitglied alle übrigen, dem gleichen LBC zugeordneten lokalen Gruppenmitglieder anhand einer Broadcast-Übertragung erreicht.

5.1.1.4 Die Unterstützung von Mobilität

Die von lokalen Repräsentanten periodisch versendeten Heartbeats dienen nicht nur dem zuvor beschriebenen Bootstrapping sondern auch der Unterstützung der Mobilität von Gruppenmitgliedern. Im Hinblick auf diese Mobilität ist etwa darauf zu achten, dass lokale Gruppenmitglieder sich aus dem Übertragungsbereich ihrer lokalen Repräsentanten hinaus bewegen können. Um derartigen Änderungen zu begegnen, wird aufgrund der bekannten Heartbeat-Periodizität durch den Empfang bzw. durch das

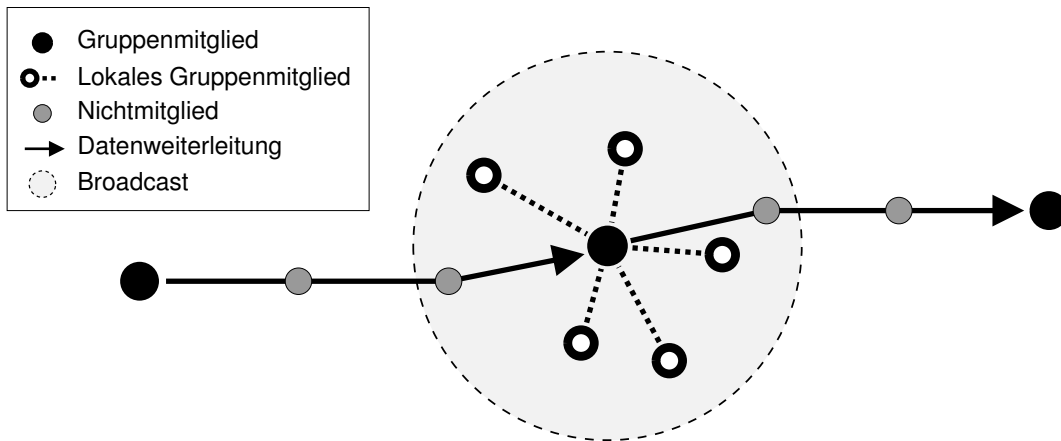


Abbildung 5.1 Nutzung des geteilten Mediums bei der Datenweiterleitung.

Ausbleiben erwarteter oder unerwarteter Heartbeats eine Bewertung lokaler Repräsentanten vorgenommen. Die dabei zum Einsatz kommenden Bewertungsmechanismen sind Anwendungs-abhängig, so dass Umsetzungen erst später diskutiert werden sollen. Während des Versendens von Heartbeats können die folgenden Situationen eintreten:

- *Ein lokales Gruppenmitglied empfängt ein Heartbeat von seinem lokalen Repräsentanten:* Dies ist die Standardsituation, anhand welcher ein lokal beigetretenes Gruppenmitglied feststellt, dass es sich in Übertragungsbereich seines lokalen Repräsentanten befindet und Daten von diesem beziehen kann. Durch den Empfang des Heartbeat verbessert sich die Bewertung des lokalen Repräsentanten.

- *Ein lokales Gruppenmitglied empfängt ein Heartbeat von einem anderen lokalen Repräsentanten:* Bei bestimmten räumlichen Verteilungen der Gruppenmitglieder kann es dazu kommen, dass sich ein lokal beigetretenes Gruppenmitglied in Übertragungsbereich von zwei oder mehr lokalen Repräsentanten befindet. Man spricht hierbei von einer *Überlappung lokaler Broadcast-Cluster*. Eine solche Überlappung äußert sich für lokal beigetretene Gruppenmitglieder, wie beispielsweise für Gruppenmitglied A in Abbildung 5.2, dadurch, dass Heartbeats unterschiedlicher Herkunft empfangen werden. Auf Basis der Bewertung, die für alle sich in Reichweite befindenden Repräsentanten vorgenommen wird, trifft ein lokales Gruppenmitglied eine Auswahl seines lokalen Repräsentanten: So ordnet sich das lokale Gruppenmitglied stets dem am besten bewerteten lokalen Repräsentanten zu, da davon ausgegangen wird, dass zu diesem die beste Verbindung besteht. Um im Fall ungünstiger Heartbeat-Empfänge eine Alternierung zwischen lokalen Repräsentanten zu vermeiden und so gemäß Abschnitt 3.1.10 eine Stabilisierung der Multicast-Bäume zu erreichen, erfolgt eine Änderung der LBC-Zuordnung erst dann, wenn die Qualität des bisherigen Repräsentanten für 3 aufeinander folgende Heartbeat-Intervalle schlechter als die eines anderen Repräsentanten ist.

– *Ein lokales Gruppenmitglied vermisst ein Heartbeat:* Diese Situation kann einerseits im Zusammenhang mit einem in der Nähe erfolgenden Medienzugriff eintreten, welcher den Heartbeat-Empfang stört. Andererseits kann der Verlust des Heartbeat auf die Tatsache zurückzuführen sein, dass das lokale Gruppenmitglied den Übertragungsbereich seines lokalen Repräsentanten verlässt. In beiden Fällen wirkt sich der Verlust durch eine Verschlechterung der Bewertung des lokalen Repräsentanten aus, so dass es bei einer Überlappung lokaler Broadcast-Cluster zu einem Wechsel der Repräsentanzordnung kommen kann. Fällt durch den Verlust eines Heartbeat die Bewertung unter einen gewissen Schwellenwert wird davon ausgegangen, dass sich das lokale Gruppenmitglied gänzlich aus dem Übertragungsbereich des lokalen Repräsentanten hinaus bewegt hat, d.h. die Verbindung zu diesem abgebrochen ist. Liegt keine Überlappung von LBCs vor, d.h. sind dem lokalen Gruppenmitglied keine weiteren umliegenden lokalen Repräsentanten bekannt, ist die Eingliederung des lokalen Gruppenmitglieds in das Overlay erforderlich, um es auch weiterhin an der Gruppenkommunikation teilhaben zu lassen. Diese Eingliederung erfolgt anhand der Bootstrapping-Mechanismen des Overlay. Dabei kann die Suche nach Overlay-Mitgliedern in der Regel übergangen werden, da mit dem ehemaligen lokalen Repräsentanten oftmals ein Overlay-Mitglied bereits bekannt ist. Nur für den Fall, in welchem eine Kontaktaufnahme zu diesem fehlschlägt, ist eine erneute Suche nach Overlay-Mitgliedern notwendig. Zu bemerken sei an dieser Stelle auch, dass nach seiner erfolgreichen Eingliederung das neue Overlay-Mitglied ebenfalls einen eigenen LBC aufspannt und in diesem die Rolle des lokalen Repräsentanten übernimmt.

– *Ein Overlay-Mitglied empfängt ein Heartbeat:* Befindet sich ein Overlay-Mitglied innerhalb des Übertragungsbereichs eines anderen Overlay-Mitglieds äußert sich dies durch den Empfang dessen Heartbeats. Man spricht hierbei von einer *Kollision lokaler Broadcast-Cluster*. Eine solche ist in Abbildung 5.2 dargestellt: Hier befindet sich das Overlay-Mitglied B in Übertragungreichweite von Overlay-Mitglied C. Da B nicht nur die Heartbeats von C sondern auch die von C als Broadcast weitergeleiteten Daten empfängt, besteht für B keine Notwendigkeit mehr, dem Overlay beigetreten zu sein. Als Folge dessen zieht B sich aus dem Overlay-Netz zurück und ordnet sich als lokal beigetretenes Gruppenmitglied dem lokalen Broadcast-Cluster von C zu. Dieser Rückzug entspricht einer Auflösung von Bs lokalem Broadcast-Cluster. Diese Auflösung wird von B den ihm eventuell zugeordneten lokalen Gruppenmitgliedern anhand eines speziellen, als Broadcast versendeten **AbandonCluster**-Pakets signalisiert. Für lokale Gruppenmitglieder, welche von der Auflösung betroffen sind, ergeben sich zwei mögliche Situationen. Entweder befinden sie sich in einer Überlappung von LBCs (wie beispielsweise das Gruppenmitglied D in Abbildung 5.2) oder der ehemalige lokale Repräsentant war der einzige, ihnen bekannte umliegende lokale Repräsentant (vgl. Gruppenmitglied E in Abbildung 5.2). Im ersten Fall ordnen sich die lokalen Gruppenmitglieder dem besten, ihnen noch bekannten lokalen Repräsentanten zu. Im zweiten Fall hingegen ist

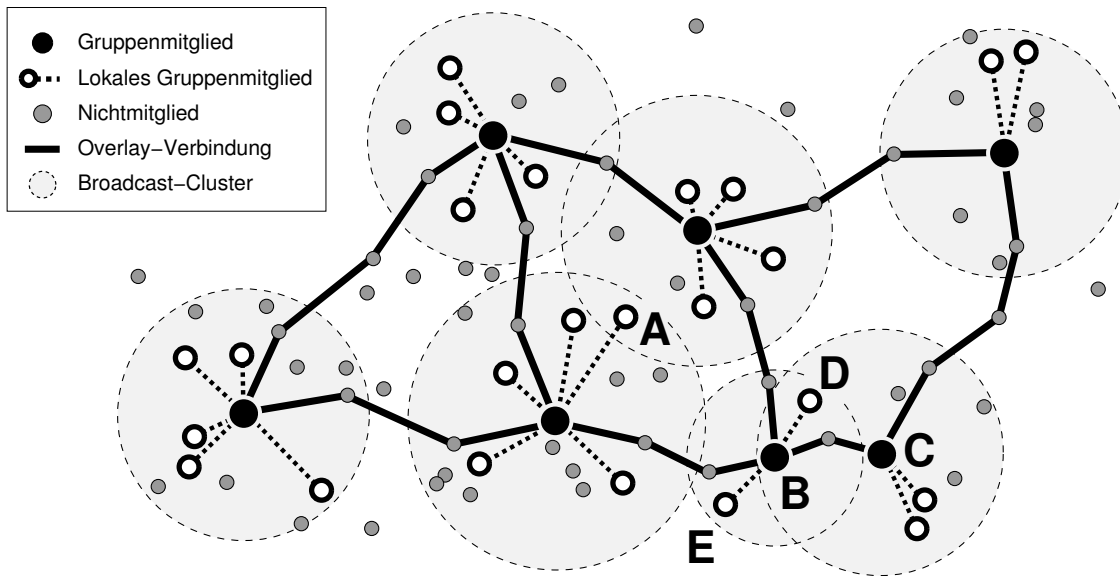


Abbildung 5.2 Topologie eines um das LBC-Konzept erweiterten Overlay-Netztes.

die Eingliederung eines lokalen Gruppenmitglieds in das Overlay notwendig. Dies erfolgt über die durch das Overlay-Protokoll vorgegebenen Bootstrapping-Mechanismen. Auch hier spannt das neu beigetretene Overlay-Mitglied einen eigenen LBC auf und übernimmt die Rolle eines lokalen Repräsentanten.

5.1.1.5 Zustandsübergänge und Zustandsautomat

Zur Verdeutlichung der angesprochenen Zustandsübergänge während des Gruppenbeitritts und der eigentlichen Gruppenmitgliedschaft, sollen an dieser Stelle die wichtigsten Aussagen durch den in Abbildung 5.3 dargestellten Zustandsautomaten zusammengefasst werden. Dabei signalisieren Grau hinterlegte Zustände, dass für ein Gruppenmitglied derzeit eine Verbindung zur Multicast-Gruppe besteht.

Anfangs verfügt ein Endgerät über keinerlei Verbindung zur Multicast-Gruppe und befindet sich im Zustand **Unverbunden**. Beim Auslösen des Gruppenbeitritts geht das Endgerät in den Zustand **Lokaler Bootstrap** über, in welchem die Anwesenheit eines lokalen Repräsentanten geprüft wird. Wird ein solcher gefunden, bewerkstelligt das Endgerät einen lokalen Beitritt und befindet sich fortan im Zustand **Lokales Mitglied**. Konnte hingegen im Laufe des lokalen Bootstrap kein lokaler Repräsentant gefunden werden, geht das Endgerät in den Zustand **Overlay-Suche** über, in welchem nach vorhandenen Overlay-Mitgliedern gesucht wird. Ist die Suche erfolglos, ist der Gruppenbeitritt fehlgeschlagen, so dass das Endgerät sich erneut im Zustand **Unverbunden** befindet. Konnte ein Overlay-Mitglied bestimmt werden, kann die Eingliederung des neuen Gruppenmitglieds in die Overlay-Topologie eingeleitet werden: Diese erfolgt im Zustand **Overlay-Bootstrap**. Ist die Eingliederung aufgrund einer fehlgeschlagenen Kontaktaufnahme zu

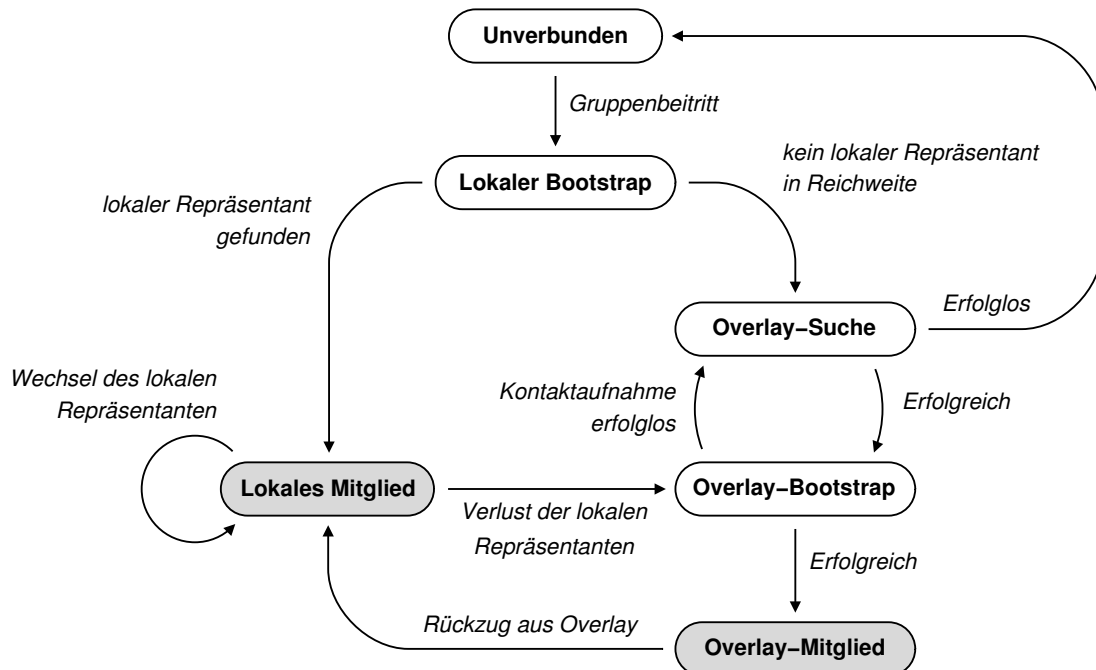


Abbildung 5.3 Zustandsautomat zur Beschreibung der Gruppenmitgliedschaft bei Verwendung des LBC-Konzepts.

den zuvor bestimmten Overlay-Mitgliedern erfolglos, findet sich das Endgerät erneut im Zustand **Overlay-Suche** wieder. Nach einer erfolgreichen Eingliederung befindet sich das neue Gruppenmitglied im Zustand **Overlay-Mitglied** und nimmt an der Gruppenkommunikation teil. Aufgrund einer Kollision lokaler Broadcast-Cluster kann sich ein Overlay-Mitglied aus dem Overlay zurückziehen, um fortan als lokales Gruppenmitglied im Zustand **Lokales Mitglied** zu verweilen. In Abhängigkeit der in diesem Zustand empfangenen und vermissten Heartbeats, kann das lokale Gruppenmitglied seinen lokalen Repräsentanten wechseln. Gehen diese aus Sicht des lokalen Gruppenmitglieds gänzlich verloren, ist die Eingliederung des Gruppenmitglieds in das Overlay erforderlich, so dass das Gruppenmitglied in den Zustand **Overlay-Bootstrap** übergeht. Wie in Abschnitt 5.1.1.4 angedeutet, kann die Suche nach existierenden Overlay-Mitgliedern hierbei übergangen werden, da mit dem ehemaligen lokalen Repräsentanten in der Regel ein bekanntes Overlay-Mitglied vorliegt. Schlägt jedoch die Kontaktaufnahme zu diesem fehl, geht das Gruppenmitglied in den Zustand **Overlay-Suche** über, in welchem es erneut nach Overlay-Mitgliedern Ausschau hält.

Ein wichtiger Aspekt beim Übergang aus dem Zustand **Overlay-Mitglied** in den Zustand **Lokales Mitglied** ist die Tatsache, dass der eigentliche Zustandsübergang instantan erfolgt: So nimmt ein Gruppenmitglied die Rolle eines lokalen Gruppenmitglieds ein, sobald die Entscheidung getroffen wurde, das Overlay zu verlassen. Die eigentliche Ausgliederung aus dem Overlay wird zu diesem Zeitpunkt jedoch erst angestoßen, so dass

innerhalb des Zustands **Lokales Mitglied** Protokollmechanismen des Overlay teilweise noch aktiv sein können. Dies liegt darin begründet, dass ein sich aus dem Overlay zurückziehendes Gruppenmitglied einerseits sofort seine Rolle als lokales Gruppenmitglied einnehmen soll und andererseits im Laufe seiner Ausgliederung eventuell noch immer Weiterleitungsaufgaben zu erfüllen hat.

5.1.1.6 Eigenschaften lokaler Gruppenmitglieder

Lokal beigetretene Gruppenmitglieder nehmen keinerlei Austausch von Kontrollinformation mit anderen Gruppenmitgliedern vor, weder mit umliegenden lokalen Repräsentanten noch mit anderen, dem gleichen lokalen Broadcast-Cluster zugeordneten lokalen Gruppenmitgliedern. Dies hat die folgenden Konsequenzen:

- Sofern ein lokales Gruppenmitglied nicht selbst als Sender agiert, hat es einen rein lauschenden Charakter: Es empfängt lediglich die von dem umliegenden lokalen Repräsentanten als Broadcast versendeten Daten und Heartbeats und führt selbst keine Medienzugriffe durch. Ein lokal beigetretenes Gruppenmitglied hat daher keinerlei Auswirkung auf die durch die Gruppenkommunikation verursachte Netzbelastung.

- Ein lokales Gruppenmitglied ist sowohl seinem lokalen Repräsentanten als auch anderen, dem gleichen LBC zugeordneten lokalen Gruppenmitgliedern gänzlich unbekannt. So kann auch im Fall von sehr dicht besetzten LBCs eine deutliche Reduktion des Speicheraufwands, sowohl für lokal beigetretene Gruppenmitglieder als auch für die lokalen Repräsentanten selbst, erreicht werden.

- Da ein lokales Gruppenmitglied Daten lediglich empfängt und selbst nicht weiterleitet, wird sein Energieverbrauch deutlich reduziert. Obgleich sie nicht Gegenstand dieser Arbeit sind, bietet sich daher die Konzeption von Protokollmechanismen an, welche leistungsschwächere Endgeräte gezielt als lokale Gruppenmitglieder beitreten lassen und die mit erhöhtem Energieverbrauch verbundene Funktion der lokalen Repräsentanten leistungstärkeren Endgeräten zuordnen. Dies kann durch eine Kodierung diverser Größen (Systemleistung oder Akkumulatorzustand) innerhalb der von lokalen Repräsentanten versendeten Heartbeats erfolgen: Empfängt ein leistungsstarkes Endgerät **A** ein Heartbeat, welches auf ein leistungsschwaches Endgerät **B** schließen lässt, kann **A** selbst die Rolle eines lokalen Repräsentanten übernehmen und somit das bisherige Gruppenmitglied **B** in seiner Funktion ablösen.

5.1.1.7 Diskussion und Bewertung

Bei dem lokalen Broadcast-Clustering handelt es sich um ein generisches Konzept, um welches ein beliebiges Overlay-Protokoll ergänzt werden kann. Es bringt verschiedene Vor- und Nachteile mit sich, welche im Folgenden kurz angesprochen werden.

Der bedeutendste Vorteil des Konzepts zeigt sich in Netzarealen, in welchen eine hohe Gruppenmitgliedsdichte herrscht. Hier kann eine Zustellung von Daten nun mit einer

deutlich geringeren Anzahl von Medienzugriffen vollzogen werden. Die damit in direktem Zusammenhang stehende Entlastung des Mediums wird zusätzlich durch die Tatsache verstärkt, dass auf den Aufbau einer Overlay-Topologie in derartigen Arealen gänzlich verzichtet wird. Da als Folge weitaus weniger Gruppenmitglieder dem eigentlichen Overlay beitreten, wird der durch das Overlay-Protokoll induzierte Kontrollfluss bzw. der zur Verwaltung der Gruppenmitglieder erforderliche Aufwand entsprechend reduziert. Wie in Abbildung 5.2 verdeutlicht ist werden Overlay-Verbindungen nun nur noch zwischen Gruppenmitgliedern aufgebaut, welche einen Abstand von mindestens 2 Übertragungsabschnitten haben. Das Overlay verbindet eine Menge von Gruppenmitgliedern miteinander, welche die Eigenschaft besitzen, dass die Gesamtheit ihrer Übertragungsbereiche das Netzareal abdeckt, in welchem sich die Gruppenmitglieder zu einem bestimmten Zeitpunkt aufhalten. Aufgrund von sich überlappenden LBCs kann keine Garantie dafür gegeben werden, dass diese Abdeckung jederzeit optimal ist, d.h. aus einer minimalen Anzahl lokaler Repräsentanten besteht. Jedoch kann durch die in Abschnitt 5.1.1.4 beschriebenen Mechanismen die Mobilität von Gruppenmitgliedern unterstützt und die dynamische Anpassung des abgedeckten Areals an die von den Gruppenmitgliedern über die Zeit tatsächlich besetzten Netzareale erreicht werden.

Aus Sicht einer übergeordneten Anwendung besteht der Hauptnachteil des LBC-Konzepts in der Fehleranfälligkeit von Paketen, welche als Broadcast übertragen werden. Wie in Abschnitt 2.1.1.3 verdeutlicht wurde, sind Broadcast- im Gegensatz zu Unicast-Übertragungen gänzlich unbestätigte Übertragungen und unterliegen somit im Fall von Paketkollisionen keinerlei Übertragungswiederholungen. Aus diesem Grund ist innerhalb von LBCs, welche in besonders stark belasteten Netzarealen liegen, mit einer höheren Fehlerrate zu rechnen als bei reinen Unicast-Übertragungen. Bereits an dieser Stelle soll deshalb angemerkt werden, dass sich der Einsatz des LBC-Konzepts nicht innerhalb aller Anwendungsszenarien empfiehlt. Sein Einsatz ist stattdessen im Kontext der Anwendungsanforderungen zu überlegen und abzuwägen [14].

Ein weiterer Nachteil des lokalen Broadcast-Clustering besteht im Wechsel der Rolle eines Gruppenmitglieds als lokal oder dem Overlay beigetretenes Gruppenmitglied: Sowohl der Rückzug eines Gruppenmitglieds aus dem Overlay-Netz (zwecks Auflösung seines lokalen Broadcast-Cluster) als auch die Eingliederung eines lokalen Gruppenmitglieds in das Overlay ist mit einer Umstrukturierung der Overlay-Topologie verbunden. Je nach Overlay-Topologie und deren Flexibilität erfolgen diese Umstrukturierungen mit erheblichem Aufwand und können so die in Abschnitt 3.1.5 angesprochenen Konsistenzprobleme verursachen. Der Einsatz des lokalen Broadcast-Clustering empfiehlt sich daher im Zusammenspiel mit einfach gehaltenen, flexiblen Overlay-Topologien, welche Umstrukturierungen mit geringem Aufwand bewerkstelligen können. Ein Überblick in-

wieweit sich die in dieser Arbeit betrachteten Overlay-Topologien zur Kombination mit LBCs eignen ist in Tabelle 5.1 gegeben. Diese soll im Folgenden kurz diskutiert werden.

Aufgrund seiner Leichtgewichtigkeit erweist sich TrAM als sehr gut mit LBCs kombinierbar: Einerseits lässt sich der Beitritt zu dem von TrAM aufgebauten Overlay sehr schnell und einfach bewerkstelligen. Andererseits sind auch die im Zusammenhang mit der Auflösung eines lokalen Broadcast-Cluster erforderlichen Overlay-Austritte als unkritisch zu betrachten, da sie unter Vermeidung von Topologieinkonsistenzen durchgeführt werden. Dies trifft auch auf die von Narada, PAST-DM und ALMA aufgebauten Overlay-Topologien zu, da hier die Ein- und Ausgliederung von Gruppenmitgliedern durch Austausch weniger Kontrollpakete mit umliegenden Gruppenmitgliedern erfolgt. Da bei Narada und PAST-DM Gruppenmitglieder in der Regel jedoch stärker mit anderen Gruppenmitgliedern verwoben sind als bei reinen baumförmigen Overlay-Topologien, ist ein Austausch von Kontrollpaketen mit einer größeren Anzahl von Gruppenmitgliedern erforderlich. Einzig die von NICE aufgebaute Overlay-Topologie eignet sich kaum zur Kombination mit LBCs: Hier bedeutet die Ausgliederung eines Gruppenmitglieds aus dem Overlay im schlimmsten Fall den Austritt des hierarchisch am höchsten eingestuften Gruppenmitglieds. Die dadurch erforderliche, vollständige Umstrukturierung der Hierarchie bezieht eine große Zahl von Gruppenmitgliedern und auszutauschenden Kontrollpaketen mit ein, so dass sie in kurzer Zeit nicht konsistent durchführbar ist. Zur Lösung dieser Problematik bietet es sich an, Gruppenmitgliedern auf höheren Hierarchieebenen den Rückzug aus dem Overlay aufgrund einer LBC-Kollision zu verbieten und ihn lediglich den nur auf tiefster Ebene vertretenen Gruppenmitgliedern zu erlauben. Während sich so die Kombinierbarkeit des NICE-Overlay mit dem LBC-Konzept verbessern lässt, ist zu beachten, dass sich bei weitläufigen Bewegungen der Gruppenmitglieder die LBC-Abdeckung über die Kommunikation hinweg deutlich verschlechtern kann, da Kollisionen zwischen LBCs nicht mehr gänzlich aufgelöst werden, und sich LBCs folglich verstärkt überlappen können.

LBCs besitzen die Eigenschaft, dass unter ihrem Einsatz der für die Gruppenkommunikation anfallende Aufwand abgeschätzt werden kann. Geht man von einem räumlich beschränkten Netzareal aus, so können darin nur eine beschränkte Anzahl lokaler Repräsentanten kollisionsfrei Platz finden. Ist das Netzareal vollständig durch die Übertragungreichweiten lokaler Repräsentanten abgedeckt, können alle übrigen Gruppenmitglieder einen lokalen Gruppenbeitritt vollziehen. Dies bedeutet einerseits, dass die Anzahl n der dem Overlay beigetretenen Gruppenmitglieder und somit auch der für die Verwaltung des Overlay anfallende Kontrollfluss nicht länger mit der Anzahl der Gruppenmitglieder wächst, sondern durch die Ausdehnung des Netzareals beschränkt wird. Da gemäß Abschnitt 5.1.1.6 lokal beigetretene Gruppenmitglieder selbst keinen Kontrollfluss verursachen, wird demnach durch den Einsatz von LBCs der für die Ver-

	TrAM	Narada	NICE	PAST-DM	ALMA
Kombinierbarkeit mit LBCs	⊕⊕	⊕	⊖⊖/○*	⊕	⊕⊕

Tabelle 5.1 Kombinierbarkeit der in dieser Arbeit betrachteten Overlay-Topologien mit dem Konzept lokaler Broadcast-Cluster (*: Siehe Text).

waltung der Multicast-Gruppe anfallende Gesamtaufwand durch das von der Gruppe besetzte Netzareal beschränkt. Handelt es sich bei dem zu erbringenden Gruppenkommunikationsdienst um einen unzuverlässigen Dienst, kann analog zu dem verursachten Kontrollfluss ebenfalls der zur Vermittlung von Nutzdaten anfallende Aufwand abgeschätzt werden: Bei einer Maximalzahl von n Overlay-Mitgliedern sind für eine zyklensfreie baumbasierte Vermittlung $n-1$ Weiterleitungen innerhalb des Overlay erforderlich. Dabei ist jedoch zu berücksichtigen, dass jede Weiterleitung eines Datenpakets entlang einer Overlay-Verbindung wiederum mindestens zwei Vermittlungsschritte auf Vermittlungsschicht erfordert. Die Vermittlung der Datenpakete an die übrigen (lokalen) Gruppenmitglieder erfolgt anhand von Broadcast-Übertragungen innerhalb der LBCs, wobei pro LBC und Datenpaket genau ein Mediengriff erforderlich ist.

In diesem Zusammenhang sind zwei Größen von besonderem Interesse. Zum einen ist die Anzahl der ausgebildeten LBCs betrachtenswert, da diese ausschlaggebend für den zur Gruppenverwaltung anfallenden Aufwand ist. Zum anderen ist die Wahrscheinlichkeit interessant, mit welcher ein neu beitretendes Gruppenmitglied einen lokalen Gruppenbeitritt bewerkstelligen kann. Beide Größen hängen von diversen Faktoren ab, wie die Größe des Netzareals, die Anzahl der einer Gruppe bereits angehörenden Gruppenmitglieder sowie die Übertragungsbereichsweite der drahtlosen Endgeräte, welche dem Radius der ausgebildeten LBCs entspricht. Zur Bestimmung beider Größen wurden unterschiedliche Rechner-gestützte Experimente durchgeführt, bei welchen auf Basis einer Gleichverteilung 100 Endgeräte zufällig nacheinander in ein quadratisches Areal gesetzt werden: Befindet sich ein soeben platziertes Endgerät innerhalb einer bestimmten Entfernung d zu einem bereits platzierten Endgerät, wird das neue Endgerät als lokal beigetretenes Gruppenmitglied gewertet und wieder aus dem Areal entfernt – andernfalls wird es als neues Overlay-Mitglied angesehen und im Areal belassen. Die maximale Entfernung d , bis zu welcher eine lokale Zuordnung stattfindet, entspricht dem Übertragungsbereich der Endgeräte und somit dem Radius der LBCs. Sie beträgt für die hier gezeigten Ergebnisse $d = 150m$, wobei durch Variation der Arealsgröße jedoch ebenso Aussagen für andere Übertragungsbereiche möglich werden: So sind Messungen bei einer Arealsgröße von $2000 \times 2000 m^2$ und einem Übertragungsbereich von $300m$ identisch zu Messungen bei einer Arealsgröße von $1000 \times 1000 m^2$ und einem Übertragungsbereich von $150m$. Die Anzahl der durchschnittlich ausgebildeten lokalen Repräsentanten ist in Abbildung 5.4 für verschiedene Netzarealsgrößen in Abhängigkeit der Anzahl bereits

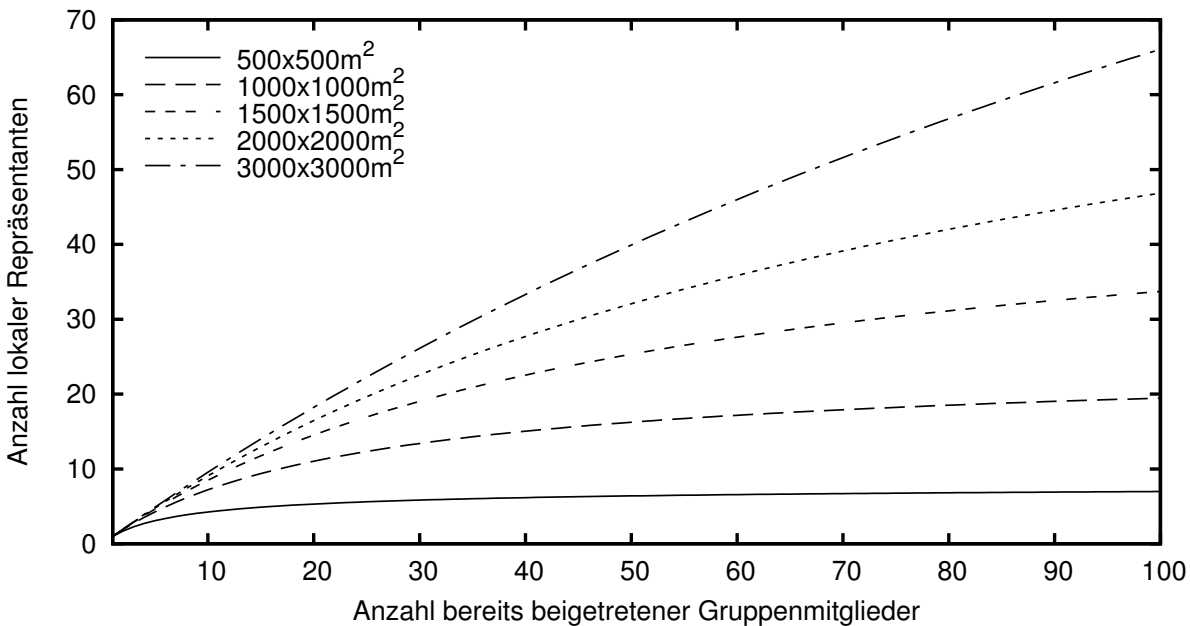


Abbildung 5.4 Anzahl ausgebildeter lokaler Repräsentanten bei einer Übertragungsbereichsweite von $150m$.

beigetretener Gruppenmitglieder dargestellt. Dabei ist insbesondere für die Kurven kleiner Netzareale erkennbar, dass die Werte zügig gegen eine obere Schranke konvergieren, welche in dieser Arbeit aufgrund der in [99, 108] dargestellten Problematik jedoch nicht näher bestimmt werden soll. Die genannte Schranke gibt die Maximalzahl der lokalen Repräsentanten an, die Kollisionsfrei auf dem entsprechenden Netzareal untergebracht werden können. In Abbildung 5.5 hingegen ist die Wahrscheinlichkeit aufgetragen, mit welcher ein neu beitretendes Gruppenmitglied einen lokalen Beitritt bewerkstelligen kann. Hier ist zu erkennen, dass bei kleinen Netzarealen die Wahrscheinlichkeit eines lokalen Beitritts rapide ansteigt: So kann beispielsweise bei einer Arealsgröße von $500 \times 500 m^2$ das 20. beitretende Gruppenmitglied bereits mit einer Wahrscheinlichkeit von über 92% einen lokalen Beitritt bewerkstelligen. Bei einem Areal von $3000 \times 3000 m^2$ hingegen liegt die Wahrscheinlichkeit für das Mitglied lediglich bei 18%.

5.1.2 Die Stabilisierung des Routing

In dieser Arbeit konnte mitunter gezeigt werden, dass die von einer Overlay-Topologie vorgenommene Datenvermittlung auch über längere Zeiträume anhaltende Lücken aufweisen kann, welche auf Inkonsistenzen der Routing-Information zurückgehen [10, 11]. Insbesondere im Zusammenhang mit einer Gruppenkommunikation, welche zuverlässig erfolgen soll, erweisen sich derartige Lücken als problematisch, da sie durch ein Transportprotokoll behoben werden müssen und daher den Datenaustausch verzögern. Um diese Problematik einzudämmen, wurde ein einfaches Verfahren zur Stabilisierung des

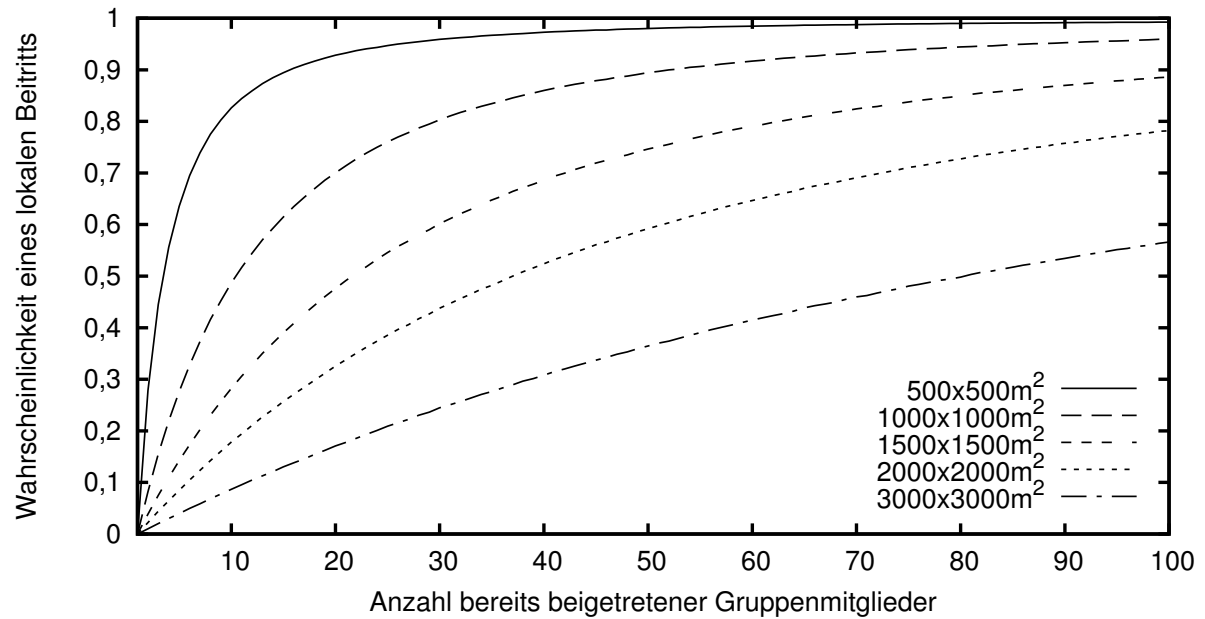


Abbildung 5.5 Wahrscheinlichkeit eines lokalen Gruppenbeitritts bei einer Übertragungreichweite von $150m$.

Routing entwickelt, welches einen generischen Charakter aufweist und sich daher mit beliebigen Overlay-Topologien kombinieren lässt. Es basiert auf einer *Entkopplung des Datenflusses* von der durch das Overlay vorgenommenen Weiterleitung der Datenpakete, indem für Multicast-Quellen, gemäß Abschnitt 3.1.5, konsistente Multicast-Bäume explizit aufgebaut und für die Weiterleitung von Datenpaketen verwendet werden.

Zur Entkopplung des Datenflusses verwaltet jedes Gruppenmitglied g für jede aktive Multicast-Quelle q zusätzliche Datenstrukturen r_q und \mathcal{N}_q , welche unabhängig von den Routing-Tabellen des Overlay-Netzes sind. Während g in r_q dasjenige Gruppenmitglied g speichert, bei welchem es derzeit als Nachfolger der Multicast-Quelle q registriert ist, stellt \mathcal{N}_q aus Sicht von g wiederum die eigene Menge der Nachfolger für die Multicast-Quelle q dar. Die Weiterleitung von Datenpaketen der Quelle q erfolgt nun nicht mehr aufgrund des Overlay-Routing, sondern aufgrund der in jedem Gruppenmitglied verwalteten Menge \mathcal{N}_q : So leitet ein Gruppenmitglied g Datenpakete der Multicast-Quelle q genau an jene Gruppenmitglieder weiter, welche in \mathcal{N}_q enthalten sind.

Um Gruppenmitglieder in die Menge \mathcal{N}_q aufzunehmen, sieht die Entkopplung des Datenflusses vor, dass eine aktive Multicast-Quelle q periodisch so genannte **UpdateMulticastTree**-Pakete versendet. Diese werden gemäß dem *ursprünglichen* Overlay-Routing zu den Gruppenmitgliedern verteilt. Empfängt ein Gruppenmitglied g ein solches **UpdateMulticastTree**-Paket von einem Gruppenmitglied g' , so erfährt es, dass sich g' derzeit zuständig für die Weiterleitung der Daten von q an g sieht. Aus diesem Grund registriert

sich g als Nachfolger bei g' , indem es diesem ein `RegisterAsChild`-Paket zusendet. Empfängt g' dieses Paket, fügt es g der Menge \mathcal{N}_q hinzu und leitet fortan Datenpakete der Multicast-Quelle q an g weiter. Nach Empfang des ersten Datenpakets der Multicast-Quelle q durch g' sieht g seine Registrierung bei g' als erfolgreich und setzt $r_q = g'$. War g zuvor bereits bei einem anderen Gruppenmitglied g^* als Nachfolger registriert, so erfolgt bei diesem eine Abmeldung durch Zusenden eines `UnregisterAsChild`-Pakets. Bei dessen Empfang entfernt g^* das Gruppenmitglied g aus der Menge \mathcal{N}_q .

Tritt eine Inkonsistenz des Overlay-Routing auf, bei welcher sich g^* *nicht mehr* und g' *noch nicht* für die Weiterleitung der Datenpakete von q an g zuständig sieht, betrifft dies nunmehr lediglich die `UpdateMulticastTree`-Pakete. g ist jedoch noch immer als Nachfolger für q bei g^* registriert, so dass von q stammende Datenpakete entsprechend noch immer an g weitergeleitet werden. Nach Behebung der Inkonsistenz leitet g' zunächst lediglich die von q versendeten `UpdateMulticastTree`-Pakete an g weiter, so dass sich g entsprechend als Nachfolger bei g' anmeldet. Da g' nun auch Datenpakete von q an g weiterleitet, meldet sich g als Nachfolger bei g^* ab.

Die Effizienz des Verfahrens ist eng an die Anzahl aktiver Multicast-Quellen und den von ihnen ausgesendeten Datenverkehr gebunden. Vorteilhaft zeigt sich das Verfahren vor allem für Szenarien, in welchen einzelne Multicast-Quellen einen kontinuierlichen Datenstrom aussenden, dessen Übertragung einerseits zuverlässig erfolgen soll und dessen Datenpakete andererseits in einem zeitlichen Abstand gesendet werden, der geringer als die Periodizität der `UpdateMulticastTree`-Pakete ist [10, 11]. Bei mehreren Multicast-Quellen, welche einen unregelmäßigen Datenverkehr aussenden, zeigt sich das Verfahren als kontraproduktiv [68]. Dies geht auf die Tatsache zurück, dass jede Multicast-Quelle mit den versendeten `UpdateMulticastTree`-Paketen einen zusätzlichen Kontrollfluss erzeugt, welcher mit ansteigender Quellenzahl ein drahtloses Ad-hoc-Netz zu sehr belastet. Einerseits lässt sich dieser durch eine Erhöhung der `UpdateMulticastTree`-Periodizität zwar reduzieren. Andererseits wird die Weiterleitung der Daten hierdurch träger bezüglich Änderungen innerhalb der Overlay-Topologie, da die über die Mengen \mathcal{N}_q explizit aufgebauten Multicast-Bäume nur selten aktualisiert werden.

Das hier beschriebene Verfahren zur Stabilisierung des Routing eignet sich vor allem zur zuverlässigen Gruppenkommunikation. Da ausgiebige Untersuchungen in diesem Bereich jedoch über den Rahmen dieser Arbeit hinausgehen, soll das Verfahren hier lediglich erwähnt, in späteren Evaluierungen jedoch nicht weiter betrachtet werden.

5.1.3 Die initiale Nachbarsuche

Bezüglich der initialen, von Endgeräten vor dem Gruppenbeitritt durchzuführenden Nachbarsuche herrscht in der Literatur oftmals Uneinstimmigkeit im Hinblick auf das dafür eingesetzte Verfahren. In dieser Arbeit sollen jedoch die Grundsätze drahtloser Ad-

hoc-Netze (Dezentralität und Selbstorganisation) nicht gebrochen werden, so dass für die initiale Nachbarsuche das Vorhandensein eines Rendez-vous-Punktes nicht infrage kommt. Um darüber hinaus für alle in dieser Arbeit betrachteten Overlay-Topologien identische Ausgangsbedingungen zu schaffen, sollen für sie gleichermaßen das dezentrale ERS-Verfahren herangezogen werden.

Während es sich bei dem ERS-Verfahren selbst um ein von der Overlay-Topologie unabhängiges Verfahren handelt, erfordert seine Integration in bestehende Overlay-Protokolle jedoch gewisse Anpassungen deren Bootstrapping-Mechanismen. Diese wiederum sind Protokoll-spezifisch, so dass sie in Abschnitt 5.2 vertieft werden.

5.2 Protokollspezifische Modifikationen

In diesem Abschnitt werden einzelne Modifikationen eingebracht, welche an den in dieser Arbeit untersuchten Overlay-Topologien vorgenommen wurden, um deren Leistungsfähigkeit bei einem Betrieb in drahtlosen Ad-hoc-Netzen zu steigern. Dies betrifft auch das eigens für drahtlose Ad-hoc-Netze entworfene Protokoll PAST-DM, da dessen ursprüngliche Beschreibung Lücken aufweist und im Hinblick auf manches Protokollverhalten innerhalb dieser Arbeit ergänzt werden musste.

5.2.1 NICE

Dieser Abschnitt fasst Konzepte zusammen, welche in dieser Arbeit entwickelt wurden, um das Overlay-Protokoll NICE bei einem Betrieb in drahtlosen Ad-hoc-Netzen leistungsfähiger zu machen.

5.2.1.1 Der Gruppenbeitritt

Während die initiale Nachbarsuche bei NICE anhand eines Rendez-vous-Punktes erfolgt, wurde dieses Verhalten in dieser Arbeit auf ein ERS-Verfahren umgestellt. Dabei wird jedoch die durchgeführte Ringsuche nicht zur Bestimmung eines Gruppenmitglieds verwendet, von welchem anschließend das sich derzeit auf höchster Hierarchieebene befindende Gruppenmitglied in Erfahrung gebracht wird. Vielmehr wird das ERS-Verfahren direkt verwendet, um Repräsentanten eines Cluster auf Ebene 0 zu lokalisieren. Hierdurch wird nicht nur eine schnellere sondern auch eine effizientere Einbindung des Gruppenmitglieds in die NICE-Hierarchie erreicht, da der tatsächlich nächstgelegene Cluster-Repräsentant auf Ebene 0 direkt und nicht anhand des iterativen Durchlaufs der NICE-Hierarchie bestimmt wird.

Zur Umsetzung des Verfahrens versendet der Repräsentant eines Ebene-0-Cluster bei Empfang eines ERS-Paketes eine entsprechende Antwort an das beitretende Gruppenmitglied. Das beitretende Gruppenmitglied sammelt die eintreffenden Antworten. Auf Basis der darin kodierten Abstandsinformation wählt es den nächstgelegenen Repräsentanten aus und tritt dessen Ebene-0-Cluster bei.

5.2.1.2 Das Aufspalten eines Cluster

Das Aufspalten eines Cluster wurde im Laufe dieser Arbeit als ein innerhalb von drahtlosen Ad-hoc-Netzen kritischer Prozess erkannt. Dies resultiert aus der Tatsache, dass nach der Aufspaltung eines Cluster, der Repräsentant des ehemaligen Cluster gegebenenfalls seine Repräsentantenfunktion abgeben muss. Der sich daran anschließende Prozess, bei welchem sich der ehemalige Repräsentant aus höheren Ebenen zurückzieht, ist in drahtlosen Ad-hoc-Netzen anfällig gegenüber dem Verlust von Kontrollpaketen und den daraus resultierenden Inkonsistenzen innerhalb der Overlay-Topologie. Aus diesem Grund behält in dieser Arbeit nach der Aufspaltung eines Cluster der ehemalige Repräsentant seine Funktion bei.

5.2.1.3 Optimierungsmetrik

In dieser Arbeit wurde die von NICE verwendete RTT-basierte Metrik zur Bewertung von Overlay-Verbindungen durch die Anzahl der Übertragungsabschnitte ersetzt. Dabei wird die Übertragungsabschnittszahl direkt aus der durch das Unicast-Routing-Protokoll geführten Routing-Tabelle ausgelesen und somit zusätzlicher Aufwand vermieden.

5.2.2 Narada

In diesem Abschnitt werden Modifikationen beschrieben, welche an dem Overlay-Protokoll Narada vorgenommen wurden, um dessen Leistungsfähigkeit in drahtlosen Ad-hoc-Netzen zu erhöhen.

5.2.2.1 Die initiale Nachbarsuche

Die ursprüngliche Protokollbeschreibung für Narada geht davon aus, dass einem beitretenden Gruppenmitglied implizit eine Liste aller bereits beigetretenen Gruppenmitglieder vorliegt. Um derartiges Vorwissen zu vermeiden, wurde Narada innerhalb dieser Arbeit dahingehend modifiziert, dass eine solche Liste von einem bereits beigetretenen Gruppenmitglied bezogen wird. Dieses Gruppenmitglied wird wiederum anhand eines ERS-Verfahrens lokalisiert.

5.2.2.2 Aggregation von Topologie- und Routing-Information

Das ursprüngliche Narada-Protokoll sieht vor, dass über dem aufgebauten Overlay ein separates Routing-Protokoll aufgesetzt wird, welches die Weiterleitung von Datenpaketen anhand von Shortest-Path-Information vornimmt. Zur Reduktion der erforderlichen Medienzugriffe werden in dieser Arbeit die ausgesendeten Refresh-Pakete mit den Routing-Paketen zusammengefasst. Dabei kann die Shortest-Path-Information auf einfache Weise komprimiert werden: So besteht die Shortest-Path-Information aus der Aneinanderreihung von Adressen einiger Gruppenmitglieder, welche jedoch schon innerhalb der

im gleichen Paket kodierten Refresh-Information aufgelistet sind. Anstatt einen Shortest-Path durch eine Abfolge vollständiger Adressen a_0, a_1, \dots, a_n zu kodieren, wird in dieser Arbeit eine Abfolge von Indizes i_0, i_1, \dots, i_n verwendet: Der Wert eines Indexes i_j referenziert dabei die innerhalb der Refresh-Information an der Stelle i_j aufgelistete Gruppenmitgliedsadresse. Aufgrund der in dieser Arbeit anvisierten Gruppengrößen kann bei Verwendung von 8-Bit-Indizes der Platzbedarf zur Kodierung der Shortest-Path-Information gegenüber vollständigen IPv4-Adressen um 75% reduziert werden.

5.2.2.3 Intelligent Probing

Zur Bewertung potenzieller Overlay-Verbindungen wählt Narada in bestimmten zeitlichen Abständen ein zufälliges Gruppenmitglied aus. Zu diesem wird die Verbindungsqualität ermittelt, so dass anschließend die Nützlichkeit der Verbindung bestimmt werden kann. Um zu vermeiden, dass ein Gruppenmitglied, zu welchem die Verbindung als unnützlich erkannt wurde, zeitlich nah erneut ausgewählt wird, wurde Narada dahingehend modifiziert, dass Gruppenmitglieder nicht gleichverteilt ausgewählt werden. Ähnlich einer arithmetischen Kodierung werden nun zunächst die Nützlichkeiten N_i aller Verbindungen auf das Intervall

$$\mathcal{N} := \left[0, \frac{N_0}{\sum N_i}, \frac{N_0 + N_1}{\sum N_i}, \dots, \frac{N_0 + N_1 + \dots + N_{n-1}}{\sum N_i}, 1\right)$$

normiert. Auf Basis einer Gleichverteilung wird eine Zahl $p \in [0, 1)$ gezogen und das mit p in \mathcal{N} assoziierte Gruppenmitglied ausgewählt.

5.2.3 PAST-DM

Zur Steigerung der Leistung des PAST-DM-Protokolls werden in diesem Abschnitt zwei Verfahren vorgestellt, anhand welcher das ursprüngliche Protokoll in dieser Arbeit abgewandelt wurde.

5.2.3.1 Source-Routen-Konstruktion

Das ursprüngliche PAST-DM Protokoll verwendet zur Konstruktion der Source-Routen ein Verfahren auf Basis von Steiner-Bäumen. In dieser Arbeit werden die von PAST-DM verwendeten Source-Routen anhand des Dijkstra-Algorithmus berechnet: Auf diese Weise wird erreicht, dass ein Datenpaket ausgehend von einer Multicast-Quelle jedes Gruppenmitglied über einen Overlay-Pfad mit den geringsten Kosten erreicht.

Die verteilte Berechnung der Source-Routen bleibt im Vergleich zum ursprünglichen Protokoll erhalten: Demzufolge beauftragt eine Multicast-Quelle anhand des Ergebnisses des Dijkstra-Algorithmus benachbarte Gruppenmitglieder mit der Weiterleitung eines Datenpakets an entsprechende Subgruppen.

5.2.3.2 Globale Optimierung der Overlay-Topologie

Das in Abschnitt 3.4.1 für PAST-DM beschriebene Verfahren zur Anpassung der Overlay-Topologie an die vorherrschende Mobilität gleicht einer lokalen Verbesserung der Overlay-Topologie: So werden Overlay-Verbindungen bevorzugt zu nahe gelegenen Gruppenmitgliedern aufgebaut, während Overlay-Verbindungen zu entfernten Gruppenmitgliedern nur zur Vermeidung von Overlay-Partitionierungen bestehen bleiben.

In dieser Arbeit wurde das PAST-DM-Protokoll um Konzepte erweitert, welche an das Protokoll Narada angelehnt sind: Dieses sieht eine Overlay-Verbindung zu einem Gruppenmitglied g genau dann als nützlich, wenn ein bestimmter Anteil der Gruppenmitglieder durch Hinzunahme dieser Overlay-Verbindung mit geringeren Kosten als über das bisherige Overlay erreicht werden kann. Dies ermöglicht insbesondere den Aufbau von Overlay-Verbindungen zu entfernten Gruppenmitgliedern, sofern diese anhand der neuen Verbindung günstiger zu erreichen sind als über die Verkettung existierender Overlay-Verbindungen. Ähnlich zu dem für Narada in Abschnitt 5.2.2.3 beschriebenen Probing-Verfahren wählt PAST-DM in dieser Arbeit periodisch ein zufälliges Gruppenmitglied aus und misst die Entfernung zu diesem. Aufgrund der eigens verwalteten Link-States kann das messende Gruppenmitglied mittels des Dijkstra-Algorithmus im Anschluss entscheiden, ob sich anhand der getesteten Overlay-Verbindung Verteilbäume effizienter gestalten lassen.

5.3 Zusammenfassung

In diesem Kapitel wurden unterschiedliche Konzepte vorgestellt, welche in dieser Arbeit entwickelt wurden, um die Leistungsfähigkeit von P2P-basierten Gruppenkommunikationsprotokollen in drahtlosen Ad-hoc-Netzen zu verbessern. Einige dieser Konzepte sind Protokoll-spezifisch und wurden gezielt dahingehend entwickelt, um die Leistungsfähigkeit bestimmter Overlay-Topologien in drahtlosen Ad-hoc-Netzen durch Modifikation des Protokollverhaltens zu erhöhen. Darüber hinaus wurden zusätzliche Mechanismen vorgestellt, welche wiederum einen generischen Charakter aufweisen, so dass sie sich auf beliebige Overlay-Topologien anwenden lassen: So zum Beispiel das Konzept lokaler Broadcast-Cluster, welches den Verwaltungsaufwand von Overlay-Topologien reduziert, indem bestimmte Gruppenmitglieder gezielt aus der Overlay-Topologie ausgeschlossen und anhand von Broadcast-Übertragungen mit Daten versorgt werden. Im Hinblick auf die Generizität dieses Konzepts ist insbesondere die Kapselung seiner Protokollmechanismen ebenso von Interesse wie die Definition geeigneter Schnittstellen, welche eine Kombination mit beliebigen Overlay-Topologien tatsächlich ermöglichen. Auf diese Aspekte soll im folgenden Kapitel näher eingegangen werden.

6. Die MAAM-Architektur – Konzepte und Umsetzungen

In diesem Kapitel wird die *Modulare Architektur für Application-Layer Multicast* (die MAAM-Architektur) vorgestellt, welche in dieser Arbeit entwickelt wurde [12, 16]. Sie erlaubt die modulare Komposition [120] eines P2P-basierten Gruppenkommunikationsdienstes auf Basis von Protokollkomponenten, so dass sie sich als flexibel gegenüber der Berücksichtigung heterogener Anwendungsanforderungen erweist.

Für die Vorstellung der MAAM-Architektur werden anfangs die genauen Ziele definiert, welche mit dieser Architektur verfolgt werden. Um die Möglichkeiten zu verdeutlichen, welche die Architektur zur Komposition von P2P-basierten Gruppenkommunikationsdiensten bietet, werden anschließend die von der Architektur definierten Module und Modulschnittstellen näher betrachtet: Dabei wird sowohl auf Aspekte bezüglich der Integration von Overlay-Topologien als auch von Transportprotokollen eingegangen, sowie deren optionalen Kombination mit dem Konzept lokaler Broadcast-Cluster zur Erstellung skalierbarer Gruppenkommunikationsdienste. Darüber hinaus werden zusätzliche Konzepte betrachtet, anhand derer die MAAM-Architektur Systemunabhängigkeit erlangt: Auf Basis dieser kann die Architektur und die darin integrierten Protokollmechanismen sowohl innerhalb von Netzwerksimulationsumgebungen evaluiert als auch auf realen Endgeräten betrieben werden. Um das Zusammenspiel der einzelnen Module zu verdeutlichen, wird abschließend beispielhaft die Behandlung von Datenpaketen bei deren Versendung und Weiterleitung betrachtet.

Dieses Kapitel zielt darauf, lediglich die Konzepte hinter der MAAM-Architektur zu vermitteln und nicht eine Implementierung zu spezifizieren. Aus diesem Grund werden

Implementierungs-spezifische Details der in dieser Arbeit tatsächlich durchgeführten praktischen Umsetzung der Architektur nicht weiter vertieft. Dennoch soll der von der MAAM-Architektur verfolgte, Objekt-orientierte Ansatz dargelegt werden, da dieser die Darstellung von Modulabhängigkeiten und -gemeinsamkeiten deutlich vereinfacht.

6.1 Ziele und Anforderungen

Die MAAM-Architektur verfolgt unterschiedliche Ziele, sowohl im Hinblick auf die Unterstützung heterogener Anwendungsanforderungen, als auch hinsichtlich der Portierbarkeit der auf Basis der Architektur entwickelten Gruppenkommunikationsdienste. Die verfolgten Ziele können wie folgt zusammengefasst werden:

1. *Die Integration beliebiger Overlay-Netze und Zuverlässigkeitsmechanismen:* Um heterogenen Anwendungsanforderungen möglichst gerecht werden zu können, sollen von der Architektur keinerlei Einschränkungen bezüglich der tatsächlich integrierbaren Protokollmechanismen gemacht werden.
2. *Der nahtlose und transparente Austausch von Overlay-Netzen und Zuverlässigkeitsmechanismen:* Die Architektur soll die in Abschnitt 3.3 bereits angedeutete Trennung von Overlay-Netzen und Zuverlässigkeitsmechanismen vollziehen. Durch deren Umsetzung als modulare Protokollkomponenten soll ihre flexible Kombination und Komposition zu einem Gruppenkommunikationsdienst ermöglicht werden. Hiermit soll einerseits erreicht werden, dass Anwendungen auf spezialisierte Module zur Befriedigung ihrer Anforderungen zurückgreifen können. Andererseits wird hierdurch auch die Wiederverwendung einzelner Protokollkomponenten angestrebt, so dass ein Overlay-Netz im Zusammenspiel mit unterschiedlichen Zuverlässigkeitsmechanismen und Anwendungen betrieben werden kann.
3. *Die transparente Integration des Konzepts lokaler Broadcast-Cluster:* Dabei soll die tatsächliche Nutzung des Konzepts für Anwendungen optional gehalten werden, da nicht notwendigerweise jedes Anwendungsszenario von einer Datenverteilung anhand von Broadcast-Übertragungen profitiert.
4. *Der System-unabhängige Betrieb der Architektur sowie der darin integrierten Protokollmechanismen:* Hiermit soll insbesondere die Migration entwickelter P2P-basierter Gruppenkommunikationsdienste zwischen a) ihrer Evaluation innerhalb von Netzwerksimulationsumgebungen und b) ihrem Betrieb auf realen Endgeräten vereinfacht werden.

Um diese Ziele zu erreichen, sind diverse Entwurfsentscheidungen nötig und zahlreiche Mechanismen zu definieren, beispielsweise zur Architektur-internen Verarbeitung und

Weiterleitung von Datenpaketen. Der sich daraus ergebende Aufbau der Architektur wird innerhalb des nächsten Abschnitts verdeutlicht.

6.2 Der Aufbau der MAAM-Architektur

Obwohl die MAAM-Architektur über einen modularen Aufbau verfügt, kann sie ähnlich dem ISO/OSI Referenzmodell als Schichtenarchitektur dargestellt werden. Eine solche Darstellung ist in Abbildung 6.1 zu finden, in welcher die Architektur zwischen einer übergeordneten Benutzerschnittstelle (GUI) und einem untergeordneten Basissystem zu finden ist. Während die Rollen der Benutzerschnittstelle und des Basissystems später verdeutlicht werden, sollen zunächst die Aufgaben und die Funktionalität der dazwischen liegenden MAAM-Module betrachtet werden.

Um das Zusammenspiel und die Austauschbarkeit dieser Module nachvollziehen zu können, soll hier bereits festgehalten werden, dass die in der Abbildung aufgezeigten “Module” nicht nur für die darin untergebrachten Protokollmechanismen stehen, sondern auch die *von diesen Protokollmechanismen zu erfüllenden Schnittstellen* benennen: Während über die Schnittstelle **Application Core** auf die Kernfunktionalität von Anwendungen zugegriffen werden kann¹⁶, findet der Zugang zu Zuverlässigkeitsmechanismen über die Schnittstelle **Transport Base** statt. Die Schnittstelle **Multicast Base** fungiert wiederum als Zugang zu Protokollmechanismen für die Weiterleitung von Datenpaketen. Abgerundet wird die Architektur durch Strategien zur Paketverwaltung, welche über die Schnittstelle **Packet Queue** zugänglich sind.

Zum besseren Verständnis der Abhängigkeiten zwischen der Spezifikation von *Modulschnittstellen* und der tatsächlichen *Integration* von Protokollmechanismen ist in Abbildung 6.2 das von der MAAM-Architektur verwendete Klassen- und Vererbungsdiagramm dargestellt. Um die in Abbildung 6.1 dargestellten Schnittstellen besser auffinden zu können, sind diese in Abbildung 6.2 durch schwarze Kästchen in der oberen rechten Ecke markiert. Anhand des Diagramms wird insbesondere deutlich, welche Module und Protokollmechanismen sich hinter den in Abbildung 6.1 dargestellten Schnittstellen befinden können.

In den folgenden Abschnitten wird auf die Integration von Overlay-Topologien und Zuverlässigkeitsmechanismen in Form von Modulen in die MAAM-Architektur eingegangen. Dabei werden in den entsprechenden Abschnitten gleichzeitig die von den Mechanismen zu erfüllenden Schnittstellen diskutiert, welche von den jeweiligen Oberklassen definiert werden. Unter dem Begriff der *Oberklasse von X* wird dabei diejenige Schnittstellendefinition verstanden, von welcher X in Abbildung 6.2 abgeleitet ist (erbt).

¹⁶Die genaue Bedeutung des Begriffs “Kernfunktionalität” sowie der Zweck ihrer Abtrennung von der (übergeordneten) Benutzerschnittstelle soll später im Zusammenhang mit der Umsetzung der angestrebten Systemunabhängigkeit der Architektur geklärt werden.

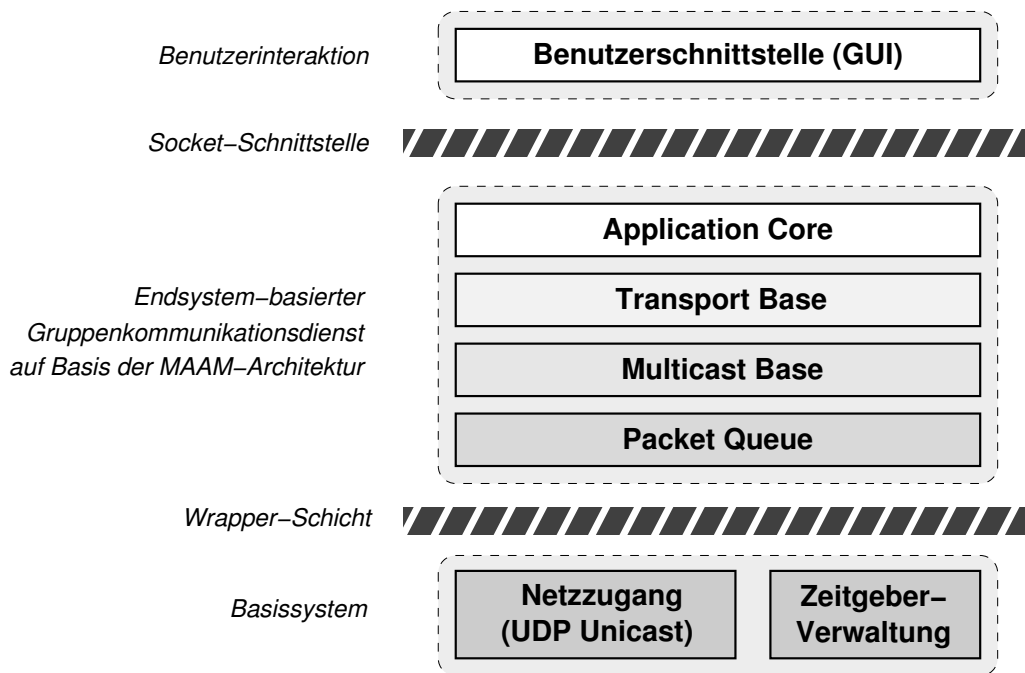


Abbildung 6.1 Der Aufbau eines Gruppenkommunikationsdienstes auf Basis der MAAM-Architektur.

6.2.1 Allgemeines zur Funktionsweise

Um die bevorstehenden Schnittstellen- und Modulbeschreibungen besser verstehen zu können, sollen zunächst einige Grundlagen bezüglich der in der MAAM-Architektur umgesetzten Konzepte geschaffen werden. Diese beinhalten insbesondere eine vereinheitlichte Paketverwaltung durch Paketwarteschlangen sowie Prozessdefinitionen zur Verarbeitung gesendeter und empfangener Datenpakete.

6.2.1.1 Agenten

Die Protokollmechanismen, welche über die in Abbildung 6.1 dargestellten Schnittstellen zugänglich sind, sind alle in die Verarbeitung der von einer Anwendung ausgesendeten Datenpakete involviert. Aus diesem Grund verfügen diese Schnittstellen über eine gemeinsame Oberklasse `DataFlowAgent`. Die Schnittstelle `DataFlowAgent` sieht zwei Methoden `::sndData(d)` und `::rcvData(d)` vor¹⁷, anhand welcher ein Datenpaket `p` zwischen den einzelnen Protokollmechanismen übergeben werden kann.

Zur Kapselung zusätzlicher, allgemeiner Information verfügt die Schnittstelle `DataFlowAgent` über eine Oberklasse `Agent`. In dieser ist eine allgemeine (später näher erläuterte)

¹⁷Die Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` werden später in Abschnitt 6.2.4 im Zusammenhang mit der Unterstützung lokaler Broadcast-Cluster behandelt.

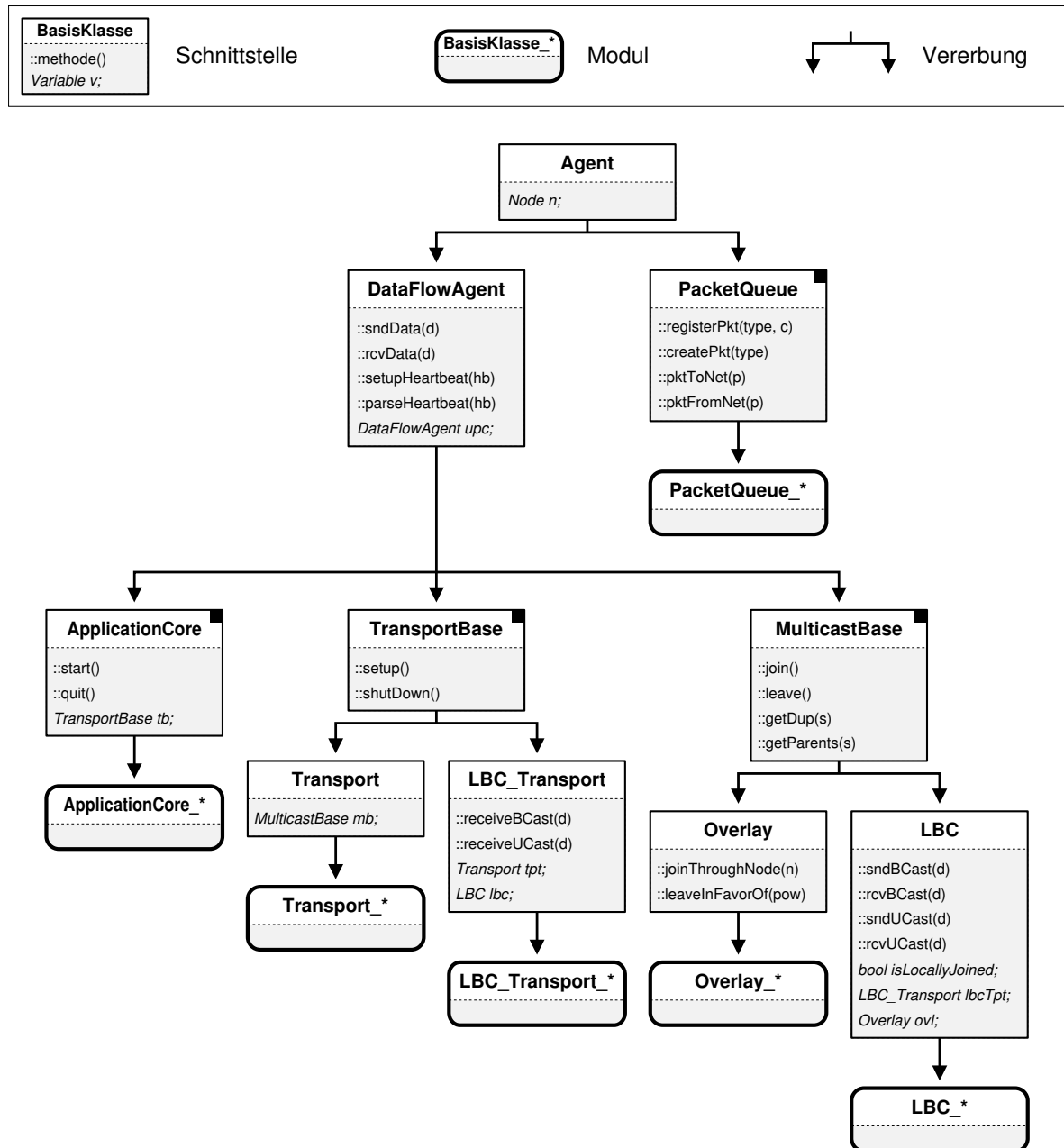


Abbildung 6.2 Vererbungsdiagramm und Schnittstellen der in die MAAM-Architektur integrierten Modulklassen.

Node-Struktur n bekannt, in welcher Endgerät-spezifische Daten abgelegt sind und über welche der Netzzugang gebildet wird.

6.2.1.2 Paketverwaltung

Die MAAM-Architektur ermöglicht die Integration von Warteschlangentechniken anhand entsprechender Paketwarteschlangen. Während genaue Umsetzungen in dieser Arbeit nicht betrachtet werden, definiert die MAAM-Architektur für Paketwarteschlangen

dennoch eine Schnittstelle `PacketQueue`, anhand welcher die Paketverwaltung vereinheitlicht und vereinfacht werden soll:

- Die Methode `::registerPkt(type, c)` nimmt die Registrierung eines Pakettyps `type` vor. Wichtig ist, dass in der Paketwarteschlange mit `type` *keinerlei* strukturelle Information des Pakets verbunden ist: Vielmehr wird der Pakettyp `type` lediglich mit einer Behandlungsroutine (*callback*) `c` assoziiert, um die später erforderliche Verarbeitung des Pakets zu vereinfachen.

- Die Methode `::createPkt(type)` erzeugt ein Paket des Typs `type`.

- Die Methode `::pktToNet(p)` veranlasst das Versenden des Pakets `p` durch Rückgriff auf die in Abschnitt 6.5 beschriebene Kapselung des Basissystems. Wichtig hierbei ist, dass die Paketwarteschlange für das Kodieren des Pakettyps `p::type` innerhalb des an das Basissystem übergebenen Pakets verantwortlich ist.

- Die Methode `::pktFromNet(p)` wird von der in Abschnitt 6.5 beschriebenen Kapselung des Basissystems aufgerufen, wenn ein Paket `p` über das Netz empfangen wird. Die Aufgabe der Paketwarteschlange ist es, das Paket in Abhängigkeit des von ihr darin kodierten Pakettyps `type` an die mit diesem Typ assoziierte Behandlungsroutine `c` zu übergeben.

6.2.1.3 Der Prozess zur Behandlung von Datenpaketen

Die MAAM-Architektur sieht vor, dass gesendete und empfangene Datenpakete von diversen Modulen sequentiell verarbeitet werden. Während an dieser Stelle der Verarbeitungsprozess zum besseren Verständnis der bevorstehenden Modulschnittstellendefinitionen nur kurz angedeutet werden soll, erfolgt in Abschnitt 6.4 eine genauere Beschreibung anhand eines Beispiels.

Ein von einer Anwendung ausgesendetes Datenpaket wird durch Zugriff auf die Schnittstelle `TransportBase` zunächst an die Zuverlässigkeitsmechanismen übergeben. Hier erfolgt eine etwaige Vergabe von Sequenznummern sowie die Duplizierung des Datenpakets zur Weiterleitung an weitere Gruppenmitglieder. Die zur Duplizierung erforderliche Information wird durch Rückgriff auf die Schnittstelle `MulticastBase` bezogen, hinter welcher sich ein `Overlay`-Modul verbirgt. Die bei der Duplizierung erhaltenen Duplikate werden wiederum der `MulticastBase` zu weiteren Verarbeitung übergeben. Obwohl dieser Schritt umständlich erscheint, ist er erforderlich, um die Unterstützung von `Overlay`-Netzen mit `Source-Routing`-Ansätzen zu unterstützen: So benötigen diese die Möglichkeit, den einzelnen Duplikaten spezielle `Source-Routing`-Paketköpfe hinzuzufügen.

Im Gegensatz hierzu wird ein über eine `Overlay`-Verbindung empfangenes Datenpaket zunächst von einem `Overlay`-Modul verarbeitet. Auf Basis der von Paketwarteschlangen umgesetzten Paketverwaltung wird der direkte Aufruf der für diese Datenpakete zustän-

digen Behandlungsroutine innerhalb des **Overlay**-Moduls ermöglicht. Um ein empfangenes Datenpaket d in Richtung der Anwendung zu übergeben, durchläuft es die gleichen Protokollkomponenten wie im Zuge des Versendens, jedoch in umgekehrter Reihenfolge. Die Paketübergabe zwischen den Komponenten erfolgt durch Aufruf von `upc::rcvData(d)` (*upc = upper protocol component*), wobei die richtige Verkettung der Protokollkomponenten ausschlaggebend für die korrekte Verarbeitung von Datenpaketen ist. Das Setzen der `upc`-Variablen in jedem `DataFlowAgent` wird daher in Abschnitt 6.3 genauer beschrieben. Bei seiner Übergabe in Richtung der Anwendung durchläuft ein Datenpaket die Zuverlässigkeitsmechanismen. In diesen wird einerseits die weitere Übergabe des Datenpakets in Richtung der Anwendung durchgeführt. Andererseits wird über eine Weiterleitung des Datenpakets zu anderen Gruppenmitgliedern entschieden: Dies erfolgt analog zum Versenden von Datenpaketen durch Duplizierung des Datenpakets anhand entsprechender, über die Schnittstelle `MulticastBase` erhaltener Information.

6.2.2 Die Integration von Overlay-Topologien

Die Integration von Protokollmechanismen zum Aufbau eines Overlay-Netzes findet durch Erweiterung der Schnittstelle `Overlay` zu einem `Overlay-Modul Overlay_*` statt. Diese müssen die von den Oberklassen `Overlay`, `MulticastBase` und `DataFlowAgent` spezifizierten Schnittstellen umsetzen:

- Die Methode `::join()` löst den Gruppenbeitritt des Endgeräts zu dem implementierten `Overlay-Netz` aus. An diesen Aufruf schließt sich folglich die Lokalisation umliegender Gruppenmitglieder, die Kontaktaufnahme zu diesen sowie die Eingliederung des Endgeräts in die `Overlay-Topologie` an.
- Die Methode `::leave()` stößt den Gruppenaustritt des Endgeräts aus dem implementierten `Overlay-Netz` an. Nach dem Aufruf dieser Methode beginnt das Endgerät damit, seinen Austrittswunsch an die innerhalb der `Overlay-Topologie` benachbarten Gruppenmitglieder zu signalisieren und seine `Overlay-Verbindungen` zu trennen.
- Die Methode `::getDup(s)` ermittelt eine Liste von Endgeräten, an welche ein von der `Multicast-Quelle s` empfangenes Datenpaket weiterzuleiten ist. Sie steht somit in direktem Zusammenhang zu dem innerhalb der `Overlay-Topologie` umgesetzten Routing-Protokoll. Diese Methode ist insbesondere deshalb erforderlich, da die eigentliche Duplizierung von Datenpaketen zur Weiterleitung über das `Overlay` nicht durch das `Overlay-Routing`, sondern durch die Zuverlässigkeitsmechanismen erfolgen muss.
- Die Methode `::getParents(s)` wiederum ermittelt eine Liste von innerhalb der `Overlay-Topologie` benachbarten Endgeräte, von welchen Datenpakete einer bestimmten `Multicast-Quelle s` bezogen werden können. Die Methode ist erforderlich, um den Zuverlässigkeitsmechanismen einen Anhaltspunkt zu bieten, von welchen Gruppenmitgliedern gegebenenfalls Übertragungswiederholungen verlorener Datenpakete angefordert werden

können. Während nicht jede Overlay-Topologie derartige Rückschlüsse erlaubt, sind an dieser Stelle dennoch geeignete Methoden zu implementieren. So kann es sich bei dem zurückgegebenen Gruppenmitglied beispielsweise um dasjenige Gruppenmitglied handeln, von welchem das letzte von `s` stammende Datenpaket empfangen wurde.

- Die Methode `::sndData(d)` wird von einem Zuverlässigkeitsmechanismus aufgerufen, um ein Datenpaket `d` über das Overlay zu versenden. Nach Verarbeitung des Datenpakets durch das Overlay-Modul, wird es letztendlich durch Aufruf von `n::pq::pktToNet(d)` der Paketwarteschlange übergeben und durch diese versendet.

- Bei der Methode `::rcvData(d)` handelt es sich um die Behandlungsroutine für Datenpakete, welche über eine Overlay-Verbindung empfangen werden. Sie wird demnach von der Paketwarteschlange aufgerufen, sobald diese ein solches Datenpaket über die drahtlose Kommunikationsschnittstelle empfangen hat. Nach seiner Verarbeitung durch das Overlay-Modul wird das Datenpaket durch Aufruf von `upc::rcvData(d)` in Richtung der Anwendung übergeben.

- Die Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` sind für die Erstellung bzw. die Verarbeitung von Heartbeat-Paketen verantwortlich. Ihre Verwendung wird im Zusammenhang mit der Unterstützung des LBC-Konzepts in Abschnitt 6.2.4 erläutert.

6.2.3 Die Integration von Zuverlässigkeitsmechanismen

Die Integration von Zuverlässigkeitsmechanismen findet durch eine Erweiterung der Schnittstelle `Transport` zu `Transport-Modulen` `Transport_*` statt. Ähnlich zu `Overlay-Modulen` sind `Transport-Module` dazu verpflichtet, die von der Schnittstellen `Transport`, `TransportBase` und `DataFlowAgent` vorgegebenen Methoden umzusetzen:

- Die Methode `::setup()` wird zur Initialisierung eines `Transport-Moduls` aufgerufen. Die Aufgabe dieser Methode ist nicht nur auf die Erzeugung und Initialisierung eventuell erforderlicher Paketpuffer (`PacketBuffer`) `pb` beschränkt, sondern beinhaltet auch den Verbindungsaufbau zur Multicast-Gruppe. Während dieser üblicherweise aus der Veranlassung des Overlay-Beitritts besteht, kann er durch Aufruf der Methode `mb::join()` der bekannten `MulticastBase` `mb` ausgelöst werden.

- Die Methode `::shutdown()` veranlasst im Gegenzug zur Methode `::setup()` den Verbindungsabbau zur Multicast-Gruppe. Während dieser durch den Aufruf der Methode `mb::leave()` der bekannten `MulticastBase` `mb` initiiert werden kann, ist die Methode `::shutdown()` ebenso dafür zuständig, vorhandene Paketpuffer `pb` zu räumen. Darüber hinaus sind eventuell zustandsbehaftete Verbindungen zu anderen Gruppenmitgliedern Protokoll-konform abzubauen.

- Die Methode `::sndData(d)` wird von einer Anwendung aufgerufen, um ein Datenpaket `d` zu versenden. Die Aufgabe des `Transport-Moduls` besteht in der Duplizierung und

Weiterleitung des Datenpakets. Die Nachfolger, an welche Duplikate d_i weiterzuleiten sind, werden durch Aufruf von `mb::getDup(d::ms)` bestimmt, wobei `d::ms` die in `d` eingetragene Adresse der Multicast-Quelle (multicast-source, `ms`) des Datenpakets ist. Die Duplikate selbst werden durch Aufruf von `mb::sndData(d_i)` versendet.

- Die Methode `::rcvData(d)` wird über `upc::rcvData(d)` von einem untergeordneten Modul aufgerufen, um ein empfangenes Datenpaket in Richtung der Anwendung zu leiten. Innerhalb dieser Methode muss das `Transport`-Modul (analog zum Versenden eines Datenpakets) die Duplizierung und Weiterleitung des Datenpakets übernehmen. Darüber hinaus ist das Datenpaket durch Aufruf von `upc::rcvData(d)` weiter in Richtung der Anwendung zu leiten.

- Analog zu `Overlay`-Modulen sind die Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` für die Erstellung bzw. die Verarbeitung von Heartbeat-Paketen verantwortlich. Ihre Verwendung wird in Abschnitt 6.2.4 im Zusammenhang mit der Integration des LBC-Konzepts erläutert.

6.2.4 Die Integration lokaler Broadcast-Cluster

Die MAAM-Architektur integriert das in Abschnitt 5.1.1 entworfene LBC-Konzept. Die Integration erfolgt dabei derart, dass seine Verwendung sowohl transparent als auch optional ist. Eines der bei der LBC-Integration verfolgten Ziele ist es, zusätzlichen Aufwand in `Overlay`- und `Transport`-Modulen weitgehend zu vermeiden.

Die Umsetzung des LBC-Konzepts schlägt sich daher in zwei zusätzlichen Schnittstellen nieder: Die Schnittstelle `LBC`, welche die Schnittstelle `MulticastBase` als Oberklasse besitzt, und die Schnittstelle `LBC_Transport`, welche die Schnittstelle `TransportBase` erweitert. An dieser Stelle können die für die `Transport`- und `Overlay`-Module vorgenommenen Schnittstellenabstraktionen `TransportBase` und `MulticastBase` begründet werden: Diese ermöglichen die geforderte Transparenz der LBC-Nutzung. So interagiert eine Anwendung bei Zugriff auf die Schnittstelle `TransportBase` fortan nicht mehr mit einem `Transport`- sondern mit einem `LBC_Transport`-Modul. Analog hierzu interagiert ein `Transport`-Modul bei Zugriff auf die Schnittstelle `MulticastBase` nicht mehr mit einem `Overlay`- sondern mit einem `LBC`-Modul.

Die Integration des LBC-Konzepts veranlasst dem `Overlay` beigetretene Gruppenmitglieder periodische Heartbeat-Pakete zu versenden. Innerhalb der MAAM-Architektur sollen diese Heartbeats jedoch nicht nur zur Signalisierung der Präsenz eines LBC-Repräsentanten dienen. Vielmehr sollen alle an der Verarbeitung von Datenpaketen beteiligten Module die Möglichkeit erlangen, Information innerhalb der Heartbeats zu kodieren. Dies wird über die in der Schnittstelle `DataFlowAgent` eingeführten Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` ermöglicht, welche Information in ein Heartbeat `hb` einbetten oder aus diesem extrahieren.

6.2.4.1 Die Erweiterung der MulticastBase

Für die Umsetzung des LBC-Konzepts verfügt die Schnittstelle LBC über das Wissen, welches Overlay-Modul `ovl` verwendet wird. Anhand dieser kann das LBC-Modul das Verhalten des Overlay-Moduls kontrollieren.

Um die zur Unterstützung des LBC-Konzepts erforderliche Erweiterung von Overlay-Modulen minimal zu halten, integriert ein LBC-Modul alle in Abschnitt 5.1.1 eingeführten Konzepte: Diese beinhalten das Broadcasten von Datenpaketen, das Versenden, der Empfang und die Verarbeitung von Heartbeats sowie die Bewertung umliegender LBC-Repräsentanten. Ebenso setzt das LBC-Modul den in Abbildung 5.3 dargestellten Zustandsautomaten um und führt somit den derzeitigen Zustand des Gruppenmitglieds (“lokaler Beitritt” oder “Overlay-Eingliederung”) innerhalb der Variable `isLocallyJoined` mit. Stellt das LBC-Modul für ein lokal beigetretenes Gruppenmitglied den Verlust des LBC-Repräsentanten fest, so könnte es durch Aufruf von `ovl::join()` die Eingliederung des Gruppenmitglieds in das Overlay anstoßen. Oftmals kann der Overlay-Beitritt jedoch deutlich beschleunigt werden, da mit dem ehemaligen lokalen Repräsentanten bereits Wissen über ein Overlay-Mitglied vorliegt. Durch Integration der Methode `::joinThroughNode(n)` in die Schnittstelle `Overlay`, wird einem Overlay-Modul der ehemalige lokale Repräsentant `n` signalisiert und hierdurch die bei einem Aufruf von `ovl::join()` zunächst durchgeführte Suche nach Gruppenmitgliedern vermieden.

Analog hierzu kann ein LBC-Modul durch Aufruf von `ovl::leave()` den Overlay-Austritt eines Gruppenmitglieds veranlassen: Dieses Verhalten wird von dem LBC-Konzept bei der Kollision zweier LBCs gefordert, welche innerhalb eines LBC-Moduls über den Empfang eines Heartbeat durch ein Overlay-Mitglied detektiert wird. Da der Overlay-Austritt für Gruppenmitglieder unterschiedlichen Aufwand zeigen kann, soll über den tatsächlichen Austritt eines Overlay-Mitglieds gezielt entschieden werden. Hierfür bietet sich die Integration zusätzlicher Information in den periodisch versendeten Heartbeats an. Versendet das LBC-Modul ein Heartbeat `hb`, so veranlasst es durch Aufruf von `ovl::setupHeartbeat(hb)` das Overlay-Modul, eine Größe `pow` innerhalb des Heartbeat zu kodieren, welche den Aufwand eines Overlay-Austritts beschreibt. Entsprechend hierzu gewinnt das LBC-Modul bei Empfang eines Heartbeat `hb` die darin enthaltene Information `pow` durch den Aufruf von `ovl::parseHeartbeat(hb)` zurück. Ob der eigene Overlay-Austritt aufgrund eines geringeren Aufwands im Vergleich zum kollidierenden Repräsentanten letztendlich stattfinden soll, bringt ein LBC-Modul durch den Aufruf der Methode `ovl::leaveInFavorOf(pow)` in Erfahrung. Findet der Austritt statt, so ist das LBC-Modul dazu verpflichtet, die Aufgabe des LBCs anhand eines entsprechenden, als Broadcast versendeten `AbandonCluster`-Pakets (vgl. Abschnitt 5.1.1.4) zu signalisieren.

Zur korrekten Verarbeitung und Übergabe von Datenpaketen zwischen einzelnen Protokollkomponenten ist es wichtig, die Art und Weise zu unterscheiden, wie ein Daten-

paket empfangen wurde: So können Datenpakete 1. über eine Overlay-Verbindung oder 2. als Broadcast eines LBC-Repräsentanten oder 3. als Unicast von einem lokal beigetretenen Gruppenmitglied an einen LBC-Repräsentanten empfangen werden. Wie in Abschnitt 6.4 verdeutlicht wird, ist je nach Art des Empfangs ein Datenpaket unterschiedlich zu behandeln. Durch die von Paketwarteschlangen gebotene Funktionalität ist die Unterscheidung der Empfangsarten einfach möglich: So kann für jede Empfangsart ein eigener Pakettyp mit der entsprechenden Behandlungsroutine registriert werden.

Ein wichtiger Aspekt ist die Tatsache, dass ein **Transport**-Modul beim Zugriff auf die Schnittstelle **MulticastBase** nun nicht mit einem **Overlay**- sondern mit einem **LBC**-Modul interagiert. Die von den Oberklassen vorgegebenen Schnittstellen integriert ein **LBC**-Modul daher wie folgt:

- Die Methode `::join()` wird von einem **Transport**-Modul aufgerufen, um den **Overlay**-Beitritt anzustoßen. Dieser Aufruf muss von einem **LBC**-Modul abgefangen werden, da nach Integration des **LBC**-Konzepts der in Abbildung 5.3 dargestellte, abgewandelte Bootstrapping-Prozess verwendet wird. Anstatt den **Overlay**-Beitritt zu bewerkstelligen, wird zunächst die Anwesenheit eines umliegenden **LBC**-Repräsentanten geprüft. Schlägt dies fehl, stößt das **LBC**-Modul wiederum den **Overlay**-Beitritt durch Aufruf von `ovl::join()` an.

- Ebenso wird die Methode `::leave()` von einem **Transport**-Modul aufgerufen, um den **Overlay**-Austritt anzustoßen. In Abhängigkeit des derzeitigen Beitrittsstatus des Gruppenmitglieds (“lokaler Beitritt” oder “**Overlay**-Eingliederung”), muss das **LBC**-Modul den Aufruf entweder abfangen und den lokalen Gruppenaustritt bewerkstelligen¹⁸ oder ihn durch Aufruf von `ovl::leave()` an das **Overlay**-Modul weiterleiten.

- Die Methode `::getDup(s)` wird von einem **Transport**-Modul aufgerufen, um Information über die Duplizierung von Datenpaketen zu ermitteln. Ist das Gruppenmitglied lokal beigetreten, liefert das **LBC**-Modul eine leere Liste zurück, da für lokal beigetretene Gruppenmitglieder keine Weiterleitung über das **Overlay**-Netz erfolgt. Ist das Gruppenmitglied hingegen in das **Overlay** eingegliedert, ist eine Weiterleitung innerhalb des **Overlay** erforderlich: Die in diesem Fall zurückzugebende Liste wird durch Aufruf von `ovl::getDup(s)` bestimmt.

- Analog zu `::getDup(s)` wird auch die Methode `::getParents(s)` von einem **Transport**-Modul aufgerufen. Ist das Gruppenmitglied lokal beigetreten, liefert das **LBC**-Modul eine leere Liste zurück, da für lokal beigetretene Gruppenmitglieder gesonderte Zuverlässigkeitsmechanismen verwendet werden. Andernfalls wird die durch Aufruf von `ovl::getParents(s)` erhaltene Liste zurückgegeben.

¹⁸Da lokal beigetretene Gruppenmitglieder den übrigen Gruppenmitgliedern unbekannt sind, sind im Falle eines lokalen Gruppenaustritts keine weiteren Handlungen erforderlich.

– Die Methode `::sndData(d)` wird von einem `Transport`-Modul aufgerufen, um ein Datenpaket `d` über eine Overlay-Verbindung zu senden. Aus diesem Grund übergibt das LBC-Modul durch Aufruf von `ovl::sndData(d)` das Datenpaket an das `Overlay`-Modul.

– Die Methode `::rcvData(d)` wird von einem untergeordneten Modul (welches in diesem Fall lediglich ein `Overlay`-Modul sein kann) über `upc::rcvData(d)` im Laufe der Übergabe eines Datenpakets `d` in Richtung der Anwendung aufgerufen. Da ein LBC-Modul über eine Overlay-Verbindung empfangene Datenpakete nicht verarbeitet, übergibt es das Datenpaket durch Aufruf von `upc::rcvData(d)` weiter in Richtung der Anwendung.

– Die Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` sind für das Versenden bzw. das Empfangen von Heartbeats `hb` zuständig. Beim Versenden wird das Einbetten der Information der übrigen Module durch den Aufruf der dort implementierten Methoden `::setupHeartbeat(hb)` veranlasst. Analog hierzu ist diese Information bei Empfang eines Heartbeat durch Aufrufen von `::parseHeartbeat(hb)` in den jeweiligen Modulen zurück zu gewinnen.

– Die Methode `::sndBCast(d)` wird von einem `LBC_Transport`-Modul für Overlay-Mitglieder aufgerufen, um ein Datenpaket `d` als Broadcast an lokal beigetretene Gruppenmitglieder zu leiten.

– Bei der Methode `::rcvBCast(d)` handelt es sich um die Behandlungsroutine für Datenpakete, welche lokal beigetretene Gruppenmitglieder als Broadcast von ihrem LBC-Repräsentanten empfangen. Ist das Paket verarbeitet, wird es durch Aufruf von `lbcTpt::rcvBCast(d)` einem `LBC_Transport`-Modul übergeben.

– Die Methode `::sndUCast(d)` wird von einem `LBC_Transport`-Modul für lokal beigetretene Gruppenmitglieder aufgerufen, um ein Datenpaket `d` als Unicast an den LBC-Repräsentanten zu senden.

– Bei der Methode `::rcvUCast(d)` handelt es sich um die Behandlungsroutine für Datenpakete, welche Overlay-Mitglieder als Unicast von einem lokal beigetretenen Gruppenmitglied empfangen. Ist das Paket verarbeitet, wird es durch Aufruf von `lbcTpt::rcvUCast(d)` einem `LBC_Transport`-Modul übergeben.

6.2.4.2 Die Erweiterung der `TransportBase`

Insbesondere im Hinblick auf die Umsetzung von Zuverlässigkeitsmechanismen lässt sich bei der Integration des LBC-Konzepts ein Mehraufwand nicht vermeiden: Dies ergibt sich aufgrund der Tatsache, dass `Transport`-Module lediglich Zuverlässigkeit für Ende-zu-Ende-Kommunikation über Overlay-Verbindungen umsetzen. Für eine Datenverteilung durch Broadcast-Übertragungen sind folglich gesonderte Zuverlässigkeitsmechanismen nicht nur erforderlich sondern auch explizit erwünscht. Um `Transport`-Module unmodifiziert zu lassen, findet die Integration von Zuverlässigkeitsmechanismen für

LBCs durch Erweiterung der Schnittstelle `LBC_Transport` zu `LBC_Transport-Modulen` `LCB_Transport_*` statt. Ein wichtiger Aspekt hierbei ist die Tatsache, dass eine Anwendung bei Zugriff auf die Schnittstelle `TransportBase` nun nicht mit einem `Transport`- sondern mit einem `LBC_Transport-Modul` interagiert. Die vorgegebenen Schnittstellen integriert ein `LBC_Transport-Modul` daher wie folgt:

- Die Methode `::setup()` wird von einem Anwendungsmodul aufgerufen, um ein `Transport-Modul` zu initialisieren und eine Verbindung zur Multicast-Gruppe aufzubauen. Aus diesem Grund wird der Aufruf von einem `LBC_Transport-Modul` zu einem Aufruf von `tpt::setup()` übersetzt. Zu bemerken ist hierbei, dass das `Transport-Modul` daraufhin den Aufruf von `mb::join()` tätigt, wobei sich hinter `mb` ein `LBC-Modul` verbirgt.

- Die Methode `::shutdown()` wird von einem Anwendungsmodul aufgerufen, um die zu einer Multicast-Gruppe bestehende Verbindung abzubauen. Der Aufruf wird von einem `LBC_Transport-Modul` daher zu einem Aufruf von `tpt::shutdown()` übersetzt. Ähnlich zu oben erfolgt in diesem zur Einleitung des Gruppenaustritts ein Aufruf von `mb::leave()`, wobei sich hinter `mb` ein `LBC-Modul` verbirgt.

- Die Methode `::sndData(d)` wird von einem Anwendungsmodul aufgerufen, um ein Datenpaket zu versenden. Die Aufgabe des `LBC_Transport-Moduls` besteht in der Duplizierung und Weiterleitung des Datenpakets. Hierzu ist einerseits das Datenpaket durch Aufruf von `tpt::sndData(d)` dem `Transport-Modul` zu übergeben. Darüber hinaus muss anhand der Variable `lbc::isLocallyJoined` der aktuelle Beitrittsstatus des Gruppenmitglieds geprüft werden. Handelt es sich um ein lokal beigetretenes Gruppenmitglied, ist das Datenpaket durch Aufruf von `lbc::sndUCast(d)` via Unicast an den derzeitigen `LBC-Repräsentanten` zu senden. Handelt es sich hingegen um ein `Overlay-Mitglied`, ist das Datenpaket durch Aufruf von `lbc::sndBCast(d)` via Broadcast an eventuell lokal beigetretene Gruppenmitglieder weiterzuleiten.

- Die Methode `::rcvData(d)` wird über `upc::rcvData(d)` von einem untergeordneten Modul aufgerufen, um ein über das `Overlay` empfangenes Datenpaket in Richtung Anwendung zu leiten. Einerseits ist das Datenpaket durch Aufruf von `upc::rcvData(d)` weiter in Richtung Anwendung zu leiten. Andererseits ist das Datenpaket an die lokal beigetretenen Gruppenmitglieder als Broadcast zu senden. Dies erfolgt durch Aufruf von `lbc::sndBCast(d)`.

- Die Methode `::rcvBCast(d)` wird von einem `LBC-Modul` für ein als Broadcast von einem `LBC-Repräsentanten` empfangenes Datenpaket aufgerufen. Einerseits ist `d` durch Aufruf von `upc::rcvData(d)` in Richtung der Anwendung zu leiten. Andererseits ist `d` durch Aufruf von `tpt::sndData(d)` dem `Transport-Modul` zu übergeben. Obwohl es sich bei dem Gruppenmitglied offensichtlich um ein lokal beigetretenes Gruppenmitglied handelt, ist dieser Schritt erforderlich, um das Transportprotokoll über das Vorhandensein des Datenpakets `d` zu informieren.

– Die Methode `::rcvUCast(d)` wird von einem LBC-Modul für ein als Unicast von einem lokalen Gruppenmitglied empfangenes Datenpaket aufgerufen. Da es sich bei dem Gruppenmitglied demnach um ein Overlay-Mitglied handelt, ist das Datenpaket sowohl als Broadcast (`lbc::sndBCast(d)`) als auch über das Overlay (`tpt::sndData(d)`) zu versenden. Zusätzlich ist das Datenpaket durch Aufruf von `upc::rcvData(d)` in Richtung der Anwendung zu leiten.

– Die Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` dienen der Einbettung und der Zurückgewinnung von Information, welche in die von LBC-Repräsentanten periodisch ausgesendeten Heartbeats eingebettet ist.

6.3 Die Dienstkomposition

Die MAAM-Architektur sieht die Berücksichtigung von Anwendungsanforderungen durch auf Anwendungen zurechtgeschnittene Zuverlässigkeitsmechanismen und Overlay-Topologien vor. Diese Protokolle sind innerhalb entsprechender Module zu integrieren und zur Laufzeit zu einem Gruppenkommunikationsdienst zu vereinen. Da die Anwendung ihre Anforderungen am besten kennt, wird die Aufgabe der Modulauswahl und der Komposition des Gruppenkommunikationsdienstes ihr selbst überlassen. Dies erfolgt insbesondere, indem einzelne Module instanziiert und durch Setzen der durch die Schnittstellen vorgegebenen Variablen verbunden werden.

6.3.1 Ohne Verwendung des LBC-Konzepts

Erfolgt keine Verwendung des LBC-Konzepts, so wird der Gruppenkommunikationsdienst aus einem `ApplicationCore`-Modul `appCore`, einem `Transport`-Modul `tpt` und einem `Overlay`-Modul `ovl` zusammengefügt. Zur korrekten Verbindung der Module sind von der Anwendung dabei die folgenden Zuweisungen durchzuführen:

```
appCore::tb ← tpt      # von der Anwendung genutzte TransportBase
tpt::mb ← ovl         # vom Transport genutzte MulticastBase
tpt::upc ← appCore   # Leitung empfangener Datenpakete zur Anwendung
ovl::upc ← tpt       # Leitung empfangener Datenpakete zur Anwendung
```

6.3.2 Mit Verwendung des LBC-Konzepts

Wird der Einsatz des LBC-Konzepts gewünscht, so wird der Gruppenkommunikationsdienst aus einem `ApplicationCore`-Modul `appCore`, einem `Transport`-Modul `tpt`, einem `LBC_Transport`-Modul `lbcTpt`, einem `Overlay`-Modul `ovl` und einem `LBC`-Modul `lbc` zusammengefügt. Zur korrekten Verbindung der Module sind von der Anwendung dabei die folgenden Zuweisungen durchzuführen:


```

appCore::tb ← lbcTpt  # von der Anwendung genutzte TransportBase
lbcTpt::tpt ← tpt    # vom LBC_Transport kontrollierter Transport
lbcTpt::lbc ← lbc    # vom LBC_Transport verwendeter LBC
lbcTpt::upc ← appCore # Leitung empfangener Datenpakete zur Anwendung
tpt::mb ← lbc      # vom Transport genutzte MulticastBase
tpt::upc ← lbcTpt  # Leitung empfangener Datenpakete zur Anwendung
lbc::lbcTpt ← lbcTpt # vom LBC verwendeter LBC_Transport
lbc::ovl ← ovl     # vom LBC kontrolliertes Overlay
lbc::upc ← tpt     # Leitung empfangener Datenpakete zur Anwendung
ovl::upc ← lbc     # Leitung empfangener Datenpakete zur Anwendung

```

6.4 Versand und Weiterleitung von Daten

Um das Zusammenspiel der einzelnen Module zur Verarbeitung, Duplizierung und Weiterleitung von Datenpaketen zu verdeutlichen, sollen nun einfache Beispiele diskutiert werden. In Abschnitt 6.4.1 wird zunächst die Verarbeitung von Datenpaketen verdeutlicht, welche von einer Anwendung auf einem Gruppenmitglied selbst verschickt werden. Dagegen wird in Abschnitt 6.4.2 die Verarbeitung von Datenpaketen erläutert, welche von anderen Gruppenmitgliedern empfangen werden. Die gezeigten Beispiele beschränken sich auf die Fälle, bei welchen das LBC-Konzept zum Einsatz kommt.

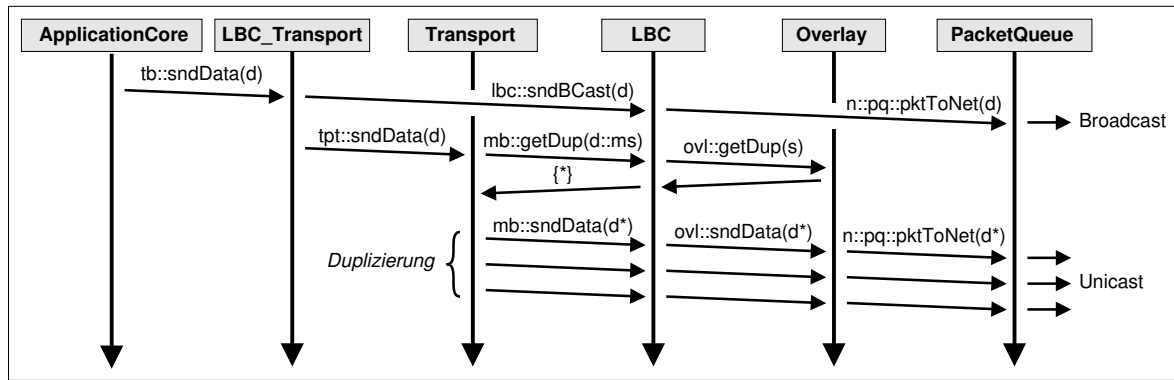
6.4.1 Die Verarbeitung eines versendeten Datenpakets

Die einzelnen Schritte, anhand derer ein selbst versendetes Datenpaket verarbeitet wird, sind in den in Abbildung 6.3 dargestellten Weg-Zeit-Diagrammen aufgetragen.

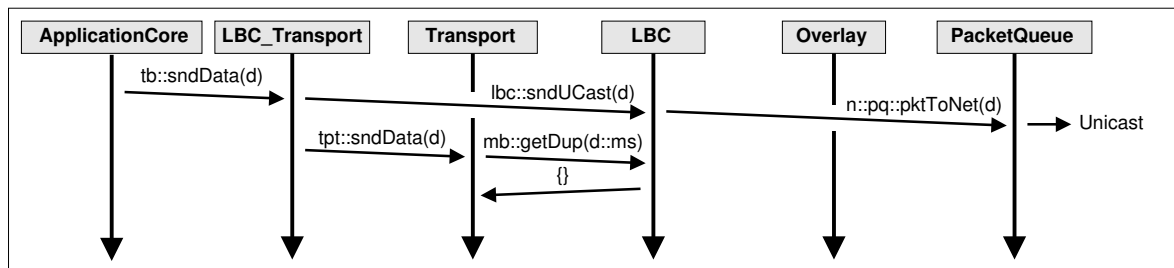
Der Fall, bei welchem es sich bei dem Gruppenmitglied um ein Overlay-Mitglied handelt, ist in Abbildung 6.3.a) dargestellt. Hier wird von einer Anwendung zunächst ein Datenpaket *d* an die Zuverlässigkeitsmechanismen durch Aufruf von `tb::sndData(d)` übergeben. Hinter der `TransportBase`-Schnittstelle befindet sich in diesem Fall ein `LBC_Transport`-Modul, welches durch Aufruf von `lbc::sndBCast(d)` das Datenpaket zunächst an das `LBC`-Modul übergibt. Das `LBC`-Modul sendet das Datenpaket als Broadcast an die lokal beigetretenen Gruppenmitglieder indem es die Broadcast-Zieladresse in das Datenpaket einträgt¹⁹ und dieses der Paketwarteschlange durch Aufruf von `n::pq::pktToNet(d)` übergibt. Da das Gruppenmitglied dem Overlay beigetreten ist, muss zusätzlich eine Weiterleitung des Datenpakets über dieses erfolgen. Hierzu übergibt das `LBC_Transport`-Modul das Datenpaket dem `Transport`-Modul durch Aufruf von `tpt::sndData(d)`. Das `Transport`-Modul ist zuständig für die Duplizierung des Datenpakets, so dass es durch Aufruf von `mb::getDup(d:ms)` eine Liste der Nachfolger bestimmt. Da sich hinter der `MulticastBase`-Schnittstelle ein `LBC`-Modul verbirgt, leitet dieses den Aufruf via `ovl::getDup(s)` an

¹⁹Dies ist in der Abbildung nicht dargestellt, da nur Methodenaufrufe aufgetragen sind.

6. Die MAAM-Architektur – Konzepte und Umsetzungen



a) Behandlung eines versendeten Datenpakets d für Overlay-Mitglieder



b) Behandlung eines versendeten Datenpakets d für lokal beigetretene Gruppenmitglieder

Abbildung 6.3 Übergabe eines zu versendenden Datenpakets zwischen den einzelnen Modulen der MAAM-Architektur.

das Overlay-Modul weiter und gibt die erhaltene Adressenliste zurück. Auf Basis dieser Information beginnt die Duplizierung des Datenpakets innerhalb des Transport-Moduls. Die einzelnen Duplikate d^* können an dieser Stelle mit entsprechenden, individuellen Paketköpfen und Zieladressen versehen werden und werden im Anschluss durch Aufruf von `mb::sndData(d*) der MulticastBase-Schnittstelle übergeben. Das sich dahinter befindende LBC-Modul leitet die Duplikate durch Aufruf von ovl::sndData(d*) an das Overlay-Modul weiter. Auch dieses kann die Duplikate gegebenenfalls mit individuellen Paketköpfen versehen, wie sie beispielsweise von Source-Routing-Overlays gefordert werden. Abschließend werden die Duplikate durch Aufruf von n::pq::pktToNet(d*) der Paketwarteschlange übergeben, welche sie daraufhin versendet. Hiermit ist die Verarbeitung des Datenpakets d abgeschlossen.`

Handelt es sich bei dem Gruppenmitglied um ein lokal beigetretenes Gruppenmitglied, so folgt die Verarbeitung des Datenpakets d dem in Abbildung 6.3.b) gezeigten Diagramm. Auch hier wird zunächst ein Datenpaket d von der Anwendung an die Zuverlässigkeitsmechanismen durch Aufruf von `tb::sndData(d)` übergeben. Das dabei aufgerufene LBC_Transport-Modul ist dafür zuständig, das Datenpaket als Unicast an den LBC-Repräsentanten zu senden. Dies erfolgt durch Aufruf von `lbc::sndUCast(d)`, woran sich die

Übergabe des Datenpakets an die Paketwarteschlange anschließt. Abschließend übergibt das `LBC_Transport`-Modul das Datenpaket `d` dem `Transport`-Modul durch Aufruf von `tpt::sndData(d)`. Dies ist erforderlich, um das `Transport`-Modul über das Vorhandensein des Datenpakets in Kenntnis zu setzen, da es hier nach einem späteren Overlay-Beitritt eventuell zur Behebung von Paketverlusten anderer Gruppenmitglieder benötigt wird. Da das `Transport`-Modul kein Wissen über den lokalen Beitritt des Gruppenmitglieds hat, versucht es, das erhaltene Datenpaket über das Overlay weiterzuleiten. Der zur Erlangung der Duplikationsinformation durchgeführte Aufruf von `mb::getDup(d::ms)` wird jedoch durch das `LBC`-Modul abgefangen und durch Rückgabe einer leeren Liste quittiert. Folglich nimmt das `Transport`-Modul keine weitere Duplizierung vor.

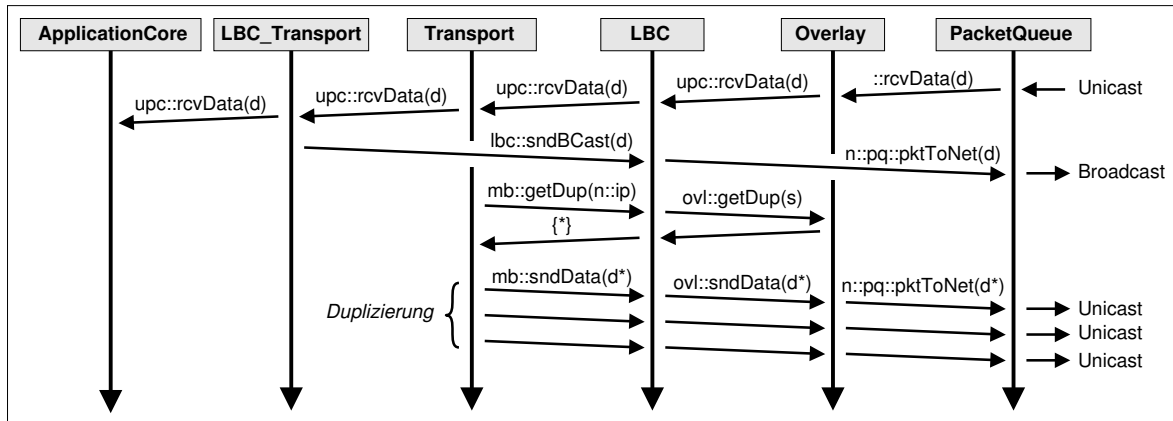
6.4.2 Die Verarbeitung eines empfangenen Datenpakets

Die Verarbeitungsschritte für ein empfangenes Datenpaket `d` lassen sich in zwei Phasen unterteilen: Während die erste Phase aus der Übergabe des Datenpakets zur Anwendung besteht, wird in der zweiten Phase eine eventuelle Duplizierung und Weiterleitung des Datenpakets vorgenommen. In Abbildung 6.4 sind die einzelnen Verarbeitungsvarianten für empfangene Datenpakete in entsprechenden Weg-Zeit-Diagrammen dargestellt.

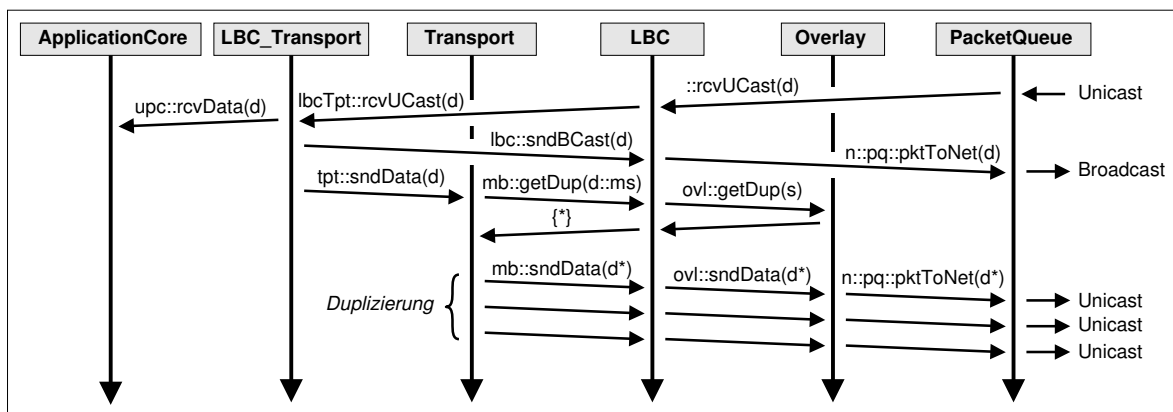
Abbildung 6.4.a) veranschaulicht den Fall, in welchem ein Overlay-Mitglied ein Datenpaket über eine Overlay-Verbindung empfängt. So trifft hier das Datenpaket zunächst als Unicast in der Paketwarteschlange ein. Die mit diesem Pakettyp assoziierte Behandlungsroutine ist die Methode `::rcvData(d)` des `Overlay`-Moduls, so dass das Datenpaket dieser Methode übergeben wird. Hier wird es durch Aufruf von `upc::rcvData(d)` bis zur Anwendung übergeben. Nach abschließender Übergabe durch das `LBC_Transport`-Modul beginnt die Weiterleitung des Datenpakets. Hierzu wird das Datenpaket zunächst als Broadcast durch das `LBC_Transport`- und das `LBC`-Modul an die lokal beigetretenen Gruppenmitglieder gesendet. Anschließend übernimmt das `Transport`-Modul, analog zu Abbildung 6.3.a), die Duplizierung und Weiterleitung des Datenpakets. Zu bemerken ist hier, dass das Datenpaket zur Duplizierung und Weiterleitung nicht erst durch das `LBC_Transport`- an das `Transport`-Modul übergeben werden muss, da dieses das Datenpaket nach der Übergabe in Richtung Anwendung automatisch weiter verarbeitet.

Der Fall, bei welchem ein Overlay-Mitglied ein Datenpaket `d` von einem lokal beigetretenen Gruppenmitglied via Unicast empfängt, ist in Abbildung 6.4.b) dargestellt. Hier trifft das Datenpaket zunächst als Unicast in der Paketwarteschlange ein, wobei nun die mit diesem Pakettyp assoziierte Behandlungsroutine der Methode `::rcvUCast(d)` des `LBC`-Moduls entspricht. Nach Erhalt und Verarbeitung des Datenpakets übergibt das `LBC`-Modul das Datenpaket durch Aufruf von `lbcTpt::rcvUCast(d)` an das `LBC_Transport`-Modul. Hier wird es durch Aufruf von `upc::rcvData(d)` zunächst an die Anwendung übergeben. Anschließend beginnt die Duplizierung und Weiterleitung des

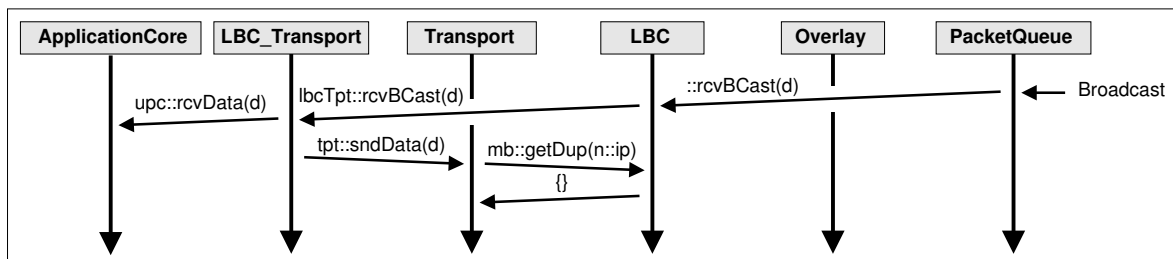
6. Die MAAM-Architektur – Konzepte und Umsetzungen



a) Behandlung eines über das Overlay empfangenen Datenpakets d



b) Behandlung eines von einem lokalen Gruppenmitglied empfangenen Datenpakets d



c) Behandlung eines vom einem LBC-Repräsentanten empfangenen Datenpakets d

Abbildung 6.4 Übergabe eines empfangenen Datenpakets zwischen den einzelnen Modulen der MAAM-Architektur.

Datenpakets. Der Ablauf hierfür ist identisch zu Abbildung 6.3.a): Zunächst wird das Datenpaket als Broadcast an lokal beigetretene Gruppenmitglieder geleitet und anschließend über das Overlay verteilt. Da das Datenpaket von einem lokal beigetretenen Gruppenmitglied empfangen wurde und unter Ausschluss des Transport-Moduls zur Anwendung geleitet wurde, muss es diesem zur Weiterleitung über das Overlay explizit übergeben werden.

Der letzte Fall, bei welchem ein lokal beigetretenes Gruppenmitglied ein Datenpaket via Broadcast von seinem LBC-Repräsentanten empfängt, ist in Abbildung 6.4.c) aufgetragen. Hier trifft das Datenpaket als Broadcast in der Paketwarteschlange ein, wobei nun die mit diesem Pakettyp assoziierte Behandlungsroutine der Methode `::rcvBCast(d)` des LBC-Moduls entspricht. Von diesem aus wird das Datenpaket durch Aufruf von `lbcTransport::rcvBCast(d)` dem `LBC_Transport`-Modul übergeben, welches es an die Anwendung übergibt. Zum Abschluss der Verarbeitung übergibt das `LBC_Transport`-Modul das Datenpaket dem `Transport`-Modul. Analog zu Abbildung 6.3.b) verhindert das LBC-Modul eine weitere Duplizierung und Weiterleitung des Datenpakets.

6.5 Die Kapselung des Basissystems

Eines der bei der Konzeption der MAAM-Architektur verfolgten Ziele ist es, den Betrieb der Architektur sowie der darin entwickelten Protokolle weitgehend unabhängig von einer bestimmten Betriebsumgebung zu gestalten. Wird diese Unabhängigkeit gewährleistet, so können Protokollumsetzungen einerseits zu simulativen Evaluationszwecken innerhalb von Netzwerksimulationsumgebungen betrieben und andererseits in Anwendungen auf realen Endgeräten experimentell eingesetzt werden.

Um eine solche Unabhängigkeit zu erreichen, sind zunächst die System-nahen Interaktionen zu identifizieren, aufgrund welcher Protokollumsetzungen üblicherweise abhängig von einer bestimmten Betriebsumgebung werden. Im Kontext des mit der MAAM-Architektur anvisierten Umfelds P2P-basierter Gruppenkommunikationsdienste wurden hierbei drei System-nahe Interaktionen erkannt:

- *Der Zugang zu einer Netzwerkkommunikationsschnittstelle:* Dies stellt die Grundanforderung für den Betrieb eines Kommunikationsprotokolls dar.
- *Die Verwaltung von Zeitgebern:* Diese Funktionalität wird von nahezu allen Protokollumsetzungen gefordert. So werden mit Hilfe von Zeitgebern üblicherweise Ereignisse ausgelöst, aufgrund derer Protokolle unterschiedliche Aktionen anstoßen.
- *Die Umsetzung von Benutzerschnittstellen und grafischen Ausgaben:* Funktionalität zur Umsetzung von Benutzerschnittstellen greift oftmals auf System-abhängige, grafische *Toolkits* zurück, deren Betrachtung nicht Gegenstand dieser Arbeit ist.

Unter dem Begriff eines *Basissystem* soll fortan eine Betriebsumgebung verstanden werden, welche sowohl die Verwaltung von Zeitgebern als auch den Zugang zu einer Netzwerkkommunikationsschnittstelle bietet. Wie in Abbildung 6.1 dargestellt ist, bietet ein solches Basissystem die Grundlage für den Betrieb der MAAM-Architektur. Durch Kapselung der von einem Basissystem gebotenen Funktionalität wird dieses austauschbar, so dass sich die MAAM-Architektur als sehr portabel erweist. Die für die Kapselung von Basissystemen definierten Schnittstellen sind in Abbildung 6.5 zu sehen und werden in den folgenden Abschnitten näher besprochen.

6.5.1 Die Wrapper-Schnittstelle

Die Kapselung eines Basissystems findet bei der MAAM-Architektur anhand so genannter *Wrapper* statt. Da Wrapper System-nahe Funktionen umsetzen, sind sie System-abhängig und müssen für jedes Basissystem getrennt umgesetzt werden.

Zunächst soll angemerkt werden, dass die Schnittstelle **Wrapper** die Verwaltung einer beliebigen Anzahl von Strukturvariablen **Node n[]** vorsieht. Dies ergibt sich aufgrund der Tatsache, dass Basissysteme wie Netzwerksimulationsumgebungen in der Regel eine Vielzahl von Endgeräten simulieren. Um dies zu unterstützen, muss ein Wrapper in der Lage sein, für jedes simulierte Endgerät eigene Datenstrukturen und Instanzen der in die MAAM-Architektur integrierten Protokollmodule zu halten. Die Verwaltung dieser Datenstrukturen erfolgt in Implementierungen der Schnittstelle **Node**, welche fortan auch als *Gerätekontext* bezeichnet wird. Sie ist wie folgt aufgebaut:

- Die Methode `::setup(ip, wrp)` wird von einem Wrapper aufgerufen, um eine **Node**-Implementierung mit einer bestimmten IP-Adresse **ip** zu initialisieren und ihr über **wrp** das Basissystem zugänglich zu machen. Während der Initialisierung der **Node**-Implementierung erfolgt üblicherweise die Erzeugung einer Paketwarteschlange **pq** sowie die Initialisierung eventuell mehrerer **ApplicationCore**-Module **a[]**.

- Die Methode `::shutdown()` wird im Gegensatz zu `::setup()` von einem Wrapper aufgerufen, um alle in dieser **Node**-Implementierung laufenden **ApplicationCore**-Module zu beenden und damit gegebenenfalls den Austritt des Endgeräts aus der Multicast-Gruppe zu veranlassen.

Node-Implementierungen sind selbst System-unabhängig, da sie auf ein Basissystem lediglich über den ihnen zur Verfügung stehenden Wrapper **wrp** zugreifen. Der eigentliche Zugang zu einem Basissystem findet über Implementierungen der Schnittstelle **Wrapper** statt. Die im Zuge dieser Arbeit für unterschiedliche Basissysteme entwickelten Wrapper werden im Folgenden kurz vorgestellt.

6.5.1.1 Umsetzungen für Linux und Windows

Zur Nutzung der MAAM-Architektur innerhalb realer Anwendungen wurde jeweils ein Wrapper zur Kapselung der Betriebssysteme Windows und Linux entwickelt. An dieser Stelle sollen die genauen Implementierungs-technischen Aspekte der Wrapper, wie Systemaufrufe und Socket-Programmierung, nicht vertieft werden. Es soll lediglich die Umsetzung der Schnittstelle **Wrapper** für beide Betriebssysteme kurz erläutert werden. Dabei ist zu bemerken, dass für Linux- und Windows-Wrapper die Verwaltung von nur einem einzigen Gerätekontext **n[0]** erforderlich ist.

- Die Methode `::initNode(ip)` wird von der innerhalb der Wrapper jeweils implementierten Methoden `int main(argc, **argv)` aufgerufen, wobei die IP-Adresse **ip** der drahtlosen

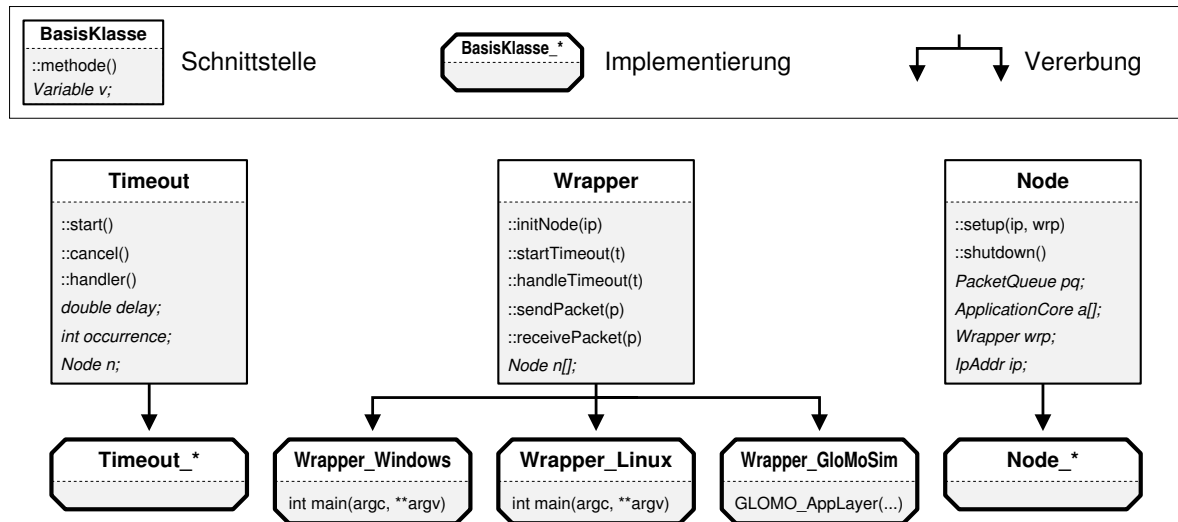


Abbildung 6.5 Kapselung System-naher Funktionen in der MAAM-Architektur.

Netzwerkkommunikationsschnittstelle als Parameter übergeben wird. Die Methode ist dafür zuständig, eine Implementierung der `Node`-Schnittstelle durch den Aufruf von `n[0]::setup(ip, wrp)` zu initialisieren.

- Die Methode `::startTimeout(t)` tätigt Systemaufrufe zum Starten eines Zeitgebers. Ein wichtiger Aspekt hierbei ist die Tatsache, dass in Abhängigkeit des Betriebssystems pro Prozess eventuell lediglich ein einziger Zeitgeber gestartet werden kann. Dies bedingt eine Zeitgeberverwaltung anhand einer entsprechenden Liste. Erfolgt nach dem Starten eines Zeitgebers t_1 mit $t_1::delay = \alpha$ etwa das Starten eines zweiten Zeitgebers t_2 mit $t_2::delay = \beta < \alpha$, so “verdrängt” der Zeitgeber t_2 den Zeitgeber t_1 . Nach der Behandlung von t_2 kann letztendlich ein neuer Zeitgeber mit der für t_1 verbleibenden Dauer $delay = \beta - \alpha$ gestartet werden.

- Die Methode `::handleTimeout(t)` entspricht einer Behandlungsroutine, welche durch das Betriebssystem bei Ablauf eines Zeitgebers aufgerufen wird. Die Behandlung des Zeitgebers erfolgt, indem die Behandlungsroutine `t::handle()` aufgerufen wird. Die zur korrekten Behandlung des Zeitgebers erforderliche Herstellung des Gerätekontexts erfolgt durch die in der `Timeout`-Implementierung vorhandene Strukturvariable `Node n`.

- Die Methode `::sendPacket(p)` versendet das übergebene Paket `p` über einen innerhalb des Wrapper verwalteten UDP-Socket an die in `p` kodierte Zieladresse.

- Die Methode `::receivePacket(p)` wird aufgerufen, sobald über die Systemmethode `pselect()` das Eintreffen eines Pakets auf einem der durch den Wrapper verwalteten Sockets festgestellt wird. Das Paket wird an die Paketwarteschlange des verwalteten Gerätekontexts durch Aufruf von `n[0]::pq::pktFromNet(p)` übergeben.

Mit Hilfe der Linux- und Windows-Wrapper konnten unterschiedliche Anwendungen auf Basis des von der MAAM-Architektur angebotenen P2P-basierten Gruppenkommunikationsdienstes nicht nur erfolgreich implementiert und betrieben, sondern auch auf renommierten Konferenzen vorgestellt werden [15, 17]. Beispiele der entwickelten Anwendungen sind in Abbildung 6.6 dargestellt. In dieser sind zu sehen:

- Eine *grafische Chat-Anwendung* zur Text-basierten Interaktion zwischen den Gruppenmitgliedern. Gruppenmitglieder können beim Starten der Anwendung einen so genannten “Nickname” angeben, anhand dessen sie im Laufe der Konversation identifiziert werden können. Zur Aufwertung der Anwendung werden die Gruppenmitglieder anhand eines ebenfalls beim Start der Anwendung aus einer Liste auswählbaren Avatars in einem virtuellen Chat-Raum dargestellt. Während es sich bei dem Chat-Raum um eine sich über mehrere Bildschirme erstreckende Europakarte handelt, können sich Gruppenmitglieder durch Verwendung der Cursor-Tasten frei bewegen. Durch Übertragung der Bewegungsinformation zu den übrigen Gruppenmitgliedern können diese die Bewegung anderer Benutzer verfolgen und ihnen innerhalb des Chat-Raums begegnen.

- Eine *Präsentationsanwendung* anhand welcher Vorlesungs- oder Seminarpräsentationen durchgeführt werden können. Die Anwendung überträgt den von einem Gruppenmitglied über einen Dateiauswahldialog bestimmten Foliensatz zu den übrigen Gruppenmitgliedern. Anhand entsprechender Pfeiltasten kann der Benutzer seine Präsentation abhalten, indem er durch Drücken der Pfeiltasten zwischen den einzelnen Folien hin- und herwechselt. In diesem Fall werden entsprechende Steuerungskommandos an die übrigen Gruppenmitglieder übertragen, wodurch auf deren Endgeräten ein Vor- bzw. Zurückspringen im Foliensatz ausgelöst wird. Die Präsentationsanwendung ermöglicht allen Gruppenmitgliedern das Abhalten individueller Präsentationen, wobei die jeweils ausgewählten Foliensätze durch die Anwendung parallel zwischen den Gruppenmitgliedern verteilt werden. Zur Steigerung der Übersicht, werden parallel abgehaltene Präsentationen von der Anwendung in unterschiedlichen *Reitern* untergebracht.

- Eine *Voice-over-IP-Anwendung* zur Sprach-gestützten Unterhaltung von Gruppenmitgliedern. Die Anwendung unterstützt die Mikrofon-basierte Spracheingabe sowie die Wiedergabe durch Rückgriff auf entsprechende Systembibliotheken. Zur Kompression der Sprachdaten kann beim Start der Anwendung über eine Dialogbox der zu verwendende CoDec ausgewählt werden: Verfügbare und in die Anwendung integrierte CoDecs sind Speex [109], GSM 06.10 [43] und iLBC [95]. Nach der Auswahl erscheint eine Liste der aktuell der Multicast-Gruppe beigetretenen Gruppenmitglieder sowie eine Schaltfläche zur Steuerung der Sprachübertragung: Wird diese Schaltfläche mit der Maus gedrückt, erfolgt für die Dauer des Mausdrucks die Abtastung und die Kompression der über das Mikrofon abgegriffenen Signale sowie das Versenden der komprimierten Daten zu den übrigen Gruppenmitgliedern. Dort werden die Daten wiederum dekom-

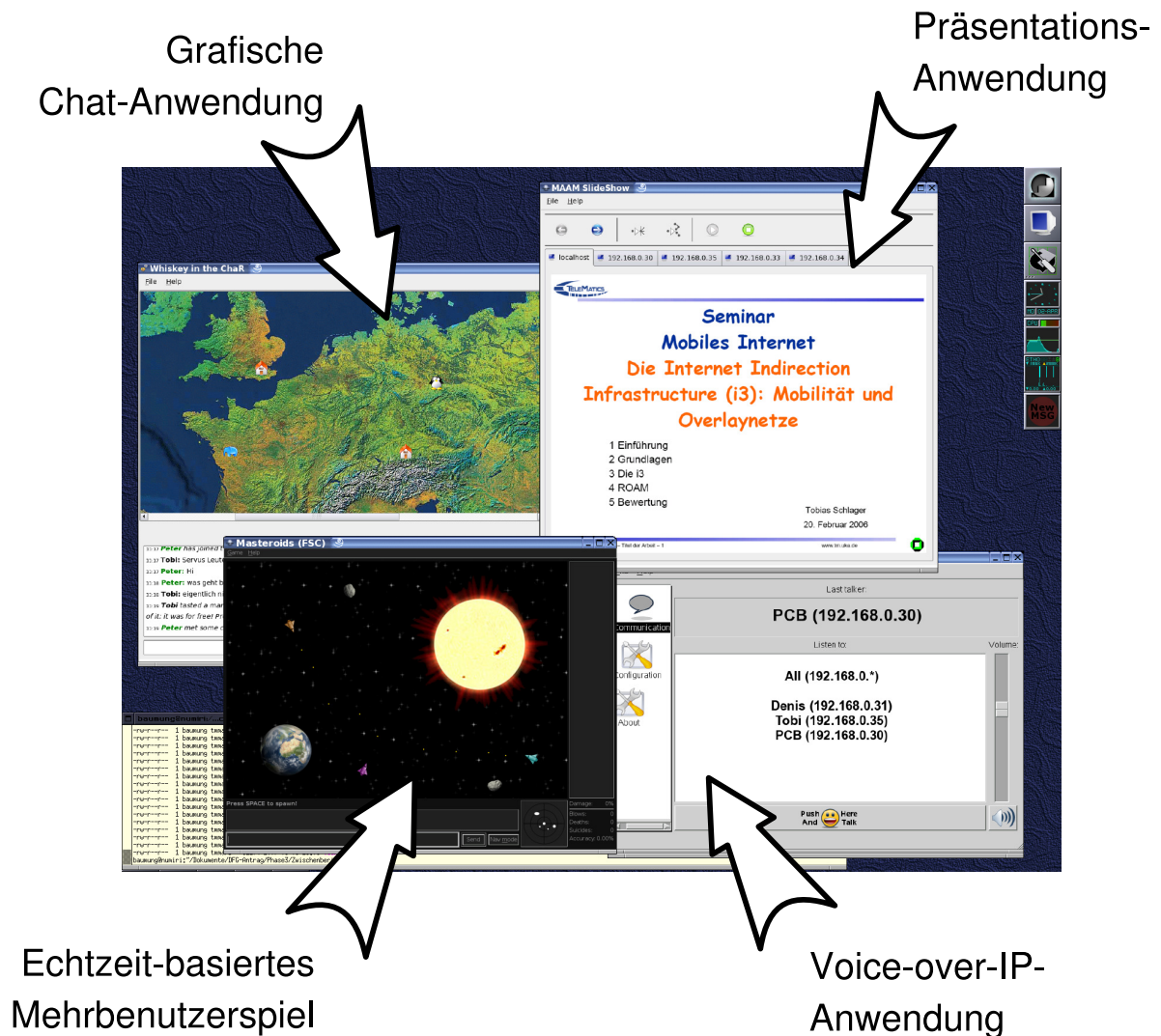


Abbildung 6.6 Linux- und Windows-Anwendungen, welche auf Basis des von der MAAM-Architektur angebotenen P2P-basierten Gruppenkommunikationsdienstes entwickelt wurden.

primiert und wiedergegeben. Durch Kennzeichnung des verwendeten CoDecs innerhalb der Pakete kann jedes Gruppenmitglied einen eigenen CoDec verwenden.

– Ein *Echtzeit-basiertes Mehrbenutzerspiel*, bei welchem die Gruppenmitglieder in einem virtuellen Sonnensystem anhand von Raumschiffen gegeneinander antreten. Aufgrund seiner Echtzeit-Anforderungen zeigt sich diese Anwendung als sehr fordernd gegenüber der zwischen Gruppenmitgliedern stattfindenden Kommunikation. Für diese Anwendung wurden daher die speziellen, in Anhang A.2 dokumentierten Zuverlässigkeitsmechanismen konzipiert und anhand entsprechender Module in die MAAM-Architektur integriert. Beim Starten der Anwendung können Spieler jeweils einen “Nickname” zu ihrer Identifikation eingeben und sich einem von 8 farblich unterschiedlich

gekennzeichneten Teams zuordnen. Anhand der Cursor-Tasten kann ein Raumschiff gedreht sowie in Blickrichtung beschleunigt oder abgebremst werden. Durch Betätigen der Space-Taste können wiederum Raketen abgefeuert werden, wobei drei Treffer zur Zerstörung eines gegnerischen Raumschiffs erforderlich sind. Jeder Spieler überträgt bei diesem Spiel seine eigene Position und Geschwindigkeit 25 mal pro Sekunde zu den übrigen Mitspielern. Zusätzliche Ereignisse, wie die Zerstörung eines Raumschiffes oder das Abfeuern einer Rakete, werden zwar in getrennten Paketen kodiert, durch die Zuverlässigkeitsmechanismen zur Einsparung von Medienzugriffen aber mit den periodisch ausgesendeten Positionsinformationen aggregiert.

6.5.1.2 Die Netzwerksimulationsumgebung GloMoSim

Zur simulativen Evaluation von P2P-basierten Gruppenkommunikationsdiensten wurde ein Wrapper zur Kapselung der Netzwerksimulationsumgebung GloMoSim [50, 118] entwickelt. Er implementiert die Schnittstelle `Wrapper` wie folgt:

- Die Methode `::initNode(ip)` wird von GloMoSim zu Beginn der Simulationszeit für jedes simulierte Endgerät aufgerufen, wobei die intern von GloMoSim für ein Endgerät verwendete `Node_ID` als `ip` übergeben wird. Der Aufruf veranlasst die Initialisierung einer `Node`-Implementierung durch den entsprechenden Aufruf von `n[ip]::setup(ip, wrp)`, wobei `wrp` der zur Kapselung von GloMoSim verwendete Wrapper ist.

- Die Methode `::startTimeout(t)` wird von Implementierungen der `Timeout`-Schnittstelle (vgl. Abschnitt 6.5.2) verwendet, um einen Zeitgeber zu starten. Bei Aufruf der Methode wird in die von GloMoSim vorgenommene Ereignisverwaltung ein zusätzliches Ereignis mit der durch `t::delay` spezifizierten Verzögerung eingebracht.

- Die Methode `::handleTimeout(t)` wird von GloMoSim nach Ablauf eines zuvor gestarteten Zeitgebers `t` aufgerufen. Die Behandlung des Zeitgebers erfolgt, indem der Wrapper die dazugehörige Behandlungsroutine `t::handle()` aufruft. Wichtig hierbei ist, dass Zeitgeber Endgerät-spezifisch sind: Die zur Behandlung des Zeitgebers erforderliche Herstellung des Gerätekontexts erfolgt durch die in der `Timeout`-Implementierung vorhandene Strukturvariable `Node n`.

- Die Methode `::sendPacket(p)` wird von Paketwarteschlangen zum Versenden eines Pakets `p` aufgerufen. Das Versenden erfolgt indem der Wrapper das Paket in GloMoSim als UDP-Paket an die in `p` kodierte Zieladresse sendet.

- Die Methode `::receivePacket(p)` wird von GloMoSim bei Eintreffen eines UDP-Pakets `p` aufgerufen. Der Wrapper extrahiert die Zieladresse `ip` des Pakets und übergibt dieses der Paketwarteschlange, welche in dem mit `ip` assoziierten Gerätekontext instanziiert ist. Dies erfolgt über den Aufruf von `n[ip]::pq::pktFromNet(p)`.

Der für GloMoSim entwickelte Wrapper wurde nicht nur zur Durchführung aller in dieser Arbeit präsentierten Evaluationen verwendet: Er wurde auch auf namhaften Konferen-

zen zur Demonstration von simulierten P2P-basierten Gruppenkommunikationsdiensten herangezogen [15, 17].

6.5.2 Die Timeout-Schnittstelle

Zur flexiblen Definition von Zeitgebern sieht die MAAM-Architektur eine einfache Schnittstelle `Timeout` vor. Die für die Umsetzung eines Zeitgebers wichtigste Methode ist die Methode `::handler()`: Diese wird von der MAAM-Architektur automatisch bei Ablauf des Zeitgebers aufgerufen und beinhaltet Funktionalität zu dessen Behandlung. Die übrigen Methoden `::start()` und `::cancel()` werden dazu verwendet, Zeitgeber zu starten und anzuhalten.

Die Schnittstelle bietet einige Variablen, welche den Umgang mit Zeitgebern vereinfachen und erst ermöglichen. Während die Variable `delay` die Dauer bis zum Ablauf des Zeitgebers bestimmt, zählt die Variable `occurrence` die Anzahl der Eintritte des Zeitgebers. Die Strukturvariable `Node n` bietet den für Zeitgeber erforderlichen Zugang zum Basissystem, welches anhand des System-spezifischen Wrapper `n::wrap` gekapselt ist: So kann ein Zeitgeber `t` durch einen Aufruf von `n::wrap::startTimeout(t)` letztendlich angesetzt werden.

6.5.3 Das ApplicationCore-Modul

Gruppenkommunikationsanwendungen, welche auf Basis der MAAM-Architektur entwickelt werden, sind ähnlich dem *Model-View-Control-Paradigma* (*MVC*) in zwei Funktionalitätskomponenten unterteilt. Während die eine Komponente die eigentliche Kernfunktionalität wie Datenverarbeitungsalgorithmen enthält, bietet die andere Komponente den Zugriff auf die Anwendungsfunktionalität anhand einer entsprechenden grafischen Benutzerschnittstelle. Während Umsetzungen von Benutzerschnittstellen in dieser Arbeit nicht weiter von Interesse sind, bietet die MAAM-Architektur die Möglichkeit, die Kernfunktionalität einer Anwendung in ein entsprechendes Modul, das `ApplicationCore`-Modul (zu sehen in Abbildung 6.2), zu integrieren.

Die Motivation hinter dieser Integration besteht darin, zumindest Teile einer Anwendungslogik derart zu kapseln, dass sie sowohl von einer real betriebenen Anwendung als auch innerhalb einer Netzwerksimulationsumgebung verwendet werden können. Interessant erscheint eine solche Kapselung beispielsweise bei der Evaluation von *Voice-over-IP*-Anwendungen (*VoIP*), bei welchen der Verlust von Datenpaketen eine wichtige Rolle spielt. Durch direkte Integration der zur Sprachkodierung eingesetzten `CoDecs` in die MAAM-Architektur, kann die von der *VoIP*-Anwendung erreichte Sprachqualität evaluiert werden, indem für Gruppenmitglieder empfangene Daten direkt dekodiert und gegebenenfalls wiedergegeben werden. Auf diese Weise erübrigen sich Sprachqualitätsprognosen auf Basis gemessener Paketverluste.

Zur möglichst geringen Restriktion der Integration von Anwendungsfunktionalität in `ApplicationCore`-Module, ist die Schnittstelle dieser Module sehr einfach gehalten:

- Die Methode `::start()` initialisiert eine Anwendung. In dieser Methode ist unter anderem die in Abschnitt 6.3 beschriebene Komposition des Gruppenkommunikationsdienstes vorzunehmen. Ebenso erfolgt in dieser Methode üblicherweise der Verbindungsaufbau zur Multicast-Gruppe durch Aufruf von `tb::setup()`.
- Die Methode `::quit()` beendet eine Anwendung. In dieser Methode erfolgt der Abbau der Verbindung zur Multicast-Gruppe durch einen Aufruf von `tb::shutdown()`.
- Die Methode `::sndData(d)` wird von `ApplicationCore`-Modulen nicht benötigt, da diese Datenpakete durch Aufruf von `tb::sndData(d)` versenden.
- Die Methode `::rcvData(d)` wird über `upc::rcvData(d)` von einem untergeordneten Modul aufgerufen, um der Anwendung empfangene Datenpakete zu übergeben. An den Empfang eines Datenpakets schließt sich seine Verarbeitung durch die innerhalb des `ApplicationCore`-Moduls integrierte Kernfunktionalität an.
- Die Methoden `::setupHeartbeat(hb)` und `::parseHeartbeat(hb)` können bei Verwendung des LBC-Konzepts zur Einbettung und Rückgewinnung von Information in periodisch versendete Heartbeats verwendet werden.

6.6 Zusammenfassung

In diesem Kapitel wurde die MAAM-Architektur vorgestellt. Sie wurde im Rahmen dieser Arbeit entwickelt, um die modulare Komposition P2P-basierter Gruppenkommunikationsdienste zu ermöglichen, welche heterogenen Anforderungen unterschiedlicher Anwendung gerecht werden [12]. Hierzu kapselt die MAAM-Architektur die beiden wichtigsten Protokollmechanismen, die Datenverteilung und die Umsetzung von Zuverlässigkeit, innerhalb von Modulen. Durch Definition fester Modulschnittstellen werden Module austauschbar und die Architektur flexibel gegenüber der Unterstützung unterschiedlicher Anwendungen. Darüber hinaus integriert die MAAM-Architektur das in Abschnitt 5.1.1 entworfene Konzept lokaler Broadcast-Cluster. Dabei erfolgt die Integration derart, dass der Einsatz des LBC-Konzepts nicht nur transparent aus Sicht von Zuverlässigkeitsmechanismen und Overlay-Topologien sondern auch optional aus Sicht von Anwendungen bleibt. Durch eine zusätzliche Schnittstelle kapselt die MAAM-Architektur den Zugriff auf so genannte Basissysteme, welche eine Zeitgeberverwaltung und den Zugriff auf eine Netzwerkkommunikationsschnittstelle bieten. Aufgrund dieser Kapselung erlangt die MAAM-Architektur einen hoch-portablen Charakter: So kann sie (und die darin entwickelten Module) nicht nur zur simulativen Evaluation von Protokollen in Netzwerksimulationsumgebungen betrieben sondern auch zur Erstellung funktionsfähiger Anwendungen auf realen Endgeräten eingesetzt werden.

7. Protokollauswertung

Um einen Einblick in das Verhalten der in dieser Arbeit diskutierten Protokolle und Overlay-Topologien zu geben und um Aussagen über sowie Vergleiche von deren Leistungsfähigkeit machen zu können, wird in diesem Kapitel eine ausführliche Evaluation anhand einer Netzwerksimulationsumgebung durchgeführt. Als Simulationsumgebung wird die *Global Mobile Information Systems Simulation Library* (kurz *GloMoSim* [50]) verwendet, da sie sowohl über zahlreiche Protokollimplementierungen für die Wegewahl und den Medienzugriff in drahtlosen Ad-hoc-Netzen verfügt als auch ein von der Forschungsgemeinschaft weitläufig verwendetes Ausbreitungs- und Interferenzmodell elektromagnetischer Funkssignale (*Two-ray-ground with Accumulative Noise*) umsetzt. Obgleich von GloMoSim inzwischen eine weiterentwickelte und kommerziell vertriebene Version (*QualNet* [90]) existiert, liegen bei dieser Protokoll- und Modellimplementierungen lediglich in kompilierter Form vor, so dass genaue Abläufe nicht nachvollzogen und etwaige Erweiterungen und Modifikationen nur schwer integriert werden können.

Zu Beginn des Kapitels erfolgt eine Darstellung der verwendeten Konfiguration der Simulationsumgebung sowie der untersuchten Szenarien im Sinne der Modellierung drahtloser Ad-hoc-Netze und der betrachteten Gruppenkommunikationsanwendungen. Vor der Evaluation der eingeführten Anwendungen erfolgt jedoch zunächst eine Untersuchung der Overlay-Topologien an sich, um einen Einblick in den mit ihrem Aufbau und ihrer Verwaltung verbundenen Aufwand zu erhalten.

7.1 Konfiguration der Simulationsumgebung

In diesem Abschnitt erfolgt zunächst eine Darstellung der Konfiguration der in dieser Arbeit evaluierten Szenarien. Hierzu zählt nicht nur die Konfiguration und die Para-

meter der verwendeten Simulationsumgebung, sondern auch die betrachteten Modellierungen von drahtlosen Ad-hoc-Netzen sowie die dazugehörige Verteilung der Endgeräte innerhalb des Simulationsareals. Darüber hinaus werden die in dieser Arbeit evaluierten Anwendungen beschrieben.

7.1.1 Allgemeine Simulationsparameter

Die für die Simulationen verwendeten Einstellungen und Parameter der einzelnen Protokollschichten sind in Tabelle 7.1 aufgelistet. Sie werden durchgehend für alle Simulationen verwendet - sofern von den Einstellungen abgewichen wird, wird dies ausdrücklich erwähnt. Aus den in der Tabelle angegebenen Werten ergibt sich gemäß dem gewählten Ausbreitungsmodell ein Übertragungsbereich von $175m$ sowie ein Erkennungsbereich von $353,8m$. Obgleich die RTS/CTS-Erweiterung des Medienzugriffsprotokolls aufgrund zahlreicher Untersuchungen (wie etwa [106]) standardmäßig deaktiviert wird, wird sie in einigen der untersuchten Szenarien nach expliziter Erwähnung zum Einsatz kommen. Die verwendete Kapazität der IP-Paketwarteschlange entspricht der Standardeinstellung aktueller Linux-Betriebssysteme.

7.1.2 Betrachtete Szenarien

In diesem Abschnitt werden die in dieser Arbeit betrachteten Simulationsszenarien beschrieben. Dabei werden zunächst die verwendeten Mobilitätsmodelle dargestellt. Darauf aufbauend werden anschließend die tatsächlich erzeugten Bewegungsszenarien diskutiert und die simulierten Anwendungen sowie deren Ablauf betrachtet.

7.1.2.1 Mobilitätsmodellierung

Zur Modellierung von Benutzerbewegungen in drahtlosen Ad-hoc-Netzen wurden in den vergangenen Jahren zahlreiche Ansätze vorgeschlagen [29]. Das von der Forschungsgemeinschaft für viele Jahre als Standardmodell eingesetzte *Random Waypoint Mobility Model* lässt Endgeräte auf Geraden wandern, welche durch die Verbindung zufällig gewählter Punkte innerhalb des Simulationsareals entstehen. Dabei wird die Geschwindigkeit nach oben und unten beschränkt. Beim Erreichen eines Geradenendpunkts wird eine Pausezeit zufällig gewählt, während welcher das Endgerät bewegungslos an dem Punkt verharrt. Nach Ablauf dieser Zeitspanne erfolgt eine neue, zufällige Wahl eines Geradenendpunkts, zu welchem sich das Endgerät anschließend bewegt. Verschiedene Untersuchungen zeigen, dass bei dem Random Waypoint Modell die initiale Gleichverteilung der Endgeräte innerhalb des Simulationsareals nicht erhalten bleibt, sondern über die Simulationszeit hinweg eine Häufung der Endgeräte im Arealszentrum beobachtet werden kann [21, 58]. Da Änderungen der Endgerätdichte unterschiedliche Auswirkungen auf die Leistungsfähigkeit von Protokollen haben [72], wird seit einigen Jahren von

Parameter	Wert
Ausbreitungsmodell	Two-ray-Ground
Antennenhöhe	1,5 m
Interferenzmodell	Accumulative Noise
Medienzugriffsprotokoll	IEEE 802.11b
Bandbreite	2 MBit/s
Wellenlänge	2,4 GHz
RTS/CTS-Erweiterung	Nein
Sendeleistung	3,9068 dBm
Rx-Schwellenwert	-81 dBm
CS-Schwellenwert	-91 dBm
Signal-to-Noise-Ratio (SNR)	10 dBm
Routing-Protokoll	AODV
Local-Repair-Erweiterung	Ja
Kapazität der IP-Paketwarteschlange	100 Pakete

Tabelle 7.1 Konfigurationsparameter der Simulationsumgebung GloMoSim.

diesem Modell zunehmend abgesehen. Komplexere Modelle erzeugen Hindernisse (wie Bäume oder Gebäude) auf dem Simulationsareal und lassen Endgeräte entsprechend ausweichen und verändern das Ausbreitungsverhalten der ausgesendeten Funksignale [59]. Insbesondere der Effekt unterschiedlicher Ausbreitungsmodelle für Funksignale ist jedoch ein erst in den letzten Jahren aufgekommener Forschungsbereich [107], so dass in dieser Arbeit auf etablierte, wenn auch ältere, Modelle zurückgegriffen wird.

In dieser Arbeit wird zur Mobilitätsmodellierung das *Random Direction Mobility Model* eingesetzt. Es basiert auf einem Prinzip, bei welchem sich Endgeräte für eine bestimmte Bewegungsdauer in eine zufällig gewählte Richtung bewegen, wobei die Bewegungsgeschwindigkeit aus einem festgelegten Intervall gewählt wird. Stößt das Endgerät im Laufe seiner Bewegung an die Grenzen des Simulationsareals, wird der Richtungsvektor der Bewegung an der Begrenzung reflektiert. Bei Ablauf der Bewegungsdauer legt das Endgerät eine Pause mit einer zufällig aus einem Intervall gewählten Länge ein und wählt anschließend zufällig eine neue Bewegungsrichtung und -dauer. Obgleich das Prinzip sehr ähnlich dem des Random Waypoint Modells ist, behält das Random Direction Modell jedoch eine Gleichverteilung der Endgeräte über das Simulationsareal bei.

Darüber hinaus wird in dieser Arbeit das *Group Mobility Model* [53] eingesetzt, bei welchem einzelne Endgeräte zu kleinen Verbänden zusammengefasst werden. Verbände haben dabei eine bestimmte Ausdehnung, so dass die maximale Entfernung von Endgeräten innerhalb des gleichen Verbands begrenzt ist. Analog zu dem genannten Random Direction Mobility Modell bewegen sich Verbände von Endgeräten mit einer gewissen Geschwindigkeit und Richtung für eine bestimmte Dauer.

7.1.2.2 Bewegungsszenarien

In dieser Arbeit wird davon ausgegangen, dass die evaluierte Gruppenkommunikation und die daran beteiligten Endgeräte einem existierenden drahtlosen Ad-hoc-Netz *überlagert* ist. Als Konsequenz dessen setzt sich das simulierte Ad-hoc-Netz aus 1. Endgeräten innerhalb eines so genannten *Basisnetzes* und 2. den (zusätzlichen) Gruppenmitgliedern der Gruppenkommunikation zusammen. Alle Endgeräte bewegen sich über die Simulationsdauer hinweg auf einem quadratischen Simulationsareal von $1000 \times 1000 m^2$.

Um für genügend Konnektivität innerhalb des drahtlosen Ad-hoc-Netzes zu sorgen, wird das Basisnetz mit einer gleich bleibenden Anzahl von 100 Endgeräten modelliert, welche sich gemäß dem Random Direction Mobility Modell bewegen. Die Geschwindigkeiten werden gleichverteilt aus dem Intervall $[1, 25 m/s, 1, 75 m/s) \hat{=} [4, 5 km/h, 6, 3 km/h)$ gewählt, welches den von Fußgängern erreichbaren Geschwindigkeiten entspricht. Endgeräte des Basisnetzes behalten ihre Bewegungsrichtung für eine Dauer bei, welche gleichverteilt aus dem Intervall $[30s, 180s)$ gewählt wird. Nach Ablauf einer gewählten Dauer wird eine Bewegungspause von maximal $5s$ eingelegt. Endgeräte aus dem Basisnetz beteiligen sich selbst nicht an der Gruppenkommunikation und dienen lediglich der Weiterleitung von Dateneinheiten.

Dem Basisnetz werden zusätzliche Endgeräte hinzugefügt, welche den an der Gruppenkommunikation teilnehmenden Gruppenmitgliedern entsprechen. In Abhängigkeit der betrachteten Szenarien und Gruppengröße, variiert die Anzahl der hinzugefügten Endgeräte zwischen 5 und 50. In dieser Arbeit wird davon ausgegangen, dass sich die Gruppenmitglieder zur Kommunikation zu kleinen Verbänden zusammenfinden. Diese werden gemäß dem Group Mobility Modell erzeugt, wobei ein Verband bis zu 4 Gruppenmitgliedern aufnehmen kann und einen Durchmesser von maximal $15m$ besitzt. Da die Gruppenkommunikation ein gewisses Maß an Aufmerksamkeit fordert, wird die Bewegungsgeschwindigkeit für Gruppenmitglieder reduziert und gleichverteilt aus dem Intervall $[1, 0 m/s, 1, 5 m/s) \hat{=} [3, 6 km/h, 5, 4 km/h)$ gewählt. Die Pausezeit wird für Gruppenmitglieder auf bis maximal $60s$ erhöht.

In dieser Arbeit werden hauptsächlich Aspekte der Skalierbarkeit von Gruppenkommunikationsprotokollen im Hinblick auf die Gruppengröße und das Datenverkehrsaufkommen betrachtet. Die durchgeführten Messungen sollen daher lediglich von diesen Variablen beeinflusst werden und nicht etwa durch unvorhersagbares Verhalten, wie es im Zusammenhang mit Netzpartitionierungen auftreten kann, verfälscht werden. Daher kommen in dieser Arbeit lediglich Bewegungsszenarien zum Einsatz, für welche eine Partitionierung der Gruppenmitglieder (sofern sie auftritt) für Zeitintervalle von höchstens $2s$ Länge auftritt. Für eine solche Maximaldauer kann der Effekt von Partitionierungen vernachlässigt werden, da sie lediglich einzelne Pakete betreffen und so von den in drahtlosen Ad-hoc-Netzen üblichen Paketverlusten nicht unterschieden werden können.

Bemerkung: Die oben beschriebenen Bewegungsszenarien finden in allen simulierten Szenarien Verwendung. Sofern davon abgewichen wird, wird dies in den entsprechenden Evaluationsabschnitten ausdrücklich erwähnt.

7.1.2.3 Anwendungsszenarien

Ein wichtiger Aspekt, welcher bei den in dieser Arbeit durchgeführten Evaluationen berücksichtigt wird, ist der Betrieb unterschiedlicher Anwendungen über einer Overlay-Topologie. Da in Abhängigkeit des von einer Anwendung ausgesendeten Datenverkehrs mitunter deutlich unterschiedliche Ergebnisse zu erwarten sind, werden die in dieser Arbeit untersuchten Overlay-Topologien im Kontext von drei unterschiedlichen Anwendungen evaluiert:

- *Einer Constant-Bitrate-Anwendung (CBR):* Bei einer solchen Anwendung geht ein Datenstrom von einer einzigen Quelle aus. Die Quelle sendet in einem gleich bleibenden zeitlichen Abstand Datenpakete (annähernd) gleicher Größe aus, so dass sie über die Zeit hinweg eine gleich bleibende Datenrate erzielt. Derartige Datenströme werden beispielsweise im Zusammenhang mit *Voice-over-IP*- oder manchen *Video-Streaming-CoDecs* beobachtet, wobei der zeitliche Abstand der einzelnen Datenpakete sowie deren Größe von dem jeweiligen CoDec abhängt.

- *Einem Mehrbenutzer-Chat:* Entgegen den inzwischen sehr populär gewordenen Instant-Messaging-Anwendungen (ICQ, Yahoo, AIM, Jabber), bei welchen ein Benutzer üblicherweise mit anderen, einzelnen Benutzern getrennt kommuniziert, bieten Chat-Räume (wie etwa das IRC [71]) die Möglichkeit, mit ganzen Gesprächspartnergruppen gleichzeitig auf textueller Basis zu kommunizieren. Im Vergleich zu einer oben genannten CBR-Anwendung tritt bei einer Chat-Anwendung Datenverkehr eines bestimmten Benutzers nur sehr sporadisch auf, abhängig von der Intensität der geführten Unterhaltung. Dahingegen nehmen unter Umständen jedoch mehrere Dutzend Benutzer gleichzeitig an einer Unterhaltung teil, so dass mit mehreren Dutzend Datenquellen zu rechnen ist.

- *Einem Echtzeit-Mehrbenutzer-Spiel:* Bei diesen Anwendungen sendet jeder teilnehmende Spieler bis zu 25 mal pro Sekunde seine aktuelle Position und seine Bewegungsgeschwindigkeit in einem virtuellen Raum an die übrigen Mitspieler. Während sich die Daten in wenigen (≈ 20 bis 30) Bytes kodieren lassen, ist eine zügige Auslieferung der Daten an die Mitspieler entscheidend, da veraltete Positionsinformationen die “gefühlte und erlebte Spielqualität” in Mitleidenschaft ziehen.

Um an der Gruppenkommunikation teilnehmen zu können, treten die Gruppenmitglieder der Multicast-Gruppe in den ersten 5 Minuten Simulationszeit bei. Hierbei erfolgt die Eingliederung der Gruppenmitglieder in die Overlay-Topologien, wobei für die getätigten Untersuchungen jeweils die Overlay-Protokolle TrAM (mit $\alpha = 0,25$ und $t_{refresh} = 10s$), Narada, NICE, PAST-DM sowie das in Anhang A.1 erläuterte

n*Unicast-Protokoll zum Einsatz kommen. Je nach Untersuchung werden die Overlay-Protokolle gegebenenfalls durch das LBC-Konzept erweitert, wobei die Bewertung von LBC-Repräsentanten durch lokale Gruppenmitglieder aufgrund der in Anhang A.3 beschriebenen Funktionen vorgenommen werden. In Abhängigkeit der jeweiligen Anwendungsszenarien schließt sich an die Gruppenbeitrittsphase eine variable Simulationsdauer an, während welcher die eigentlichen Nutzdatenübertragungen erfolgen. Sofern dies in einzelnen Abschnitten nicht explizit erwähnt wird, erfolgt die Gruppenkommunikation über ein unzuverlässiges Transportprotokoll, so dass Gruppenmitglieder keine Versuche unternehmen, Paketverluste durch Übertragungswiederholungsanforderungen zu beheben.

***Bemerkung:** Im Zusammenhang mit zeitlichen Aussagen über Simulationen werden Begriffe wie “Sekunde” und “Simulationssekunde” als Synonyme verwendet. Aussagen werden niemals über die zur Simulation benötigte Rechnerzeit sondern immer über die Simulationszeit getroffen.*

7.2 Definitionen gemessener Werte

In diesem Abschnitt erfolgt eine Definition der Größen, welche im Laufe der Evaluation gemessen werden, um den mit Overlay-Topologien einhergehenden Aufwand sowie die Leistungsfähigkeit der über Overlay-Topologien betriebenen Anwendungen vergleichen zu können. Die gemessenen Werte umfassen:

Kontrollfluss [*KByte*]: Der Kontrollfluss entspricht der Summe der Größe aller Pakete, welche auf Anwendungsschicht der Gruppenmitglieder zum Aufbau und zur Pflege der Overlay-Topologie gesendet werden. Der Kontrollfluss umfasst demnach weder die Datenpakete der Anwendung noch zusätzliche Paketköpfe, welche auf unteren Schichten angehängt werden. Ebenso wenig gibt er Auskunft über sonstigen Aufwand, der auf unteren Schichten eventuell betrieben werden muss, um eine Kommunikation zwischen Endgeräten zu ermöglichen.

Medienzugriffszeit [*s*]: Die Medienzugriffszeit ist definiert, als die Summe der Dauern aller Medienzugriffe, welche von allen Endgeräten im gesamten drahtlosen Ad-hoc-Netz durchgeführt werden. Sie umfasst demnach jegliche Kommunikation, welche zwischen beliebigen Endgeräten stattfindet.

Verworfen Pakete [*Anzahl*]: Die Anzahl der verworfenen Pakete wird über *alle* Endgeräte innerhalb des drahtlosen Ad-hoc-Netzes bestimmt. Sie umfasst somit nicht nur Messungen innerhalb der Gruppenmitglieder, sondern auch der an der Gruppenkommunikation unbeteiligten, aber eventuell zur Weiterleitung von Unicast-Paketen herangezogenen Endgeräte. Die Anzahl verworfener Pakete gibt nicht nur Auskunft über die Be- und die eventuelle Überlastung des drahtlosen Ad-hoc-Netzes: Stattdessen gibt sie

ebenfalls Auskunft über Pakete, welche wegen (etwa aufgrund von Mobilität) gebrochener Übertragungsabschnitte verworfen werden. Um genauere Aussagen über verworfene Pakete machen zu können, sind die Verwurfsursachen aufgeschlüsselt in

- *Unerreichbarkeit (MAC)*: Derartige Pakete werden von dem Medienzugriffsprotokoll in Folge einer anhaltend fehlgeschlagenen Unicast-Übertragung verworfen. Die genaue Ursache hierfür (“anhaltende Interferenzen aufgrund einer hohen Belastung des drahtlosen Ad-hoc-Netzes” oder “aufgrund von Mobilität gebrochener Link”) kann jedoch nicht festgestellt werden.
- *Warteschlangenleerung (IP)*: Hierdurch werden diejenigen Pakete erfasst, welche aus der Paketwarteschlange auf Vermittlungsschicht verworfen werden, da das Endgerät, an welche die Pakete weiterzuleiten sind, soeben von dem Medienzugriffsprotokoll als unerreichbar gemeldet wurde.
- *Warteschlangenüberlauf (IP)*: Diese Pakete werden verworfen, da die Kapazität der auf Vermittlungsschicht angesiedelten Paketwarteschlange erschöpft ist und keine weiteren Pakete aufnehmen kann.

Strukturänderungen [*Anzahl*]: Die Strukturänderungen bezeichnen die Anzahl der topologischen Änderungen innerhalb einer Overlay-Topologie. Eine Strukturänderung entspricht dabei einer auf- oder abgebauten Overlay-Verbindung.

Aufgebaute Routen [*Anzahl*]: Diese Größe gibt die Anzahl der Routen wieder, welche von dem reaktiven Routing-Protokoll AODV aufgrund der Kommunikation zwischen Endgeräten aufgebaut wurden.

Zustellrate [%]: Die Zustellrate ergibt sich bei der Anwendungsevaluation. Sie gibt an wie viel Prozent der von einer Multicast-Quelle ausgesendeten Datenpakete die einzelnen Empfänger durchschnittlich empfangen.

Latenz [*s*]: Analog zur Zustellrate ergibt sich auch die Latenz bei der Anwendungsevaluation. Sie gibt an mit welcher Verzögerung die von einer Multicast-Quelle ausgesendeten Datenpakete bei den einzelnen Empfängern durchschnittlich eintreffen.

Alle in dieser Arbeit gemessenen Größen und Werte wurden durch Mittelung der Ergebnisse von insgesamt 100 Bewegungsszenarien erhalten. Um die Signifikanz der Mittelwerte beurteilen zu können, sind in ausgewählten Diagrammen die zugehörigen 95%-Konfidenzintervalle eingezeichnet.

7.3 Auswertung der Overlay-Topologien

Zu Beginn der Evaluation wird zunächst das Verhalten der einzelnen Overlay-Topologien sowie der mit ihrem Aufbau verbundene Aufwand untersucht. Abschnitt 7.3.1

widmet sich hierfür den Overlay-Topologien selbst, während Abschnitt 7.3.2 die Auswirkung einer Kombination mit dem in Abschnitt 5.1.1 dargestellten Konzept lokaler Broadcast-Cluster untersucht. Bemerkt werden soll an dieser Stelle noch einmal, dass über den Overlay-Topologien keine tatsächliche Anwendung betrieben und somit auch kein Datenverkehr über die Overlay-Topologien geleitet wird. Alle Simulationen in diesem Abschnitt erstrecken sich über 30 Simulationsminuten, von welchen die ersten 5 auf die Gruppenbeitrittsphase entfallen.

7.3.1 Standard P2P-Multicast

Dieser Abschnitt untersucht den Aufwand, welcher mit der Verwaltung von Overlay-Topologien einhergeht. Darüber hinaus werden Aussagen über das Topologieverhalten getroffen.

7.3.1.1 Kontrollfluss und Medienzugriffszeit

Der von den einzelnen Overlay-Topologien verursachte Kontrollfluss sowie die von dem Kontrollfluss induzierte Medienzugriffszeit ist in Abbildung 7.1 für unterschiedliche Gruppengrößen aufgezeigt, wobei die pro Sekunde gemessenen Werte über die Simulationszeit hinweg aufgetragen sind.

In allen Diagrammen ist zunächst ein Anstieg der dargestellten Kurven in den ersten 5 Minuten Simulationszeit zu erkennen. Hierbei handelt es sich um die Gruppenbeitrittsphase, während welcher alle Gruppenmitglieder der Multicast-Gruppe beitreten und somit ihre Eingliederung in die Overlay-Topologie anstreben. Im Anschluss an die Beitrittsphase ist für nahezu alle Kurven ein annähernd konstanter Verlauf für die verbleibende Simulationszeit zu beobachten: Während dieser Phase erfolgt die kontinuierliche Anpassung des Overlay an die Topologie des drahtlosen Ad-hoc-Netzes. Auffällig ist hier, dass die durchgeführten Topologieanpassungen einen sehr gleich bleibenden Kontrollfluss ohne deutliche Schwankungen verursachen.

Bei Betrachtung des ausgesendeten Kontrollflusses (linke Diagrammspalte) fällt auf, dass sich die Overlay-Topologien hier teilweise deutlich unterscheiden. So zeigen insbesondere diejenigen Overlay-Topologien, welche auf dem expliziten Austausch von Routing-Information basieren (Narada und PAST-DM), einen weitaus höheren Kontrollfluss auf als Overlay-Topologien, welche die Wegewahl für Datenpakete direkt aus ihrer Topologie ableiten können (TrAM und NICE). Dabei fällt auf, dass sich der Unterschied mit steigender Gruppenmitgliedszahl (Vergleich der Diagramme von oben nach unten) mitunter drastisch vergrößert: Vor allem bei der Overlay-Topologie Narada, bei welcher Routing-Pfade für alle Gruppenmitglieder ausgetauscht werden, ist ein quadratischer Anstieg des Kontrollflusses ansatzweise erkennbar - dieser wird in den folgenden Diskussionen noch deutlicher zum Vorschein kommen.

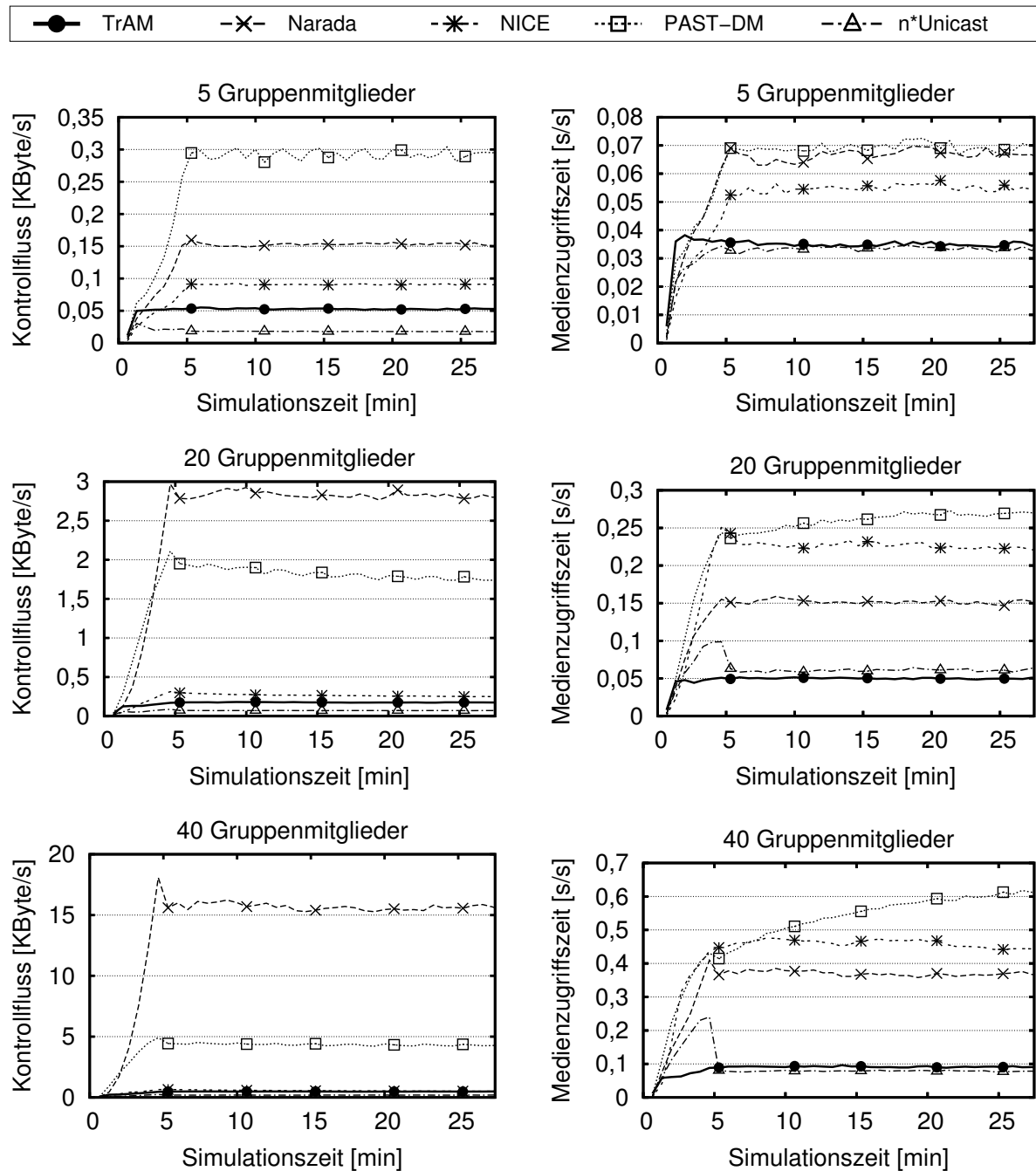


Abbildung 7.1 Verlauf des pro Simulationssekunde gemessenen Kontrollflusses und der Medienzugriffszeit über die Simulationszeit hinweg.

Der für die Pflege von Overlay-Topologien ausgesendete Kontrollfluss hat teilweise deutlich unterschiedliche Auswirkungen auf die tatsächliche Belastung des drahtlosen Mediums. Diese Belastung ist in der rechten Diagrammspalte als Medienzugriffszeit aufgetragen. Während sich hier die einzelnen Overlay-Topologien wiederum teilweise deutlich unterscheiden, sind die letztendlichen Unterschiede jedoch bedeutend geringer als bei

Betrachtung des Kontrollflusses. So zeigt PAST-DM bei 5 Gruppenmitgliedern einen Kontrollfluss auf, welcher um den Faktor 6 höher ist als bei TrAM. Die letztendlich erfolgende Medienzugriffszeit unterscheidet sich jedoch lediglich durch einen Faktor 2. Ebenso kann festgestellt werden, dass die Zusammensetzung des Kontrollflusses (Anzahl und Größe einzelner Kontrollpakete) einen ausschlaggebenden Einfluss auf die Medienzugriffszeit hat: So verursacht bei 20 Gruppenmitgliedern die NICE-Topologie, welche innerhalb ihrer vollvermaschten Cluster-Struktur sehr kleine Kontrollpakete an viele unterschiedliche Gruppenmitglieder verschickt, eine *um den Faktor 1,5 höhere* Medienzugriffszeit als die Overlay-Topologie von Narada, bei welcher die aufgrund der eingebetteten Routing-Information sehr großen Kontrollpakete jedoch nur mit wenigen Gruppenmitgliedern ausgetauscht werden. Dieses Verhalten ist umso interessanter, da bei alleiniger Betrachtung des Kontrollflusses das Gegenteil vermutet werden konnte: So weißt Narada bei 20 Gruppenmitgliedern einen Kontrollfluss auf, dessen Volumen dasjenige von NICE um nahezu *eine Größenordnung übersteigt*. Für PAST-DM ist ab 20 und insbesondere bei 40 Gruppenmitgliedern auch nach Abschluss der eigentlichen Gruppenbeitrittsphase ein weiteres Ansteigen der verursachten Medienzugriffszeit zu erkennen. Weil dies trotz einem konstant bleibenden oder sogar leicht abfallenden Kontrollfluss erfolgt, sind Effekte auf unteren Schichten, wie beispielsweise Paketkollisionen und durch das Medienzugriffsprotokoll erfolgende Übertragungswiederholungen, als Auslöser für diesen Anstieg zu sehen. Diese Aussage kann durch die ergänzende Abbildung 7.2 unterstützt werden. In dieser sind die im gesamten drahtlosen Ad-hoc-Netz pro Simulationssekunde durch das Medienzugriffsprotokoll durchgeführten Übertragungswiederholungen bei 40 Gruppenmitgliedern dargestellt: Hier ist zu erkennen, dass für PAST-DM Übertragungswiederholungen (und somit Kollisionen) zunehmend stattfinden, und diese Effekte sich über die Simulationszeit verstärken.

Das in dieser Arbeit entworfene Protokoll TrAM fällt in den dargestellten Diagrammen mit einem sehr geringen Aufwand auf, sowohl im Sinne des Kontrollflusses als auch im Sinne der Medienzugriffszeit. Lediglich die Mitgliederverwaltung des n*Unicast-Protokolls kann diesen Aufwand teilweise untertreffen: Während hier im Laufe der Gruppenbeitrittsphase ein häufigerer Austausch aktualisierter Gruppenmitgliedslisten stattfindet, sinkt die Medienbelastung im Anschluss an die Beitrittsphase deutlich ab.

Bemerkung: *Aufgrund der gezeigten Messungen kann an dieser Stelle bereits gezeigt werden, dass selbst bei der Entwicklung von Protokollen auf höheren Schichten (wie etwa der Anwendungsschicht) eine umfassende Betrachtung der tatsächlich ablaufenden Kommunikation erforderlich ist, um das Verhalten und die Auswirkungen eines Protokolls bewerten zu können.*

In Abbildung 7.3 sind Betrachtungen des durchschnittlichen Kontrollflusses und der durchschnittlichen Medienzugriffszeit pro Simulationssekunde dargestellt, wobei die Mes-

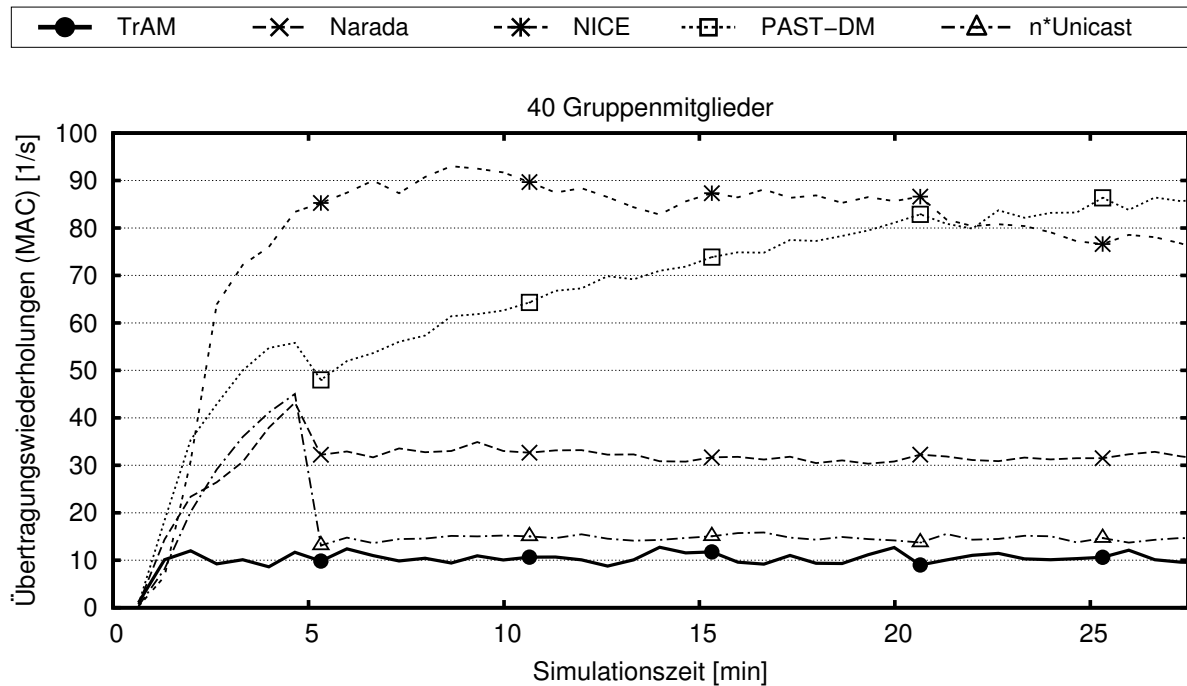


Abbildung 7.2 Vom Medienzugriffsprotokoll pro Simulationssekunde durchgeführte Übertragungswiederholungen bei 40 Gruppenmitgliedern.

sungen über die Gruppenmitgliedszahl aufgetragen sind. Bei Betrachtung des Kontrollflusses (linkes Diagramm) fällt erneut der deutliche Unterschied zwischen Overlay-Topologien auf Basis impliziter (TrAM und NICE) und expliziter (Narada und PAST-DM) Routing-Information auf: Insbesondere ist hier der quadratische Anstieg von Naradas Kontrollfluss in Abhängigkeit der Gruppenmitgliedszahl sehr gut erkennbar. Dennoch ist bei Betrachtung der Medienzugriffszeit (rechtes Diagramm) zu bemerken, dass Narada, trotz einem ebenfalls überlinearen Anstieg, für die betrachteten Gruppenmitgliedszahlen das Medium weniger belastet als beispielsweise PAST-DM und NICE. Sowohl TrAM als auch das n*Unicast-Protokoll heben sich von den übrigen Protokollen durch eine weitaus geringere Medienzugriffszeit ab. Die in die Diagramme jeweils eingezeichneten Konfidenzintervalle zeigen sich als äußerst klein, was für eine hohe Signifikanz der erhaltenen Mittelwerte spricht.

Um einen Anhaltspunkt über die innerhalb des drahtlosen Ad-hoc-Netztes verworfenen Pakete zu bekommen, sind diese in Abbildung 7.4 aufgetragen. Die Darstellung erfolgt dabei für unterschiedliche Gruppenmitgliedszahlen sowie für die einzelnen Overlay-Topologien. Obwohl die Ursachen der Verwürfe wie in Abschnitt 7.2 beschrieben aufgeschlüsselt sind, sind an dieser Stelle keine Paketverwürfe auf Vermittlungsschicht erkennbar, weder aufgrund von Wartenschlangenüberläufen noch aufgrund von Warteschlangenleerungen. Dies kann auf die Tatsache zurückgeführt werden, dass im Rah-

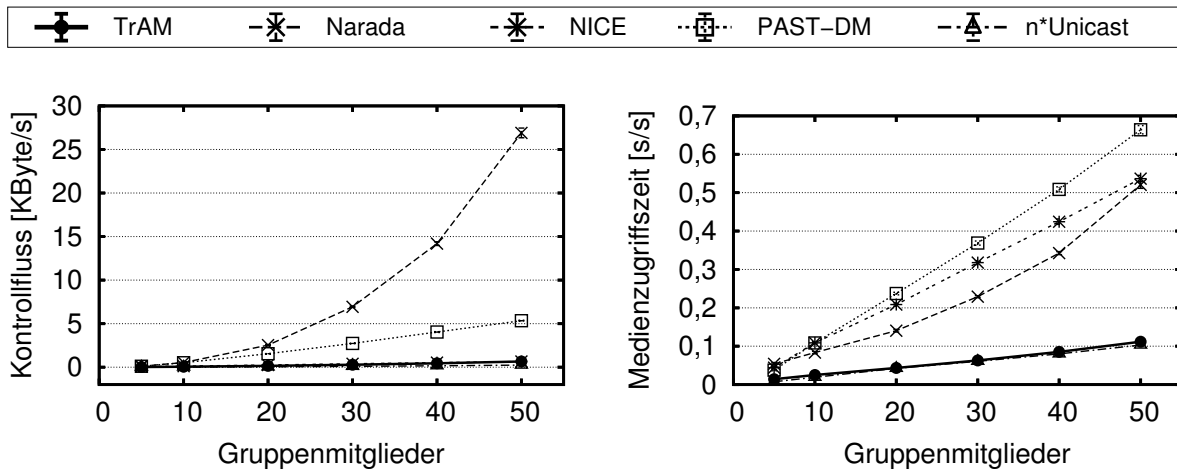


Abbildung 7.3 Pro Simulationsekunde anfallender Kontrollfluss und erfolgreiche Medienzugriffszeit.

men der alleinigen Verwaltung der Overlay-Topologien bei den betrachteten Gruppenmitgliedszahlen einerseits vergleichsweise wenig Pakete verschickt werden und andererseits noch keine Überlastung des drahtlosen Ad-hoc-Netzes erreicht wird. Aus diesem Grund werden Pakete hauptsächlich aufgrund der Mobilität von Endgeräten und somit aufgrund von Unerreichbarkeiten durch das Medienzugriffsprotokoll verworfen. Wird daraufhin die Unerreichbarkeit des Endgeräts dem Routing-Protokoll gemeldet, ist die Wahrscheinlichkeit, dass sich innerhalb der Paketwarteschlange noch weitere Pakete, die über das gleiche Endgerät weiterzuleiten sind, aufhalten äußerst gering. Dennoch ist zu bemerken, dass die Anzahl verworfener Pakete mit der Gruppenmitgliedszahl ansteigt. Dies kann wiederum dadurch erklärt werden, dass aufgrund der zufälligen Verteilung der Gruppenmitglieder mit einer steigenden Gruppenmitgliedszahl eine zunehmende Zahl von Endgeräten innerhalb des drahtlosen Ad-hoc-Netzes in die Weiterleitung von Paketen involviert wird. Wegen der Mobilität von Endgeräten wächst somit auch die Gesamtzahl brechender Übertragungsabschnitte und somit der durch das Medienzugriffsprotokoll festgestellten Unerreichbarkeiten.

7.3.1.2 Topologieverhalten

In diesem Abschnitt wird auf das Verhalten von Overlay-Topologien eingegangen, im Sinne der in den Simulationen erfassten Strukturänderungen und der durch die Topologieverwaltung verursachten Routen-Aufbauten durch das AODV-Protokoll zur Unicast-Paketvermittlung.

Entsprechende Messungen sind in Abbildung 7.5 dargestellt, wobei hier zunächst eine *kumulierte* Darstellung über die Simulationszeit hinweg aufgetragen ist. Dabei fällt auf, dass mit steigender Gruppenmitgliedszahl (Vergleich der Diagramme von oben nach unten) ein deutlicher Anstieg der innerhalb von Overlay-Topologien stattfindenden Struk-

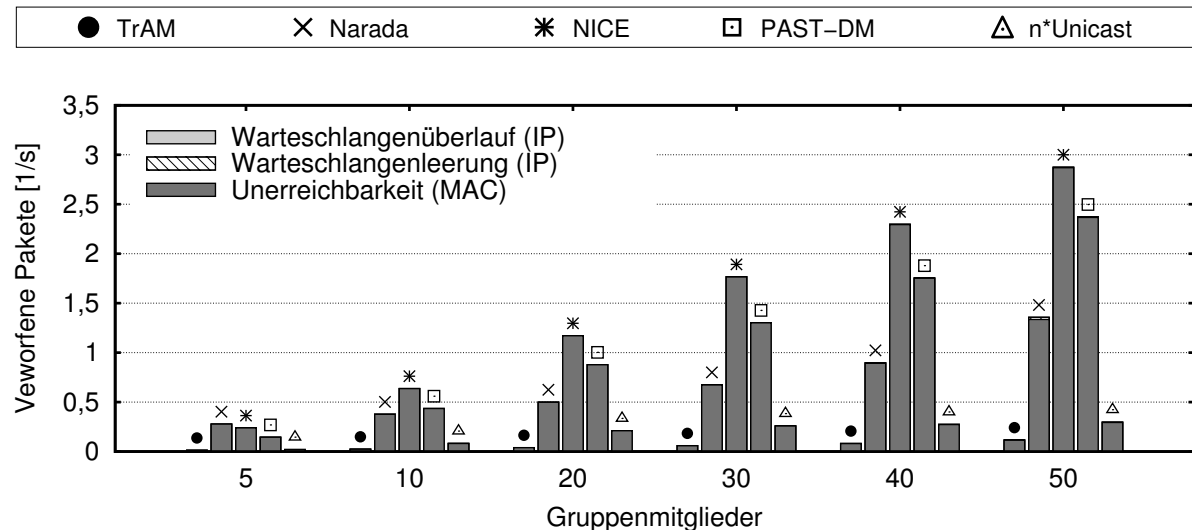


Abbildung 7.4 Pro Simulationssekunde verworfene Pakete und deren Ursache.

turänderungen (linke Diagrammspalte) festgestellt werden kann. Dieser betrifft insbesondere das in dieser Arbeit entworfene TrAM-Protokoll, für welches der Anstieg, wie später verdeutlicht, überlinear erfolgt. Dies kann auf die Tatsache zurückgeführt werden, dass TrAM auf Redundanz innerhalb seiner Overlay-Topologie verzichtet. So ergeben sich bei TrAM weitaus mehr Möglichkeiten zur Topologieanpassung als bei anderen Overlay-Topologien, da diese die Gruppenmitglieder stärker miteinander vermaschen. Aus diesem Grund erfordert die Mobilität von Endgeräten bei diesen Overlay-Topologien nicht notwendigerweise eine Anpassung der Overlay-Topologie selbst sondern lediglich eine Anpassung der Wegewahl, d.h. der Multicast-Verteilbäume, *innerhalb* der Overlay-Topologie. Am deutlichsten tritt dies beim PAST-DM-Protokoll zum Vorschein, bei welchem Strukturänderungen hauptsächlich bei der Integration neuer Gruppenmitglieder in die Overlay-Topologie im Zuge der Gruppenbeitrittsphase stattfinden. Nach deren Abschluss sind die Gruppenmitglieder relativ stark miteinander vermascht, so dass kaum noch Strukturänderungen erfolgen und Anpassungen vorwiegend in den Verteilbäumen durchgeführt werden. Für das NICE-Protokoll ist auch bei hohen Gruppenmitgliedszahlen lediglich ein langsamer Anstieg der Strukturänderungen zu erkennen, was auf die vergleichsweise geringe Flexibilität der hierarchischen Cluster-Struktur zurückgeführt werden kann. Für das n*Unicast-Protokoll sind in den Diagrammen keine Strukturänderungen eingetragen, da dieses Protokoll aufgrund seiner Natur keine Topologieanpassungen erfordert und daher auch nicht integriert.

In der rechten Diagrammspalte sind (ebenfalls kumuliert) die Anzahl der von AODV aufgebauten Routen dargestellt. Hier ist insbesondere zu erkennen, dass PAST-DM und NICE, trotz ihrem sehr langsamen Anstieg der Strukturänderungszahl, die höchste Anzahl von Routen-Aufbauten verursachen. Das TrAM-Protokoll hingegen, welches den

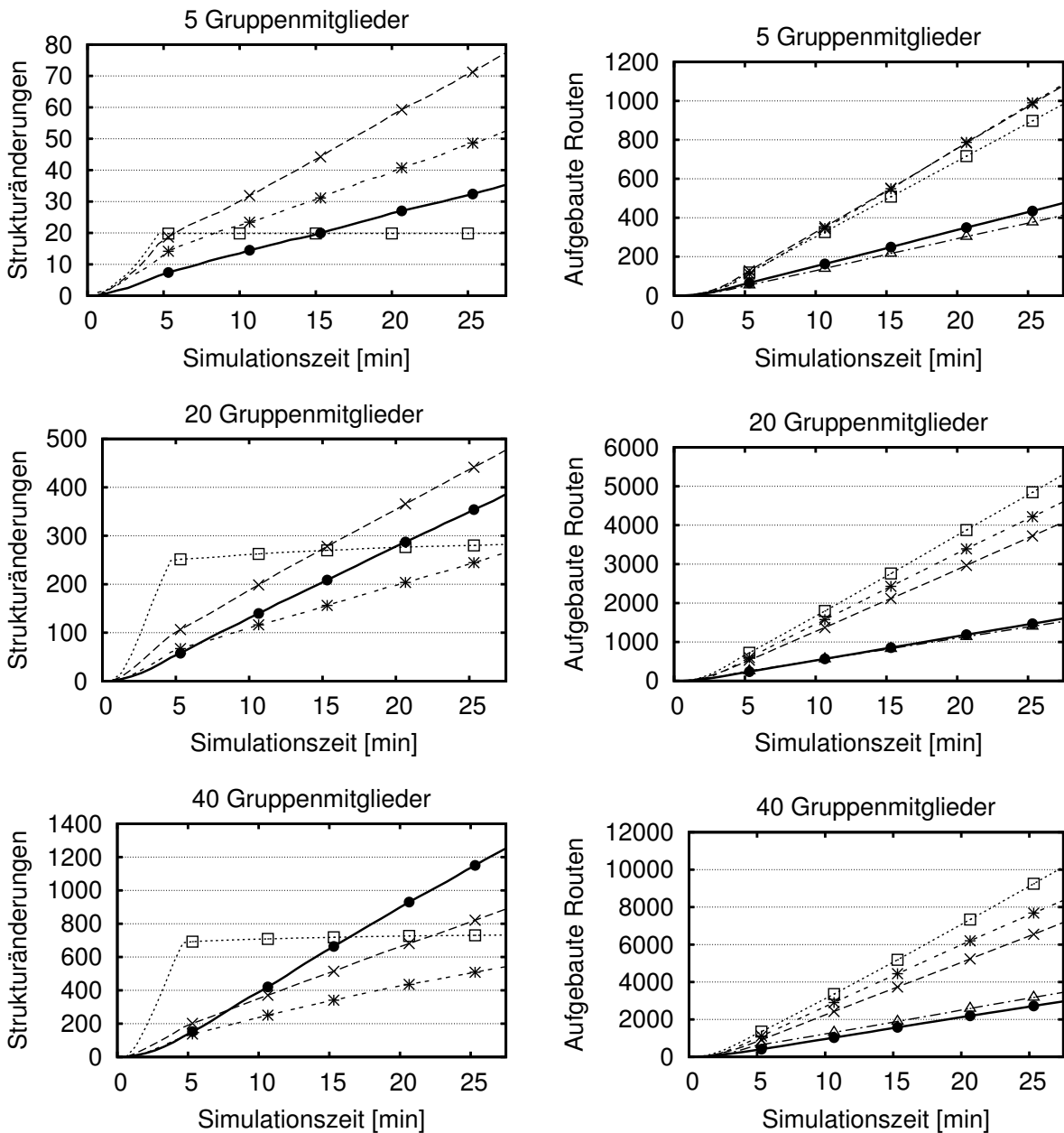


Abbildung 7.5 Kumulierte Overlay-Strukturänderungen und Routen-Aufbauten.

deutlichsten Anstieg der Strukturänderungszahl aufweist, verfügt nun über die geringste Anzahl an von AODV aufgebauten Routen.

In Abbildung 7.6 sind wiederum die pro Simulationssekunde durchschnittlich beobachteten Strukturänderungen innerhalb der Overlay-Topologien und die von AODV aufgebauten Routen jeweils in Abhängigkeit der Gruppengröße aufgetragen. Im Hinblick auf die von TrAM durchgeführten Strukturänderungen (linkes Diagramm) ist hier

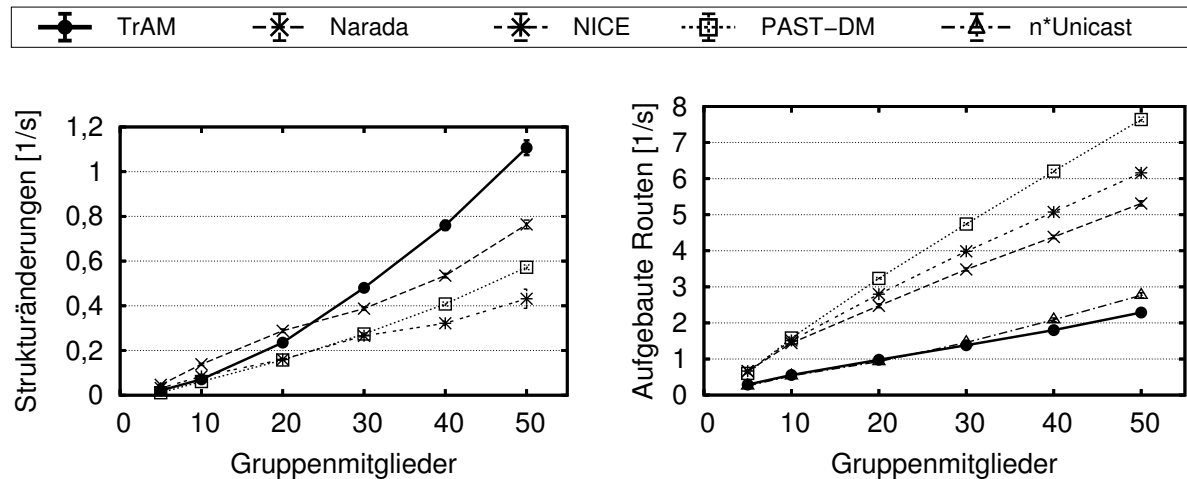


Abbildung 7.6 Pro Simulationssekunde stattfindende Overlay-Strukturänderungen und Routen-Aufbauten.

sehr gut der überlineare Anstieg zu erkennen. Hinsichtlich der tatsächlich aufgebauten Routen jedoch (rechtes Diagramm), weist TrAM von allen Protokollen den langsamsten Anstieg auf: Hier zeigt sich die Effizienz des in TrAM integrierten Mechanismus zur Lokalisierung und Bestimmung von Elternknoten sowie die allgemein sehr geringe Anzahl von Overlay-Verbindungen. Insbesondere im Hinblick auf die Skalierungen der y-Achse, sprechen auch hier die lediglich sehr kleinen Konfidenzintervalle für eine hohe Signifikanz der erhaltenen Messwerte.

7.3.2 Auswirkungen der Kombination mit LBCs

Nachdem in den vorangegangenen Abschnitten die eigentlichen Overlay-Topologien untersucht wurden, werden nun die Auswirkungen der Kombination der Overlay-Topologien mit dem Konzept lokaler Broadcast-Cluster betrachtet. Analog zu Abschnitt 7.3.1 werden dabei zunächst der Kontrollfluss und die Medienzugriffszeit herangezogen und anschließend das Topologieverhalten studiert.

7.3.2.1 Kontrollfluss und Medienzugriffszeit

Der von den einzelnen Overlay-Topologien in Kombination mit LBCs auf Anwendungsschicht pro Simulationssekunde verursachte Kontrollfluss sowie die dabei induzierte Medienzugriffszeit ist in Abbildung 7.7 aufgetragen. Um dabei Vergleichsmöglichkeiten zu den zuvor erzielten Ergebnissen zu bieten, erfolgt die Darstellung für die gleichen Gruppengrößen und über die Simulationszeit hinweg.

Insbesondere beim Betrachten des Kontrollflusses (linke Diagrammspalte) sei zunächst erwähnt, dass im Vergleich zu Abbildung 7.1 stark unterschiedliche Skalierungen der y-Achse verwendet werden: So kann durch Verwendung des LBC-Konzepts der von Overlay-Topologien erforderte Kontrollfluss deutlich gesenkt werden, da die Anzahl der in

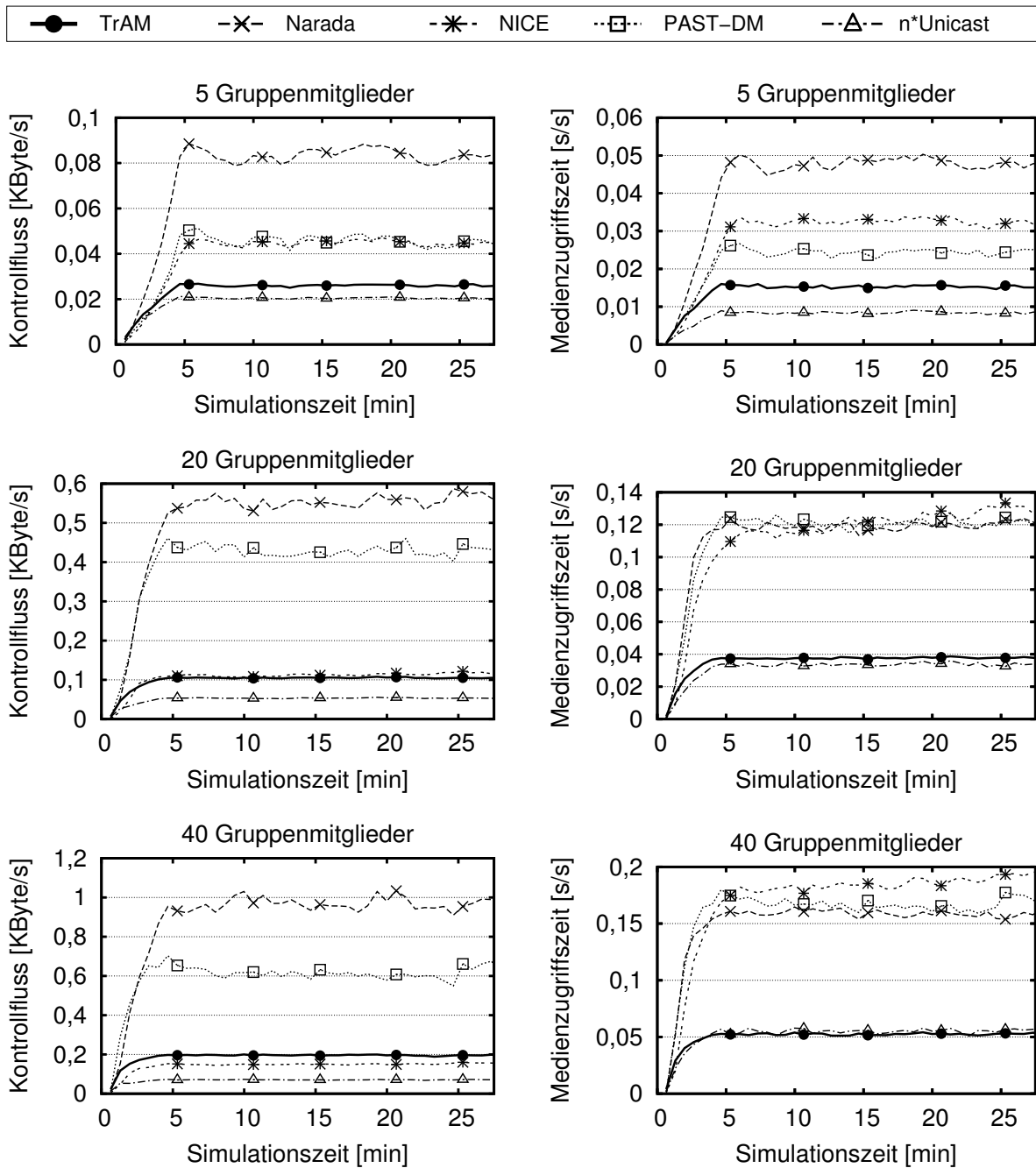


Abbildung 7.7 Verlauf des Kontrollflusses und der Medienzugriffszeit über die Simulationszeit hinweg unter Verwendung des LBC-Konzepts.

die Overlay-Topologie eingegliederten Gruppenmitglieder deutlich verringert wird. Insbesondere bei 20 und 40 Gruppenmitgliedern sind hier Reduktionen um einen Faktor 5 bis hin zu mehr als einer Größenordnung erkennbar. Obgleich sich diese Verbesserungen hauptsächlich im Zusammenhang mit den auf expliziter Routing-Information basierenden Overlay-Topologien PAST-DM und Narada bemerkbar machen, sind auch bei NICE

und TrAM Reduktionen des Kontrollflusses sichtbar: So liegen bei diesen Protokollen die erzielten Verbesserungen bei einem Faktor 2 bis 3.

Die Reduktion des Kontrollflusses ist begleitet von einer Absenkung der erfordernten Medienzugriffszeit (rechte Diagrammspalte). Wie aufgrund von Abschnitt 7.3.1 bereits vermutet werden kann, fällt die Reduktion der Medienzugriffszeit geringer aus als die für den Kontrollfluss erzielte Verbesserung. Dennoch kann im Vergleich zu Abbildung 7.7 die Medienzugriffszeit für einige Protokolle um den Faktor 2 bis 3 gesenkt werden.

In Abbildung 7.8 ist der pro Simulationssekunde durchschnittlich anfallende Kontrollfluss sowie die Medienzugriffszeit in Abhängigkeit der Gruppengröße aufgetragen. Bei Betrachtung des Kontrollflusses (linkes Diagramm) kann nun für die Protokolle Narada und PAST-DM ein deutliches Abflachen der Kurven beobachtet werden. Dieses resultiert aus der zunehmenden Abdeckung des Simulationsareals anhand von LBCs. So steigt mit zunehmender Gruppenmitgliedszahl die Wahrscheinlichkeit eines lokalen Gruppenbeitritts für neu beitretende Gruppenmitglieder, so dass diese keinen Kontrollfluss verursachen. Für das Protokoll Narada lässt sich die Abflachung des Kontrollflusses direkt auf die induzierte Medienzugriffszeit (rechtes Diagramm) übertragen. Ebenso für das Protokoll PAST-DM, wobei hier jedoch die Abflachung weniger drastisch ausfällt. Dennoch kann, in Abhängigkeit der Gruppengröße, für PAST-DM und auch für NICE die erforderliche Medienzugriffszeit durch Kombination der Overlay-Topologien mit dem LBC-Konzept um den Faktor 2 reduziert werden. Analog zu Abbildung 7.3 verursacht wiederum TrAM zusammen mit dem n*Unicast-Protokoll auch bei Verwendung des LBC-Konzepts die mit Abstand geringste Medienzugriffszeit. Wie auch bei den zuvor diskutierten Diagrammen sprechen die in Abbildung 7.8 dargestellten Konfidenzintervalle für eine hohe Signifikanz der Messergebnisse.

Da durch Verwendung des LBC-Konzepts die Anzahl der in eine Overlay-Topologie eingebundenen Gruppenmitglieder deutlich reduziert wird, fällt weniger Kommunikation über Unicast-Pakete an. Als Folge dessen wird erwartet, dass innerhalb des drahtlosen Ad-hoc-Netzes entsprechend weniger Pakete verworfen werden. Dies kann anhand von Abbildung 7.9 belegt werden, in welcher die pro Simulationssekunde durchschnittlich verworfenen Pakete in Abhängigkeit der Gruppengröße aufgetragen sind. Im Vergleich zu Abbildung 7.4 zeigt sich, dass bei Verwendung lokaler Broadcast-Cluster die Anzahl der verworfenen Pakete in etwa halbiert wird.

7.3.2.2 Topologieverhalten

In diesem Abschnitt wird das Verhalten der mit dem LBC-Konzept kombinierten Overlay-Topologien betrachtet, wobei auch hier die in den Simulationen erfassten Strukturänderungen und die durch die Topologieverwaltung verursachten Routen-Aufbauten durch das AODV-Protokoll herangezogen werden. Die Ergebnisse der durchgeführten Messungen sind in Abbildung 7.10 dargestellt.

7. Protokollauswertung

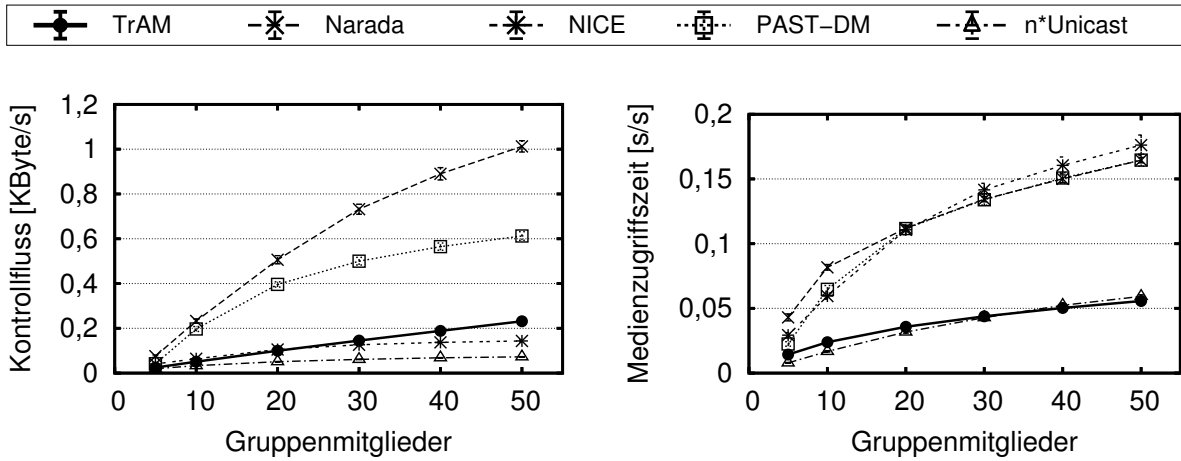


Abbildung 7.8 Pro Simulationssekunde gemessener Kontrollfluss und Medienzugriffszeit bei Verwendung des LBC-Konzepts.

Aufgrund der durch die Mobilität der Endgeräte erfolgenden Kollisionen und Verluste lokaler Broadcast-Cluster wirkt sich das LBC-Konzept auf die beobachteten Strukturänderungen (linke Diagrammspalte) innerhalb von Overlay-Topologien in Form einer Steigerung aus. Besonders deutlich zeigt sich dies bei dem Protokoll PAST-DM, für welches in Abbildung 7.5 noch ein deutliches Abflachen der Kurve nach Beendigung der Gruppenbeitrittsphase beobachtet werden konnte. Diese Abflachung bleibt bei Verwendung von LBCs aufgrund der kontinuierlichen Ein- und Ausgliederung von Gruppenmitgliedern in die bzw. aus der Overlay-Topologie aus. Einzig für TrAM kann für höhere Gruppenmitgliedszahlen eine deutliche Absenkung der Strukturänderungen beobachtet

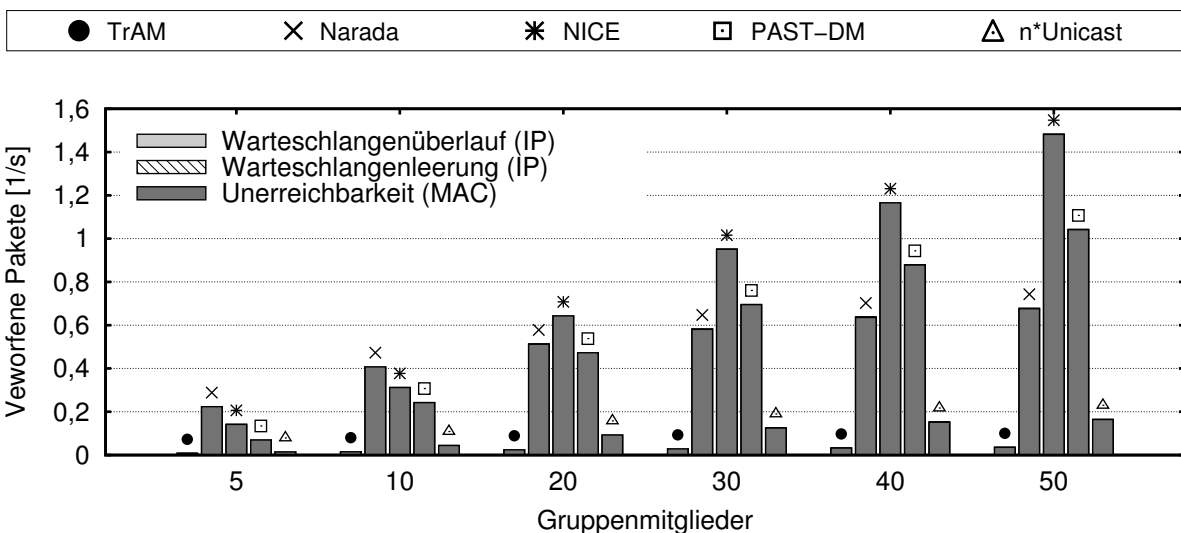


Abbildung 7.9 Pro Simulationssekunde verworfene Pakete unter Verwendung des LBC-Konzepts.

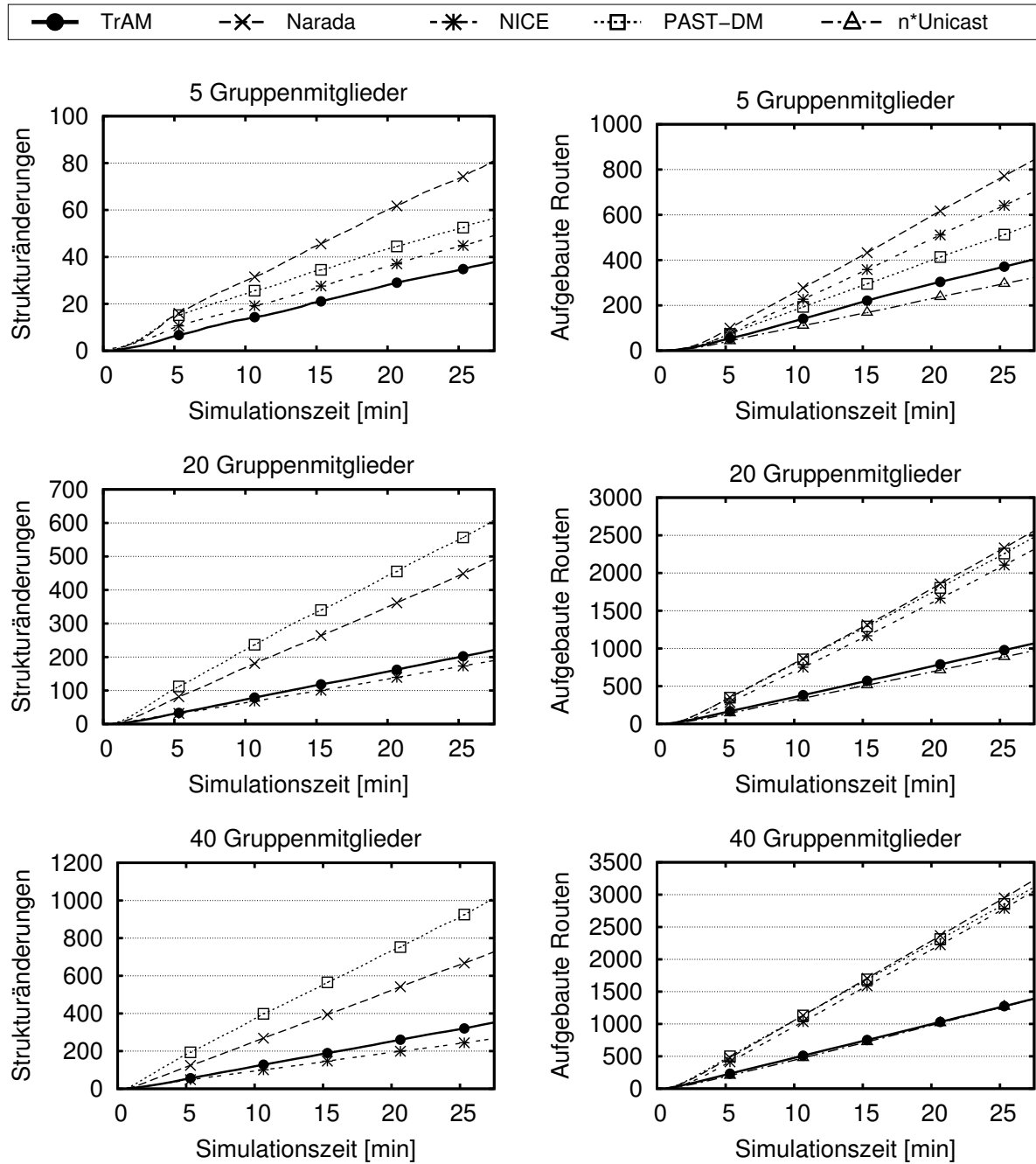


Abbildung 7.10 Kumulierte Overlay-Strukturänderungen und Routen-Aufbauten unter Verwendung des LBC-Konzepts.

werden. Dies ergibt sich aus zwei Gründen: Erstens ist die Anzahl der Gruppenmitglieder, welche als potenzielle Elternknoten in Betracht kommen, aufgrund der zahlreichen lokalen Gruppenbeitritte deutlich geringer, so dass die Möglichkeiten zur Durchführung von Strukturänderungen reduziert sind. Während dies allerdings auch auf andere Overlay-Topologien zutrifft, ist der bei TrAM entscheidende Grund, dass die Overlay-

Eingliederung eines Gruppenmitglieds (beispielsweise im Falle des Verlusts eines LBCs) durch den Aufbau von lediglich einer einzigen Overlay-Verbindung erreicht wird: Im Vergleich dazu werden bei PAST-DM Overlay-Verbindungen zu mehreren dem Overlay beigetretenen Gruppenmitgliedern aufgebaut, so dass die Anzahl der Strukturänderungen hier entsprechend höher ausfällt. Ebenso kann bei Narada ein leichter Rückgang der Strukturänderungen beobachtet werden: Auch dieser ist auf die eher leichte Vermaschung von Gruppenmitgliedern innerhalb von Naradas Overlay-Topologie und dem damit verbundenen, relativ geringen Eingliederungsaufwand zurückzuführen.

Bei Betrachtung der aufgebauten Routen (rechte Diagrammspalte) zeigt sich wiederum eine deutliche Absenkung der gemessenen Verläufe. So werden im Vergleich zu Abbildung 7.5 die bei Verwendung des LBC-Konzepts von PAST-DM, Narada und NICE aufgebauten Routen um etwa einen Faktor 3 reduziert. Mit einem Faktor von etwa 2 fällt diese Verbesserung bei TrAM geringer, aber dennoch deutlich bemerkbar aus.

Eine Übersicht der pro Simulationssekunde durchgeführten Strukturänderungen und der dabei aufgebauten Routen ist jeweils in Abhängigkeit der Gruppenmitgliedszahl in Abbildung 7.11 dargestellt. Es kann erneut festgestellt werden, dass sich für PAST-DM das LBC-Konzept in einer deutlichen Erhöhung der Strukturänderungszahl (linkes Diagramm) zeigt. Während für Narada und NICE dagegen deutlich abflachende Kurvenverläufe erkennbar sind, ist die für TrAM erzielte Absenkung der Strukturänderungsrate im Vergleich zu Abbildung 7.6 am drastischsten.

Deutliche Verbesserungen für alle Overlay-Topologien zeigen sich wiederum in der Anzahl der durchschnittlich pro Simulationssekunde aufgebauten Routen (rechtes Diagramm): Neben einem (teilweise deutlich) abflachenden Verlauf der Kurven ist hier insbesondere zu bemerken, dass durch Verwendung des LBC-Konzepts die Anzahl der erforderlichen Routen-Aufbauten auf weniger als die Hälfte reduziert werden kann.

7.4 Auswertung einer CBR-Anwendung

In diesem Abschnitt erfolgt die Auswertung einer CBR-Anwendung, bei welcher von einer Multicast-Quelle in einem gleich bleibenden zeitlichen Abstand Datenpakete der gleichen Größe versendet werden. Während die Größe der Datenpakete über die gesamte Evaluation hinweg einen Wert von *512 Byte* annimmt, wird durch Variation des zeitlichen Abstands von Datenpaketen die von der Anwendung ausgesendete Datenrate variiert. Analog zu der vorangegangenen Auswertung von Overlay-Topologien erfolgt die Evaluation zunächst ohne und anschließend mit dem Konzept lokaler Broadcast-Cluster. Alle Simulationen in diesem Abschnitt erstrecken sich über 10 Simulationsminuten, von welchen die ersten 5 auf die Gruppenbeitrittsphase entfallen. Daran schließt sich eine für 5 Minuten andauernde Datenverteilung an.

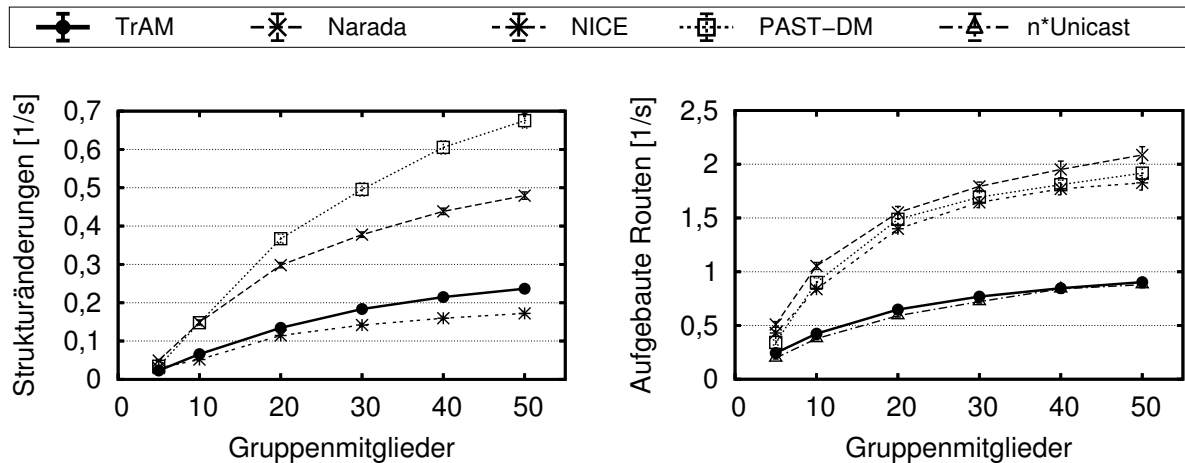


Abbildung 7.11 Overlay-Strukturänderungen und aufgebaute Routen unter Verwendung des LBC-Konzepts.

7.4.1 Standard P2P-Multicast

Zur Untersuchung der CBR-Anwendungen werden unterschiedliche Größen gemessen. Begonnen wird dabei mit den in der Fachliteratur herangezogenen Größen wie Zustellrate und Latenzen der Datenpakete. Um abschätzen zu können, inwieweit eine Datenverteilung Einfluss auf die jeweils verwendete Overlay-Topologie hat, werden im Anschluss die gemessenen Strukturänderungen innerhalb der Overlay-Topologien diskutiert. Abschließend wird die für die Datenverteilung erforderliche Medienzugriffszeit dargestellt.

7.4.1.1 Zustellraten und Latenzen

Die von den einzelnen Overlay-Topologien erzielten Zustellraten und Latenzen sind in Abbildung 7.12 aufgetragen. Während innerhalb eines Diagramms jeweils die für unterschiedliche Gruppengrößen erzielten Ergebnisse dargestellt sind, wird innerhalb einer Diagrammspalte von oben nach unten die Datenrate erhöht.

So fällt zunächst auf, dass bei einer geringen Datenrate von einem Paket pro Sekunde alle Protokolle für die betrachteten Gruppenmitgliedszahlen eine Zustellrate (linke Diagrammspalte) von über 90% erzielen. Bei Erhöhung der Datenrate auf 2 Pakete pro Sekunde sind für das n*Unicast- sowie für das PAST-DM-Protokoll bereits Einbrüche ab 30 Gruppenmitgliedern zu erkennen: Da zur Weiterleitung der Datenpakete an eine steigende Gruppenmitgliedszahl auch zunehmend Bandbreite beansprucht wird, stößt das drahtlose Ad-hoc-Netz bei diesen Protokollen zunehmend an die Grenzen des bewältigbaren Datenvolumens. Bei einer Rate von 4 Paketen pro Sekunde werden diese Einbrüche nicht nur deutlicher, sondern sind jetzt auch für die Protokolle NICE und Narada erkennbar. Einzig TrAM kann bei dieser Datenrate selbst bei 50 Gruppenmitgliedern noch eine Zustellrate von über 95% erzielen. Wie in den vorangegangenen

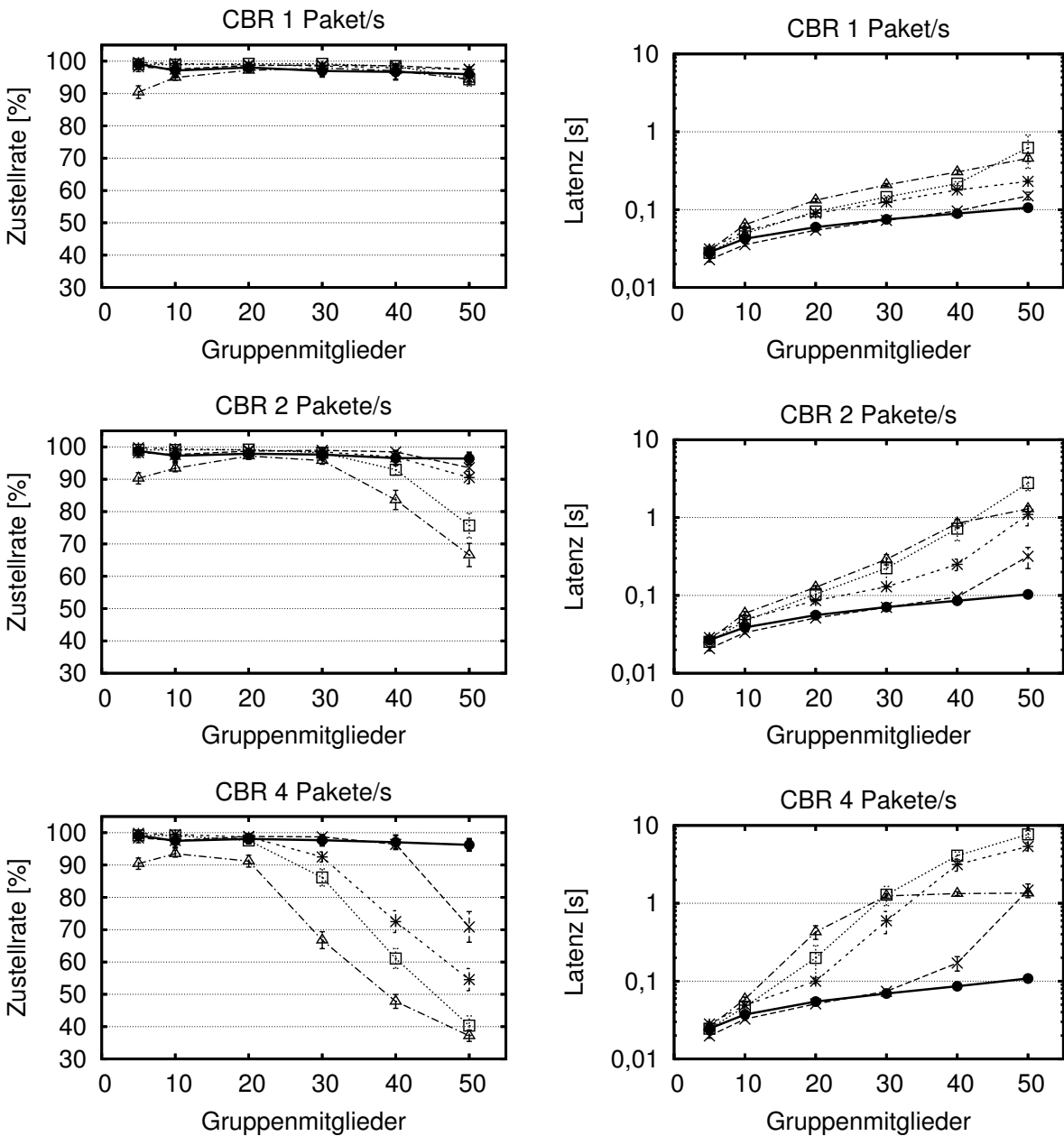


Abbildung 7.12 Zustellraten und Latenzen einer CBR-Anwendung.

Abschnitten verdeutlicht wurde, kann dies auf die Tatsache zurückgeführt werden, dass TrAM im Vergleich zu PAST-DM, Narada und NICE das drahtlose Ad-hoc-Netz aufgrund seiner effizienten Topologieverwaltung weitaus weniger belastet und somit mehr Bandbreite zur Vermittlung von (Nutz)Datenpaketen einsetzen kann. Allgemein fällt bei Betrachtung der Zustellraten auf, dass die eingezeichneten Konfidenzintervalle zusammen mit der Gruppenmitgliedszahl und der ausgesendeten Datenrate wachsen. Dies

ist auf die zunehmende Belastung des drahtlosen Ad-hoc-Netzes zurückzuführen: Insbesondere in Überlastsituationen kann der Empfang bzw. der Verlust von Paketen starken Schwankungen unterliegen, so dass keine präzisen Aussagen über erzielte Zustellraten mehr getroffen werden können.

Beim Betrachten der gemessenen Latenzen (rechte Diagrammspalte) ist deutlich erkennbar, dass sowohl eine Erhöhung der Gruppenmitgliedszahl als auch der Datenrate von einer mitunter sehr starken Erhöhung der beobachteten Latenzen begleitet wird. Dies kann durch die in Abschnitt 2.1.1.3 beschriebene Sicherung von Unicast-Dateneinheiten durch das Medienzugriffsprotokoll IEEE 802.11 erklärt werden. Da mit steigender Gruppengröße und Datenrate mehr Medienzugriffe erfolgen, steigt das Kollisionsrisiko im Laufe des Medienzugriffs. Aufgrund der beschriebenen Sicherungsmechanismen erfolgen Übertragungswiederholungen von unbestätigt gebliebenen Unicast-Dateneinheiten, wobei mit steigender Übertragungswiederholungszahl exponentiell wachsende Wartezeiten zur Reduktion der Kollisionswahrscheinlichkeit verwendet werden. Insbesondere bei der Weiterleitung von Datenpaketen über eine aus mehreren Übertragungsabschnitten bestehende Overlay-Verbindung können sich Verzögerungen entsprechend aufsummieren und so, wie in Abbildung 7.12 zu sehen, bis zu mehreren Sekunden betragen.

7.4.1.2 Stabilität der Overlay-Topologien

In Abbildung 7.13 sind die durchschnittlich pro Simulationssekunde beobachteten Strukturänderungen innerhalb der jeweiligen Overlay-Topologien aufgetragen, wobei hier die beiden Datenraten von einem und 4 Paketen pro Sekunde betrachtet werden. Aufgrund der unterschiedlichen Simulationsdauern sind die in diesem Abschnitt gezeigten Messungen mit denen aus Abbildung 7.6 nicht direkt vergleichbar.

Bei Betrachtung von Abbildung 7.13 kann festgestellt werden, dass die Effekte auf die Overlay-Topologien unterschiedlich ausfallen. Während beispielsweise bei PAST-DM keine Änderung des Topologieverhaltens festgestellt werden kann, ist bei TrAM eine geringe und bei NICE sowie Narada eine etwas deutlichere Vergrößerung der Strukturänderungsrate und der zugehörigen Konfidenzintervalle zu erkennen. Dies ist darauf zurückzuführen, dass die zur Bewertung von Overlay-Verbindungen gemessenen Größen mit steigender Belastung des drahtlosen Ad-hoc-Netzes unterschiedlich stark beeinflusst werden. So wird die auf Hop-Counts basierende Metrik von TrAM nur gering beeinflusst, während das auf Paketumlaufzeiten (RTTs) zurückgreifende Narada stärkeren Schwankungen unterliegt. Dahingegen kann bei NICE die bereits stattfindende Überlastung des drahtlosen Ad-hoc-Netzes als Grund für die deutlicher schwankenden Messwerte gesehen werden. PAST-DM wiederum setzt die Topologieanpassung vorwiegend durch Änderung der Multicast-Verteilbäume innerhalb der stark vermaschten Overlay-Topologie um, so dass diese unabhängig von der Datenrate weitgehend unverändert bleibt.

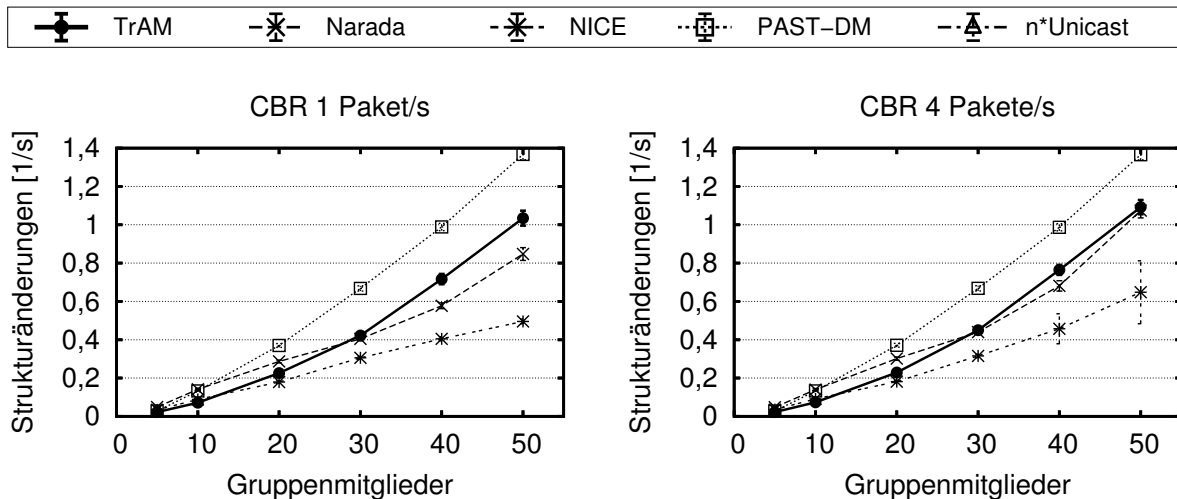


Abbildung 7.13 Pro Simulationssekunde durchgeführte Anzahl von Overlay-Strukturänderungen bei einer CBR-Anwendung.

7.4.1.3 Medienzugriffszeit und verworfene Pakete

Die für die Verwaltung der Overlay-Topologien sowie die zur Durchführung der Datenverteilungen pro Simulationssekunde durchschnittlich erforderlichen Medienzugriffszeiten sind in Abbildung 7.14 für die Datenraten von einem und 4 Paketen pro Sekunde in Abhängigkeit der Gruppengröße dargestellt. Auch hier sind aufgrund unterschiedlicher Simulationsdauern die gezeigten Messungen mit denen aus Abbildung 7.3 nicht direkt vergleichbar.

Es ist zu erkennen, dass eine Erhöhung der Gruppenmitgliedszahl ebenso wie eine Erhöhung der Paketrate von einem Paket pro Sekunde (linkes Diagramm) auf 4 Pakete pro Sekunde (rechtes Diagramm) einen entsprechenden Anstieg der Medienzugriffszeit zur Folge hat. Während dieser Anstieg bei einer Datenrate von einem Paket pro Sekunde für alle Protokolle annähernd linear mit der Gruppenmitgliedszahl verläuft, ist bei einer Datenrate von 4 Paketen pro Sekunde ein Abflachen der Kurven für PAST-DM, NICE und n*Unicast zu beobachten. Wie später bei der Diskussion verworfener Pakete verdeutlicht werden wird, stoßen die genannten Protokolle bei dieser Datenrate und einer hohen Gruppenmitgliedszahl bereits an die Grenzen des Verkehrsvolumens, welches durch das drahtlose Ad-hoc-Netz übertragen werden kann. Im Gegensatz dazu verfügt das TrAM-Protokoll aufgrund der auch bei 50 Gruppenmitgliedern noch relativ geringen Medienzugriffszeit über ausreichend Kapazitäten, um zusätzliche Datenpakete zu vermitteln.

Zur Verdeutlichung der soeben besprochenen Kurvenverläufe sind in Abbildung 7.15 die pro Simulationssekunde durchschnittlich verworfenen Pakete für die zuvor betrachteten Datenraten und Gruppengrößen aufgetragen. Bei einer Datenrate von einem Paket pro

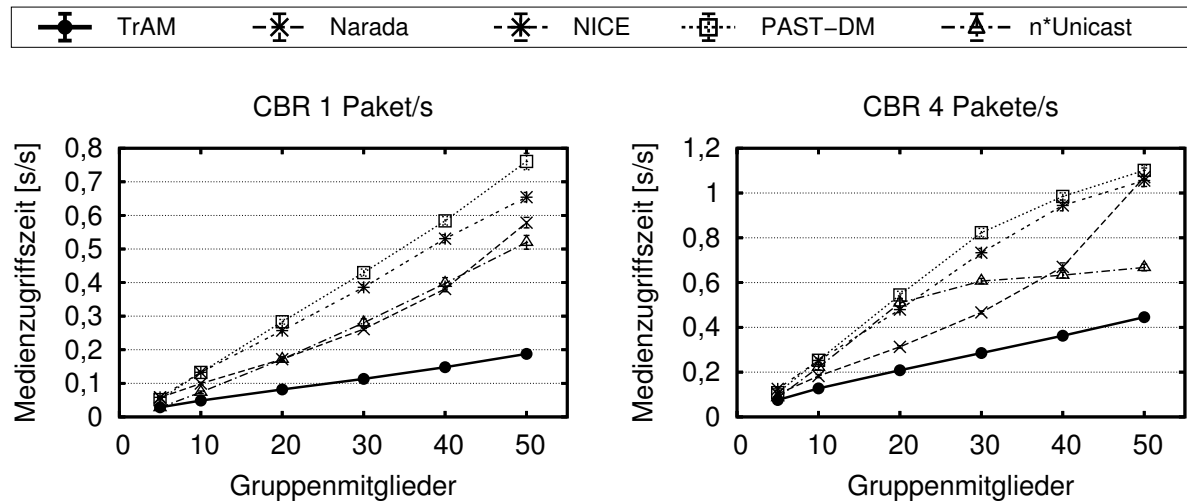


Abbildung 7.14 Pro Simulationssekunde erfolgende Medienzugriffszeit bei einer CBR-Anwendung.

Sekunde werden Pakete hauptsächlich noch aufgrund der für Abbildung 7.4 diskutierten Mobilität und den daraus resultierenden Unerreichbarkeiten verworfen. PAST-DM bildet hier als einziges Protokoll eine Ausnahme, denn wie zu erkennen ist werden bei 50 Gruppenmitgliedern zahlreiche Pakete aufgrund von Überläufen der IP-Paketwarteschlange verworfen. Ebenso werden hier für das n*Unicast-Protokoll bereits zunehmend Pakete auf Vermittlungsschicht verworfen, sei es aufgrund von Warteschlangenleerungen oder -überläufen.

Bei Erhöhung der Datenrate auf 4 Pakete pro Sekunde wird deutlich, dass Narada, NICE, PAST-DM und n*Unicast mit steigender Gruppenmitgliedszahl zunehmend an die Grenzen des Verkehrsvolumens stoßen, welches über das drahtlose Ad-hoc-Netz übertragbar ist. So werden bei diesen Protokollen und 50 Gruppenmitgliedern zahlreiche Pakete aufgrund von Warteschlangenüberläufen verworfen. Diese Paketverwürfe stehen in direktem Zusammenhang zu den in Abbildung 7.12 dargestellten einbrechenden Zustellraten und rasant steigenden Latenzen. Da das Medienzugriffsprotokoll aufgrund von Kollisionen pro Paket mehrere Übertragungswiederholungen durchführen muss, beansprucht die Übertragung eines Pakets mehr Zeit, so dass weniger Pakete pro Sekunde verarbeitet und weitergeleitet werden können. Aus diesem Grund steigt wegen der bereits zügig versendeten Datenpakete der Füllstand der Paketwarteschlange kontinuierlich an und sorgt letztendlich für das Verwerfen zahlreicher Pakete. Dies betrifft insbesondere das n*Unicast-Protokoll, da hier die Duplizierung der Datenpakete allein durch die Multicast-Quelle stattfindet: Bei einer Datenrate von 4 Paketen pro Sekunde und 50 Gruppenmitgliedern werden somit 200 Pakete pro Sekunde in die Paketwarteschlange eingereiht. Zahlreiche dieser Pakete müssen jedoch aufgrund einer zu langsamen Paketweiterleitung verworfen werden.

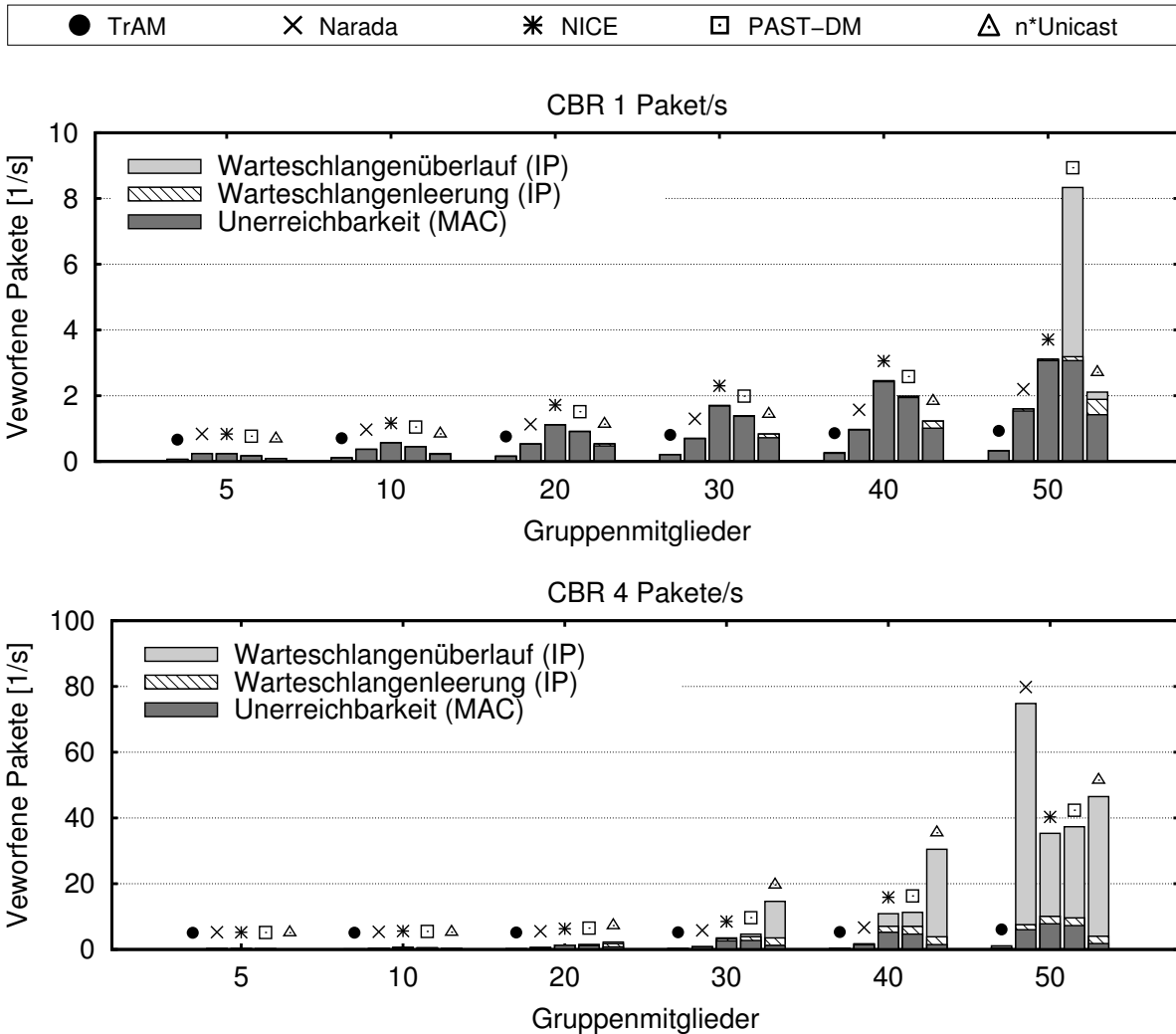


Abbildung 7.15 Pro Simulationssekunde verworfene Pakete bei einer CBR-Anwendung.

7.4.2 Auswirkungen der Kombination mit LBCs

In diesem Abschnitt werden die Auswirkungen einer Kombination der Overlay-Topologien mit dem LBC-Konzept im Kontext einer CBR-Anwendung untersucht. Um dabei den Vergleich zu dem vorangegangenen Abschnitt zu vereinfachen, wird hier ebenfalls mit den Standardgrößen Zustellrate und Latenzen begonnen. Während anschließend der Einfluss der Datenverteilung anhand der gemessenen Strukturänderungen innerhalb der Overlay-Topologien diskutiert wird, werden abschließend die Medienzugriffszeiten sowie die Paketverwürfe betrachtet.

7.4.2.1 Zustellraten und Latenzen

Die Zustellraten und Latenzen wie sie bei Verwendung des LBC-Konzepts gemessen wurden sind in Abbildung 7.16 aufgetragen. Im Vergleich zu Abbildung 7.12 sind hier

mitunter sehr deutliche Leistungssteigerungen erkennbar. So erzielen nun sowohl das n*Unicast- als auch das NICE- und das PAST-DM-Protokoll bei einer Datenrate von 4 Paketen pro Sekunde und 50 Gruppenmitgliedern eine Zustellrate von über 90%: Aufgrund der weitaus geringeren Anzahl der zur Weiterleitung von Datenpaketen erforderlichen Medienzugriffe sinkt die Belastung des drahtlosen Ad-hoc-Netzes und somit das Kollisionsrisiko. Insbesondere ist hier zu bemerken, dass das sehr einfach gehaltene n*Unicast-Protokoll hinsichtlich der erzielten Zustellraten aufwendigen Overlay-Topologien teilweise überlegen ist. Begleitet wird die Verbesserung der Zustellraten durch ein deutliches Absenken der gemessenen Latenzen: So können diese im Vergleich zu Abbildung 7.12 durch Verwendung des LBC-Konzepts um eine Größenordnung gedrückt werden. Dies ergibt sich direkt aufgrund der Tatsache, dass das Medienzugriffsprotokoll auch im Falle von Kollisionen bei der Übertragung von Broadcast-Dateneinheiten keine Sicherungsmaßnahmen wie Übertragungswiederholungen vornimmt.

Allgemein fällt für Abbildung 7.16 auf, dass sich bei den dargestellten Kurvenverläufen weder eine Abhängigkeit von der übertragenen Datenrate noch eine Vergrößerung der Konfidenzintervalle erkennen lässt. Dies ist als Anzeichen dafür zu sehen, dass die Protokolle noch nicht im Begriff sind, das drahtlose Ad-hoc-Netz zu überlasten und somit noch Bandbreite zur Übertragung zusätzlicher Datenpakete zur Verfügung steht.

7.4.2.2 Stabilität der Overlay-Topologien

Aufgrund der Untersuchungen und der in Abbildung 7.13 dargestellten Ergebnisse kann erwartet werden, dass sich bei Verwendung des LBC-Konzepts kaum eine Abhängigkeit des Topologieverhaltens von der Datenrate zeigt: Dies beruht auf der Tatsache, dass durch Verwendung des LBC-Konzepts das drahtlose Medium deutlich entlastet wird und somit die Messungen, anhand welcher die Bewertung von Overlay-Verbindungen erfolgt, weniger beeinflusst werden. Diese Überlegung wird durch Abbildung 7.17 bestätigt, in welcher die pro Simulationssekunde erfolgten Overlay-Strukturänderungen in Abhängigkeit der Datenrate und Gruppengröße aufgetragen sind. Hier sind für die dargestellten Datenraten Änderungen des Topologieverhaltens kaum festzustellen.

7.4.2.3 Medienbelastung und verworfene Pakete

Die durch Verwendung des LBC-Konzepts erreichte Entlastung des drahtlosen Mediums zeigt sich wie zu erwarten auch in einer deutlichen Reduktion der pro Simulationssekunde erfolgenden Medienzugriffszeit. Diese ist in Abhängigkeit der Datenrate und Gruppengröße in Abbildung 7.18 abgebildet. Einerseits zeigt sich hier der bereits aus Abbildung 7.8 bekannte, mit der Gruppenmitgliedszahl stark abflachende Verlauf der Kurven, welcher aus der steigenden Wahrscheinlichkeit lokaler Gruppenbeitritte resultiert. Andererseits sind im Vergleich zu Abbildung 7.14 besonders für die Protokolle NICE, PAST-DM, Narada und n*Unicast deutliche Verbesserungen zu erkennen, im

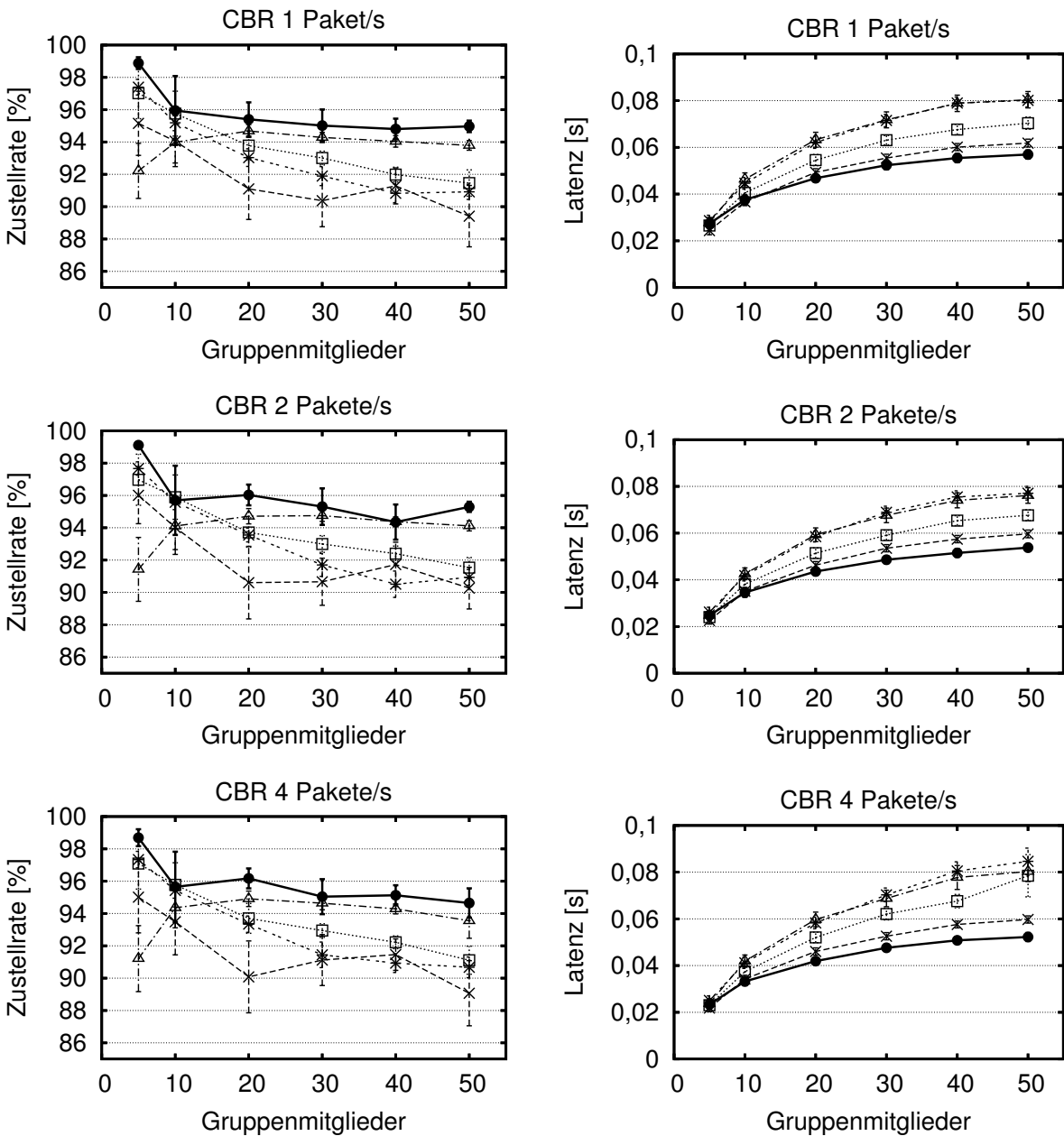


Abbildung 7.16 Zustellraten und Latenzen bei einer CBR-Anwendung unter Verwendung des LBC-Konzepts.

Zuge welcher die Medienzugriffszeit um die Hälfte reduziert wird. Dies verstärkt die Vermutung, dass innerhalb des drahtlosen Ad-hoc-Netzes noch Bandbreite verfügbar ist und zur Vermittlung zusätzlicher Datenpakete genutzt werden kann.

Bestärkt wird diese Aussage noch einmal durch Abbildung 7.19, in welcher die pro Simulationssekunde durchschnittlich verworfenen Pakete für die einzelnen Overlay-To-

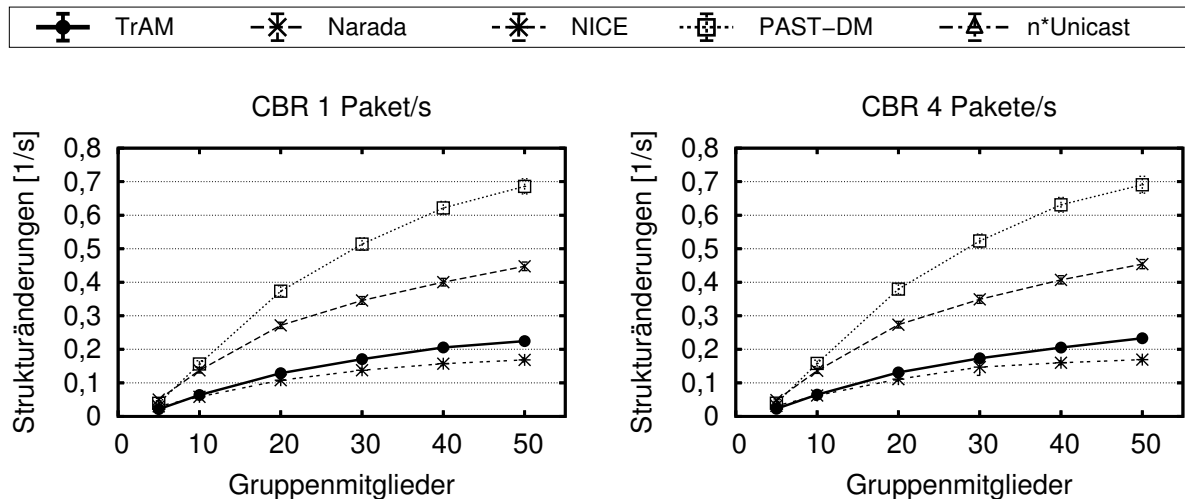


Abbildung 7.17 Pro Simulationssekunde erfolgende Overlay-Strukturänderungen bei einer CBR-Anwendung unter Verwendung des LBC-Konzepts.

pologien und Gruppengrößen aufgetragen sind. Auffällig ist hier zunächst, dass sich die Messungen bei einem und 4 Paketen pro Sekunde kaum voneinander unterscheiden: So kann im Vergleich zu Abbildung 7.15 vor allem bei einer Datenrate von 4 Paketen pro Sekunde ein deutlicher Rückgang der Anzahl verworfener Pakete verzeichnet werden. In der Tat treten bei Verwendung des LBC-Konzepts nahezu keine Paketverwürfe aufgrund von Warteschlangenüberläufen auf. Dies weist darauf hin, dass die in Warteschlangen eingereihten Pakete schnell genug verarbeitet und weitergeleitet werden können.

7.4.3 Hohe Datenraten mit TrAM und Narada

Wegen der in den vorangegangenen Abschnitten mehrfach angebrachten Vermutung, durch Verwendung des LBC-Konzepts höhere Datenraten unterstützen zu können, werden in diesem Abschnitt Untersuchungen hinsichtlich der Skalierbarkeit bezüglich der Datenrate durchgeführt. Die Messungen werden jedoch auf die Protokolle TrAM und Narada sowie auf eine Gruppengröße von 30 Gruppenmitgliedern beschränkt, da diese Protokolle gemäß den bisherigen Untersuchungen das beste Skalierungspotenzial bieten. Ebenso werden ausschließlich die erzielten Zustellraten und Latenzen betrachtet, da die Auswirkungen von Überlastsituationen etwa auf die Anzahl verworfener Pakete aufgrund der vorangegangener Untersuchungen bekannt sind. Die Messungen sind in Abbildung 7.20 aufgetragen, wobei zur Verdeutlichung der Auswirkungen lokaler Broadcast-Cluster die Protokolle jeweils mit und ohne LBC-Erweiterung betrieben wurden.

Bei Betrachtung der Zustellraten (oberes Diagramm) ist die bemerkenswerteste Eigenschaft, dass sich zum Erreichen möglichst hoher Zustellraten der Einsatz lokaler Broadcast-Cluster nicht bei jeder Datenrate empfiehlt. In der Tat kann das Diagramm bei einer Datenrate von 8 Paketen pro Sekunde in zwei Hälften geteilt werden. In der lin-

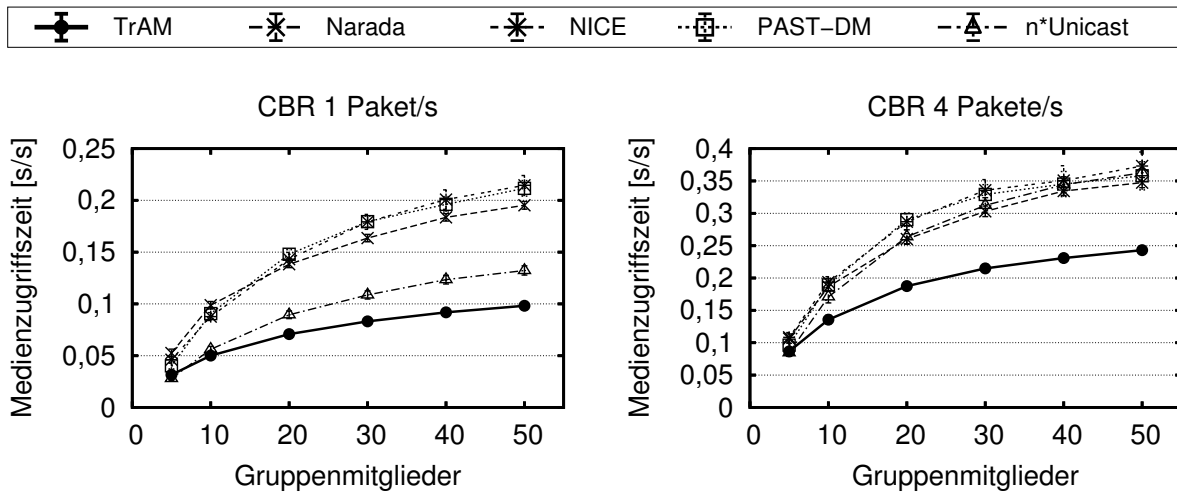


Abbildung 7.18 Pro Simulationssekunde erfolgreiche Medienzugriffszeit bei einer CBR-Anwendung unter Verwendung des LBC-Konzepts.

ken Hälfte (bei Datenraten von weniger als 8 Paketen pro Sekunde) werden von den Overlay-Topologien selbst bessere Ergebnisse erzielt als unter Verwendung des LBC-Konzepts. Erst ab einer Datenrate von 8 Paketen pro Sekunde sind durch Verwendung des LBC-Konzepts Leistungssteigerungen zu erkennen. Diese Beobachtung kann dadurch erklärt werden, dass bei einer Gruppengröße von 30 Gruppenmitgliedern bis zu der Datenrate von 8 Paketen pro Sekunde die Belastung des drahtlosen Ad-hoc-Netzes ausreichend gering ist, so dass die Overlay-Topologien die Zustellung der Datenpakete anhand von Unicast-Übertragungen bewerkstelligen können. Da Unicast-Dateneinheiten den Sicherungsmechanismen des Medienzugriffsprotokolls unterliegen, sind bis zu dieser Datenrate die erzielten Zustellraten höher als bei einer Zustellung anhand ungesicherter Broadcast-Übertragungen, wie sie bei Verwendung des LBC-Konzepts zum Einsatz kommen. Ab der Datenrate von 8 Paketen pro Sekunde beginnen die Unicast-Sicherungsmechanismen sich kontraproduktiv auf die Kommunikation auszuwirken: So sorgen die durchgeführten Übertragungswiederholungen unbestätigt gebliebener Unicast-Dateneinheiten für zusätzliche Kollisionen, welche sich letztendlich durch Paketverwürfe und somit geringer ausfallenden Zustellraten bemerkbar machen.

Aufgrund zunehmender Kollisionen kann ab der Datenrate von 8 Paketen pro Sekunde für die ohne LBC-Erweiterung betriebenen Overlay-Topologien auch ein deutliches Anwachsen der Latenzen (unteres Diagramm) beobachtet werden. Dieses Anwachsen resultiert direkt aus der exponentiellen Vergrößerung der Wettbewerbsfenster im Laufe des Medienzugriffs für Übertragungswiederholungen von Unicast-Dateneinheiten. Trotz der bei geringeren Datenraten (< 8 Paketen pro Sekunde) niedrigeren Zustellraten, kann in Verbindung mit dem LBC-Konzept jedoch in allen untersuchten Fällen eine deutliche Reduktion der gemessenen Latenzen beobachtet werden.

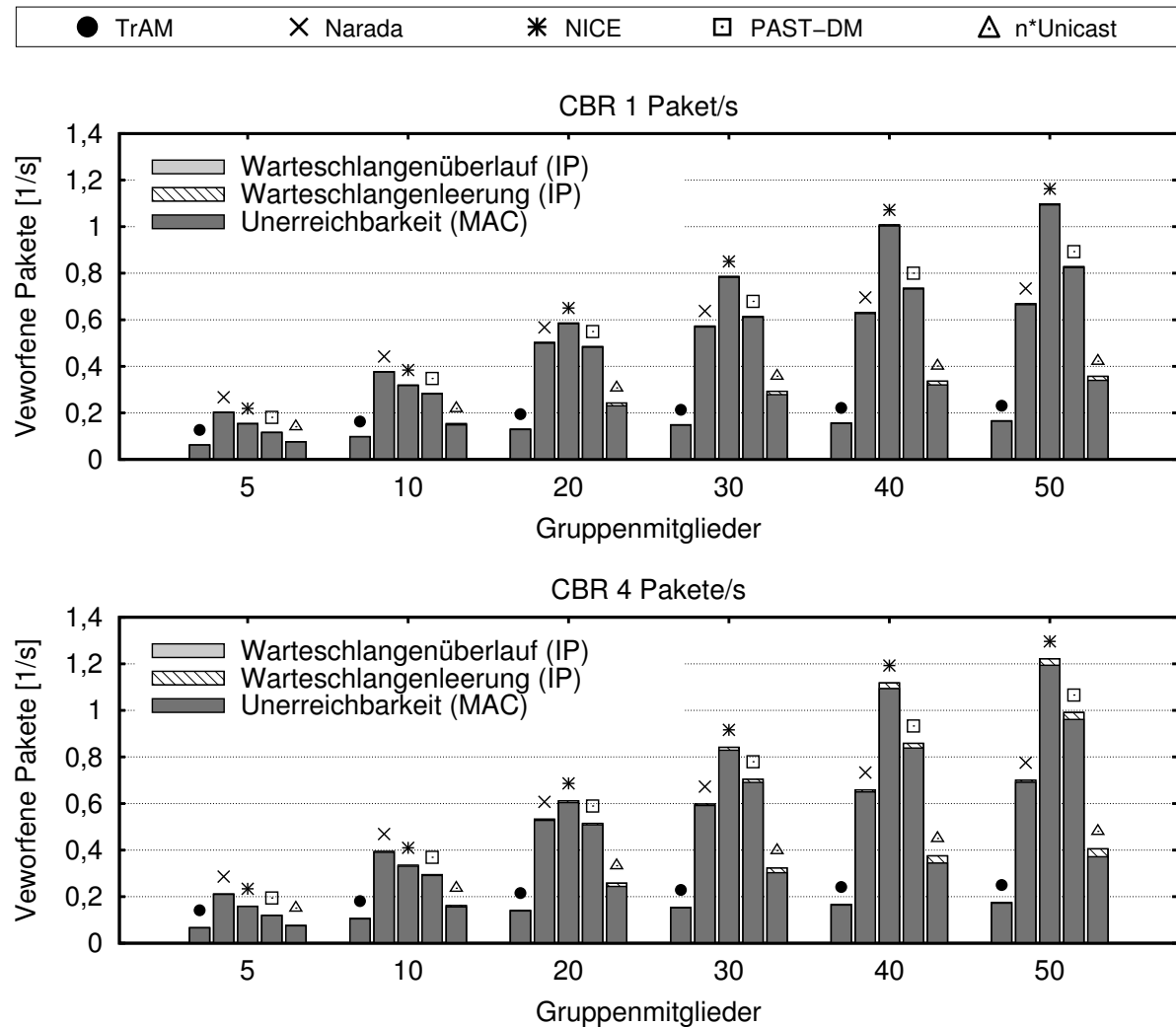


Abbildung 7.19 Pro Simulationssekunde verworfene Pakete bei einer CBR-Anwendung unter Verwendung des LBC-Konzepts.

7.5 Auswertung einer Chat-Anwendung

Nachdem in dem vorangegangenen Abschnitt eine CBR-Anwendung evaluiert wurde, welche in einem zeitlich gleich bleibenden Abstand Pakete identischer Größe versendet, wird in diesem Abschnitt eine Chat-Anwendung untersucht, welche eine Text-basierte Unterhaltung zwischen mehreren Kommunikationspartnern annähert. Im Vergleich zu der CBR-Anwendung ergeben sich bezüglich des Datenverkehrs drei große Unterschiede:

1. Ein stark unterschiedlicher zeitlicher Abstand zwischen Datenpaketen: Dieser ergibt sich aufgrund der Tatsache, dass Kommunikationspartner Textnachrichten unterschiedlich schnell lesen und selbst eingeben, aber auch durch kurzzeitige Ablenkungen von der Unterhaltung.

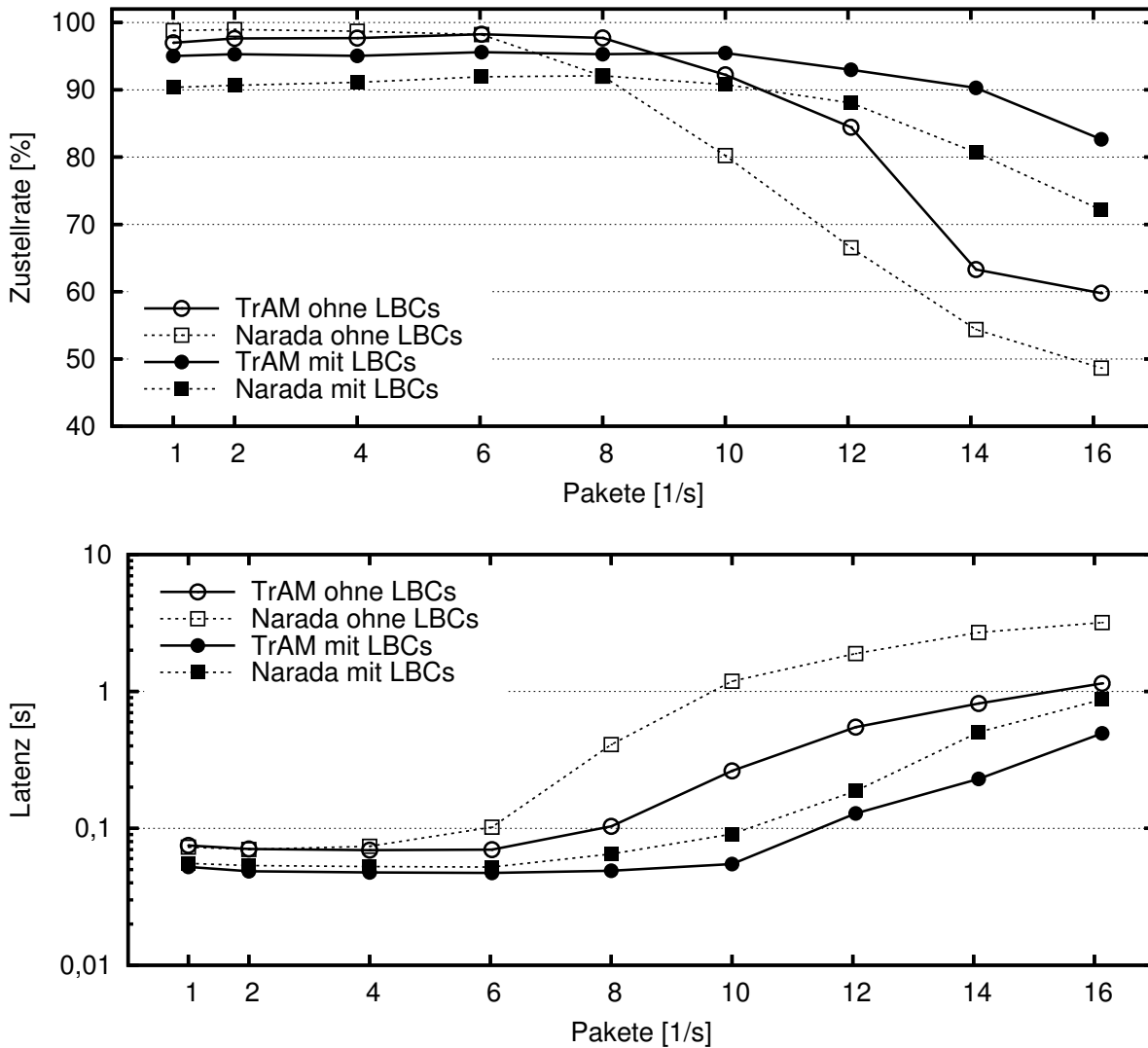


Abbildung 7.20 Von TrAM und Narada bei höheren Datenraten erzielte Zustellraten und Latenzen bei einer Gruppengröße von 30 Gruppenmitgliedern.

2. Eine deutlich schwankende Größe der ausgesendeten Datenpakete: So können von Benutzern eingegebene Textnachrichten entweder sehr kurz ausfallen und nur aus wenigen aber auch aus vielen Wörtern bestehen.
3. Eine mitunter große Anzahl von Multicast-Quellen: In Abhängigkeit der Anzahl der an der Unterhaltung teilnehmenden Kommunikationspartner, kann mit bis zu einigen Dutzend Multicast-Quellen gerechnet werden.

Um im Rahmen dieser Arbeit künstlichen Chat-Verkehr zur Evaluierung der Anwendung zu erzeugen, wurden anhand eines IRC-Clients sich über mehrere Tage erstreckende Messungen in Chat-Räumen durchgeführt. Ausgehend von den aufgezeichneten Konversationen wurde eine Wahrscheinlichkeitsverteilung berechnet, welche Aussagen über

den zeitlichen Abstand zweier aufeinander folgenden Textnachrichten eines gleichen Benutzers zulässt. Die erhaltene Verteilung ist in Abbildung 7.21 dargestellt, wobei die *Pausezeit* eines Benutzers auf der x-Achse und die zugehörige Wahrscheinlichkeit auf der linken y-Achse aufgetragen ist.

Zusätzlich zur Verteilung der Pausezeit wurde eine Korrelation zwischen der Pausezeit und der nach dieser Zeit erwarteten Textlänge erstellt. Diese ist zusammen mit der berechneten Standardabweichung ebenfalls in Abbildung 7.21 eingezeichnet, wobei die Abbildung von der x- auf die rechte y-Achse erfolgt. Ein interessantes Merkmal ist hier die mit der Pausezeit wachsende Standardabweichung der erwarteten Textlänge: Diese lässt sich dadurch erklären, dass bei einer geringen Pausezeit Kommunikationspartner aktiv an einer Unterhaltung teilnehmen und die erzeugte Textlänge maßgebend durch die Lese- und Schreibgeschwindigkeit bestimmt wird. Bei zunehmender Pausezeit sinkt jedoch die Wahrscheinlichkeit einer aktiven Beteiligung an der Unterhaltung: Hier sind Kommunikationspartner zunehmend durch andere Aktivitäten abgelenkt, so dass keine präzisen Aussagen über die erwartete Textlänge getroffen werden können.

Die in diesem Abschnitt durchgeführten Evaluierungen sind erneut für unterschiedliche Gruppengrößen und jeweils in Abhängigkeit der eingesetzten Overlay-Topologie sowie der Verwendung des LBC-Konzepts durchgeführt. Wichtig hierbei ist jedoch, dass nun *jedes* Gruppenmitglied nach seinem Gruppenbeitritt als Multicast-Quelle agiert und Chat-Verkehr gemäß den in Abbildung 7.21 dargestellten Verteilungen erzeugt. Die in diesem Abschnitt durchgeführten Simulationen haben alle eine Dauer von 20 Simulationsminuten: An die Gruppenbeitrittsphase, welche sich über die ersten 5 Simulationsminuten erstreckt, schließt sich eine 15-minütige Datenverteilungsphase an.

7.5.1 Standard P2P-Multicast

In diesem Abschnitt werden zunächst die Ergebnisse der Chat-Anwendung vorgestellt, welche unter Verwendung der Overlay-Topologien ohne den Einsatz lokaler Broadcast-Cluster erzielt wurden. Dabei werden eingangs die gemessenen Zustellraten und Latenzen diskutiert und anschließend die erfasste Belastung des drahtlosen Ad-hoc-Netztes untersucht.

7.5.1.1 Zustellraten und Latenzen

Die beim Betrieb der Chat-Anwendung in Abhängigkeit der Gruppenmitgliedszahl (und somit der Chat-Quellen) gemessenen Zustellraten und Latenzen sind in Abbildung 7.22 aufgetragen. Bei Betrachtung der Zustellraten (linkes Diagramm) ist zu erkennen, dass bis auf das n *Unicast-Protokoll alle Overlay-Topologien ähnlich gute Ergebnisse erzielen: So liegen die erfassten Zustellraten für alle betrachteten Gruppengrößen fast ausnahmslos über 90%. Trotz der bei 5 und 10 Gruppenmitgliedern sehr guten erzielten

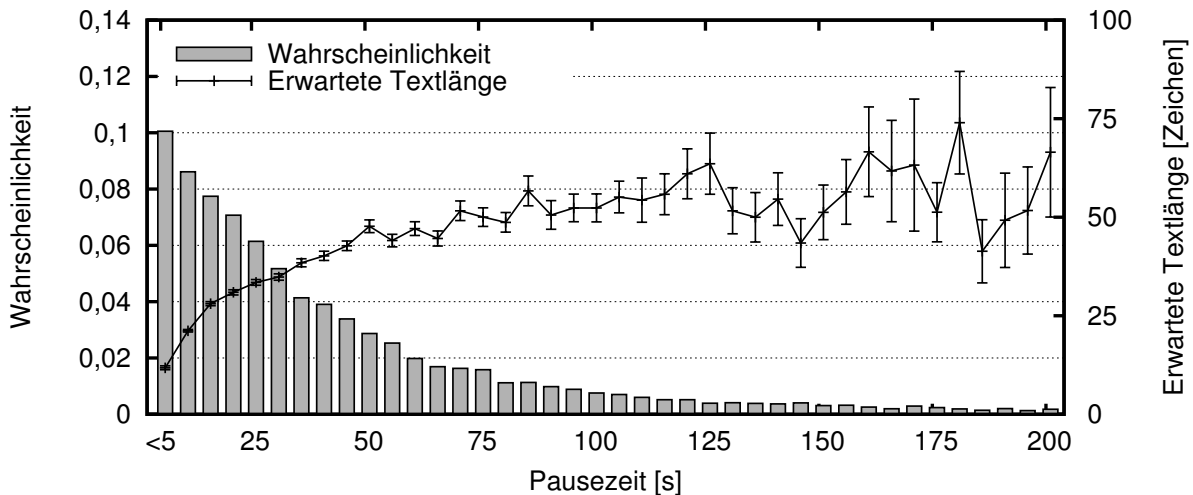


Abbildung 7.21 Wahrscheinlichkeitsverteilungen zur Erzeugung künstlichen Chat-Verkehrs.

Zustellrate ist für das n*Unicast-Protokoll ein deutlicher Leistungseinbruch ab einer Gruppengröße von 30 Gruppenmitgliedern erkennbar. Dieses Verhalten wird an dieser Stelle lediglich angemerkt und erst im Zusammenhang mit der Belastung des drahtlosen Ad-hoc-Netzes näher diskutiert.

Für die bei der Simulation der Chat-Anwendung beobachteten Latenzen (rechtes Diagramm) lässt sich feststellen, dass diese, wie zu erwarten, mit steigender Gruppengröße und Quellenzahl wachsen. Die Tatsache, dass sie für die Protokolle TrAM, Narada und NICE noch im Bereich einiger 10 bis 100 Millisekunden liegen, spricht dafür, dass die Belastung des drahtlosen Ad-hoc-Netzes bei diesen Protokollen noch weitgehend gering ist und somit kaum Verzögerungen durch Kollisionen im Zuge des Medienzugriffs auftreten. Für das Protokoll PAST-DM hingegen ist bereits ein deutlicherer Anstieg zu erkennen, welcher durch die höhere, von diesem Protokoll ausgehende Belastung des drahtlosen Ad-hoc-Netzes bedingt ist. Für das n*Unicast-Protokoll sind bereits bei kleinen Gruppengrößen deutlich höhere Latenzen als bei den übrigen Protokollen zu erkennen, wobei der Unterschied mit steigender Gruppenmitgliedszahl wächst: So können für das n*Unicast-Protokoll ab 30 Gruppenmitgliedern Verzögerungen von mehreren Sekunden festgestellt werden.

7.5.1.2 Belastung des Netzes und Paketverwürfe

Die Ursache für den deutlichen Leistungseinbruch des n*Unicast-Protokolls bei steigender Gruppenmitgliedszahl ist das Vorhandensein einer Vielzahl von Multicast-Quellen. Da durch das n*Unicast-Protokoll eine Duplizierung der Datenpakete lediglich bei der jeweiligen Quelle erfolgt, müssen jeder Quelle beim Versenden eines Datenpakets Routen zu allen übrigen Gruppenmitgliedern bekannt sein. Insbesondere zu Beginn des Daten-

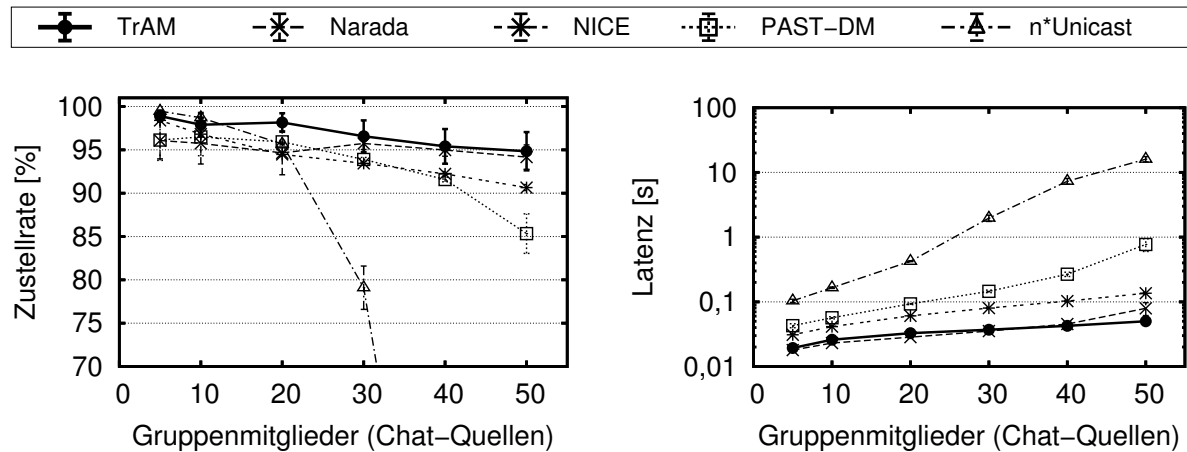


Abbildung 7.22 Zustellraten und Latenzen bei einer Chat-Anwendung.

austauschs oder im Fall größerer zeitlicher Abstände ($> 10s$) von Datenpaketen eines Benutzers ist dies jedoch nicht der Fall: In der Tat zeigt es sich hier als entscheidend, dass AODV für eine bestimmte Dauer ungenutzte Routen verwirft (vgl. Abschnitt 2.1.2.1), so dass bei einer Chat-Anwendung verstärkt Routen aufgebaut werden müssen. Bei einer Gruppengröße von 50 Gruppenmitgliedern hat das dabei erfolgende, wiederholte Fluten des drahtlosen Netzes durch AODV unweigerlich eine Überlastungssituation zur Folge, so dass zahlreiche Routen-Aufbauten fehlschlagen und die weiterzuleitenden Datenpakete zunächst in die Paketwarteschlange eingereiht werden. Das schnelle Ansteigen der pro Simulationssekunde aufgebauten Routen sowie das Kollabieren der Kommunikation im Zusammenhang mit der entstandenen Überlast ist für das n*Unicast-Protokoll in Abbildung 7.23 ab 30 Gruppenmitgliedern in Form eines stark schwankenden Verlaufs der erhaltenen Messungen zu sehen. Untermuert wird diese Aussage durch die in Abbildung 7.24 dargestellten Paketverwürfe. Hier ist zu erkennen, dass für das n*Unicast-Protokoll bereits ab 30 Gruppenmitgliedern mit einem Anstieg von Paketverwürfen aufgrund von Warteschlangenüberläufen gerechnet werden muss. Dies resultiert aus den bereits verdeutlichten, fehlschlagenden Routen-Aufbauten, aufgrund welcher Pakete zunächst in die Paketwarteschlange eingereiht werden müssen. Versendet eine Multicast-Quelle eine weitere Textnachricht bevor Routen bestimmt und die in der Paketwarteschlange verweilenden Pakete bearbeitet und weitergeleitet werden konnten, kommt es insbesondere bei 50 Gruppenmitgliedern zu zahlreichen Warteschlangenüberläufen und den bereits in Abbildung 7.22 gezeigten starken Verzögerungen.

7.5.2 Auswirkungen der Kombination mit LBCs

In diesem Abschnitt wird untersucht, inwieweit die Kombination von Overlay-Topologien mit dem Konzept lokaler Broadcast-Cluster im Kontext einer Chat-Anwendung von Vorteil ist. Hierzu werden zunächst, analog zu dem vorangegangenen Abschnitt,

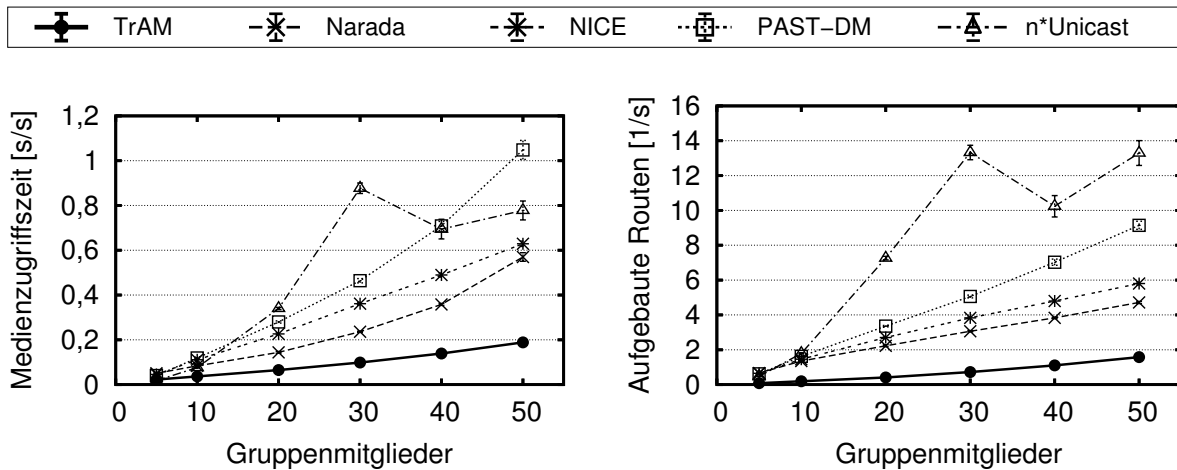


Abbildung 7.23 Pro Simulationssekunde erfolgreiche Mediengriffzeit und Routen-Aufbauten bei einer Chat-Anwendung.

die gemessenen Zustellraten und Latenzen betrachtet und anschließend die gemessene Belastung des drahtlosen Ad-hoc-Netzes untersucht.

7.5.2.1 Zustellraten und Latenzen

Die bei Verwendung des LBC-Konzepts gemessenen Zustellraten und Latenzen sind in Abbildung 7.25 aufgetragen. Dabei ist zunächst festzustellen, dass sich keine Verbesserungen sondern eher Verschlechterungen der Zustellraten (linkes Diagramm) ergeben. Ähnlich zu Abschnitt 7.4.3 liegt der Grund hierfür darin, dass der bei einer Chat-Anwendung ausgesendete Datenverkehr gering genug ist, um durch die Sicherungsmechanismen des Mediengriffprotokolls ein pro- und kein kontraproduktives Verhalten zu erzielen. Besonders zu bemerken sind außerdem einerseits die von TrAM auch bei hohen Gruppenmitgliedszahlen erzielten Ergebnisse: Während die Zustellraten der übrigen Protokolle mit steigender Gruppenmitgliedszahl zurückgehen, kann TrAM ein annähernd konstantes Niveau halten. Dies geht auf die sehr geringe, von TrAM induzierte Netzlast zurück. So ist aufgrund der geringen Netzlast die Wahrscheinlichkeit von Paketkollisionen im Zuge der Datenverteilung anhand von Broadcast-Dateneinheiten weitaus geringer als bei anderen Protokollen. Für diese steigt die Wahrscheinlichkeit des Verlusts von Broadcast-Dateneinheiten aufgrund einer höheren Netzbelastung durch die eigentliche Overlay-Topologie. Andererseits kann im Vergleich zu Abbildung 7.22 bei hohen Gruppenmitgliedszahlen eine deutliche Verbesserung des n*Unicast-Protokolls beobachtet werden. Auf diese wird im Zusammenhang mit der Belastung des drahtlosen Ad-hoc-Netzes näher eingegangen.

Im Hinblick auf die gemessenen Latenzen (rechtes Diagramm) können durch Verwendung des LBC-Konzepts die erwarteten Verbesserungen verzeichnet werden: So können insbesondere bei hohen Gruppenmitgliedszahlen die Latenzen der meisten Protokolle

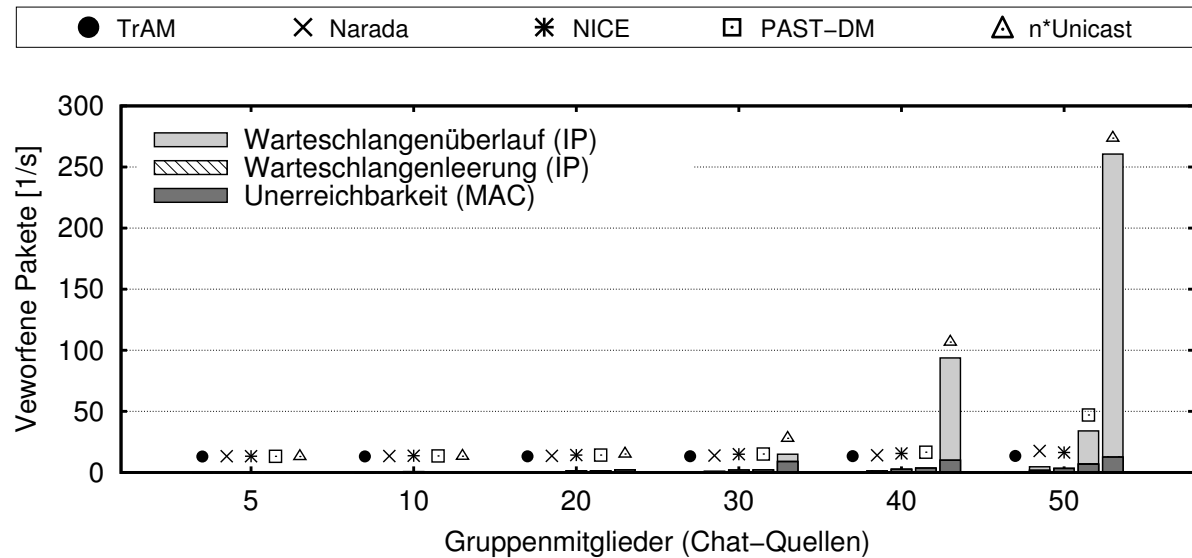


Abbildung 7.24 Pro Simulationssekunde verworfene Pakete bei einer Chat-Anwendung.

auf wenige 10 Millisekunden gedrückt werden. Einzig das n*Unicast-Protokoll weist Latenzen von etwas mehr als 100 Millisekunden auf.

7.5.2.2 Belastung des Netzes und Paketverwürfe

Als Ursache für die deutliche Leistungssteigerung des n*Unicast-Protokolls ist die deutlich geringere Anzahl der in die Overlay-Topologie eingegliederten Gruppenmitglieder zu nennen. Aus diesem Rückgang folgen direkt deutliche Reduktionen der in Abbildung 7.23 noch sehr hohen Anzahl aufgebauter Routen und der damit verbundenen Medienzugriffszeit. Dieser Rückgang ist in Abbildung 7.26 zu sehen.

Aufgrund der deutlich geringeren Belastung des drahtlosen Ad-hoc-Netztes kann in Abbildung 7.27 vor allem für das n*Unicast-Protokoll ein deutlicher Rückgang der wegen Warteschlangenüberläufen verworfenen Pakete verzeichnet werden.

7.6 Auswertung von Mehrbenutzerspielen

In diesem Abschnitt soll untersucht werden, inwieweit Echtzeit-basierte Mehrbenutzerspiele anhand des in dieser Arbeit vorgeschlagenen P2P-basierten Gruppenkommunikationsdienstes unterstützt werden können. Die drei besonderen Charakteristiken, welche für diesen Anwendungstyp aus Sicht eines Kommunikationsprotokolls berücksichtigt werden müssen, sind:

1. Der Datenstrom, welcher sich aus einer Vielzahl sehr kleiner Datenpakete zusammensetzt [46, 64]: Der Datenstrom, der von jedem Mitspieler ausgesendet wird,

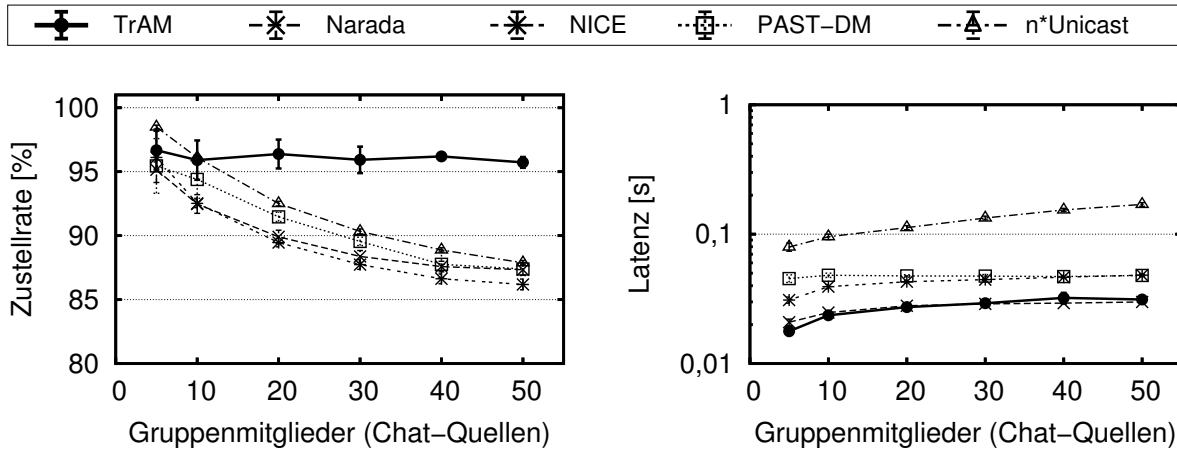


Abbildung 7.25 Zustellraten und Latenzen bei einer Chat-Anwendung unter Verwendung des LBC-Konzepts.

besteht aus so genannten *Positionen*, welche 25 mal pro Sekunde versendet werden und die aktuelle Position und Geschwindigkeit des Spielers in einem virtuellen Raum kodieren. Darüber hinaus werden zusätzliche Pakete versendet, in welchen gesonderte *Ereignisse* kodiert werden, wie etwa das Abfeuern einer Waffe oder das Aufsammeln eines Objekts.

2. Die Zuverlässigkeitsanforderungen: Beim Versenden von Positionen können *einzelne* Paketverluste durchaus toleriert werden, da anhand der in ihnen enthaltenen Information (Position sowie Bewegungsgeschwindigkeit und -richtung) die Bewegungen von Mitspielern zumindest für kurze *Zeit vorhergesagt* werden können. Im Gegensatz dazu sind Ereignisse wiederum *zuverlässig* zu übertragen, da sie spielentscheidende Aktionen kodieren.
3. Die Echtzeit-Anforderungen: Die ausgesendeten Datenpakete sind schnell an die übrigen Mitspieler zu verteilen, da nur dann ein akzeptabler Spielfluss entsteht und ein angenehmes Spielerlebnis gewährleistet werden kann [20, 44, 73, 80]. Während Latenzen von weniger als 100 Millisekunden wünschenswert sind, wird ab ca. 150 bis 200 Millisekunden Verzögerung kein überzeugender Spielfluss mehr wahrgenommen.

Auch wenn das hier verwendete Medienzugriffsprotokoll IEEE 802.11 für die Unterstützung von Anwendungen mit Echtzeit-Anforderungen nur bedingt geeignet ist [56], zeigen sich Evaluationen aufgrund des sehr starken Verbreitungsgrades dieses Medienzugriffsprotokolls dennoch als interessant.

Im Zusammenhang mit dieser Arbeit wurden unterschiedliche Untersuchungen zur Unterstützung von Mehrbenutzerspielen durch P2P-basierte Gruppenkommunikationspro-

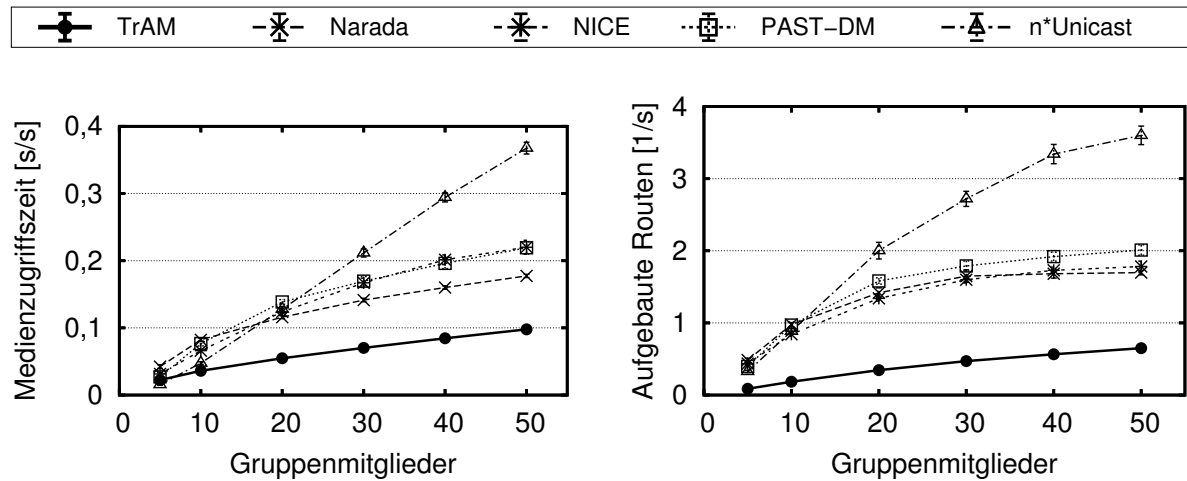


Abbildung 7.26 Pro Simulationssekunde erfolgreiche Medienzugriffszeit und Routenaufbauten bei einer Chat-Anwendung unter Verwendung des LBC-Konzepts.

tokolle getätigt. So hat sich in [55] gezeigt, dass die hohe Anzahl der pro Mitspieler ausgesendeten Datenpakete sowie deren Weiterleitung an die übrigen Mitspieler im Kontext des Zugriffs auf das drahtlose Medium als kritisch zu sehen ist und die Anzahl der unterstützbaren Mitspieler begrenzt. In dieser Arbeit sollen daher lediglich Ergebnisse diskutiert werden, welche unter Verwendung des in Anhang A.2 dokumentierten *Multi-Player-Gaming-Transport-Protokolls (MPGT-Protokolls)* erzielt wurden: Dieses Transportprotokoll leistet nicht nur den unzuverlässigen Transport von Positionen und den zuverlässigen Transport von Ereignissen sondern erzielt zusätzlich eine deutliche Reduktion der Anzahl der erforderlichen Medienzugriffe durch Aggregation von Datenpaketen. Darüber hinaus soll in den folgenden Untersuchungen die Anzahl der zur Weiterleitung von Datenpaketen erforderlichen Medienzugriffe weiter reduziert werden, indem durchgehend das LBC-Konzept zum Einsatz kommt. Modelliert wird der ausgesendete Verkehr, indem jeder Mitspieler seiner MPGT-Protokollinstanz pro Simulationssekunde 25 Pakete mit zweidimensionalen Positionen und bis zu 4 in zufälligem Abstand erzeugte Pakete mit Ereignissen übergibt. Während Positionen eine feste Größe von 9 Bytes haben, besitzen Ereignisse eine zufällig zwischen 16 und 32 Bytes gewählte Größe.

7.6.1 Eine Single-Hop-Umgebung

Aufgrund der harten Anforderungen von Mehrbenutzerspielen an die Datenverteilung, werden in dieser Arbeit lediglich einfache Szenarien untersucht: Es wird zunächst ein Single-Hop-Szenario betrachtet, welches ein klassisches “LAN-Party-Szenario” widerspiegelt. Hier treffen sich Mitspieler in einer bestimmten Räumlichkeit, um gegeneinander anzutreten. Zur Untersuchung des Szenarios werden Endgeräte innerhalb des Simulati-

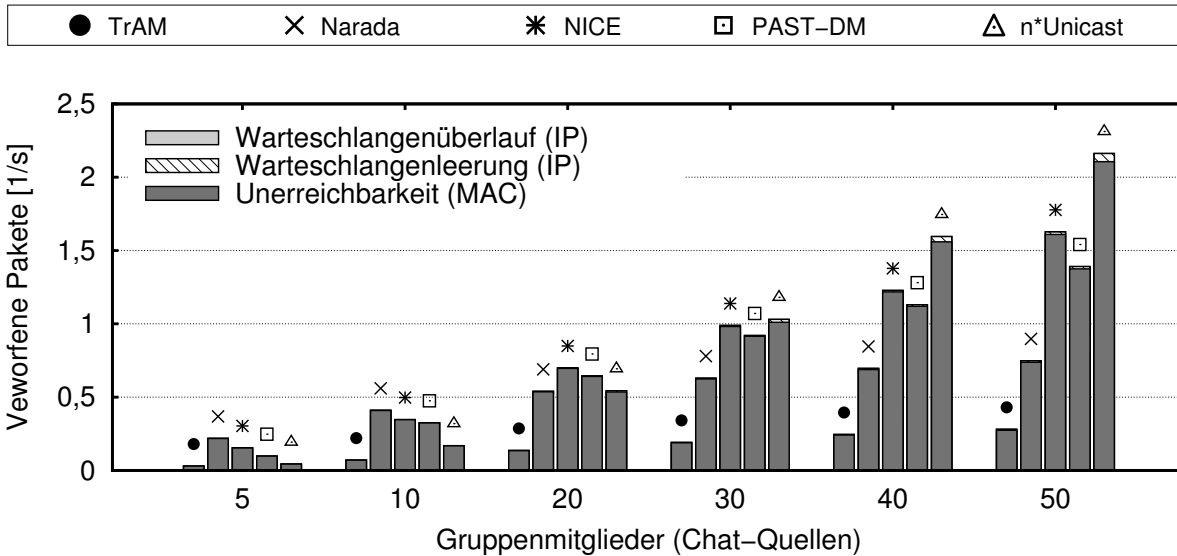


Abbildung 7.27 Pro Simulationssekunde verworfene Pakete bei einer Chat-Anwendung unter Verwendung des LBC-Konzepts.

onsareals in jeweils 3 Metern Entfernung zueinander Matrix-förmig platziert, wobei jede Zeile der Matrix bis zu 5 Endgeräte enthält. Bei Gründung der Multicast-Gruppe bildet sich zunächst ein LBC-Repräsentant aus, so dass die übrigen Endgeräte einen lokalen Gruppenbeitritt bewerkstelligen können. Da in einem solchen Szenario demnach keine Overlay-Verbindungen ausgebildet werden, spielt die eingesetzte Overlay-Topologie bei diesen Untersuchungen keine Rolle.

In Abbildung 7.28 sind zunächst für einzelne Spielerzahlen (2, 8 und 22 Spieler) die Wahrscheinlichkeiten von beobachteten Latenzen für Positionen und Ereignisse sowie für Positionszwischenankunftszeiten aufgetragen.

Im Hinblick auf die Positionen kann festgestellt werden, dass diese bei 2 Spielern (oberes Diagramm) eine Latenz von weniger als 10 Millisekunden aufweisen. Dies liegt darin begründet, dass unter Verwendung des MPGT-Protokolls die eigenen, von einem Spieler ausgesendeten Positionen sofort versendet und mit den eventuell von anderen Spielern weiterzuleitenden Positionen aggregiert werden. Da bei 2 Spielern keine Positionen zusätzlicher Spieler weiterzuleiten sind, umfassen die Aggregate folglich immer nur die eigenen Positionen, so dass sie die geringen, beobachteten Latenzen aufweisen. Bei 8 Spielern (mittleres Diagramm) hingegen sind bereits Latenzen der Positionen von bis zu 40 Millisekunden erkennbar. In diesem Szenario senden die lokal beigetretenen Gruppenmitglieder ihre eigenen Positionen dem LBC-Repräsentanten zu. In diesem verzögert das MPGT-Protokoll die Broadcast-Weiterleitung der empfangenen Positionen innerhalb des LBCs bis zu dem Versenden der eigenen Position, also um maximal 40 Millisekunden. Bei 22 Spielern (unteres Szenario) ist zu erkennen, dass Positionen teil-

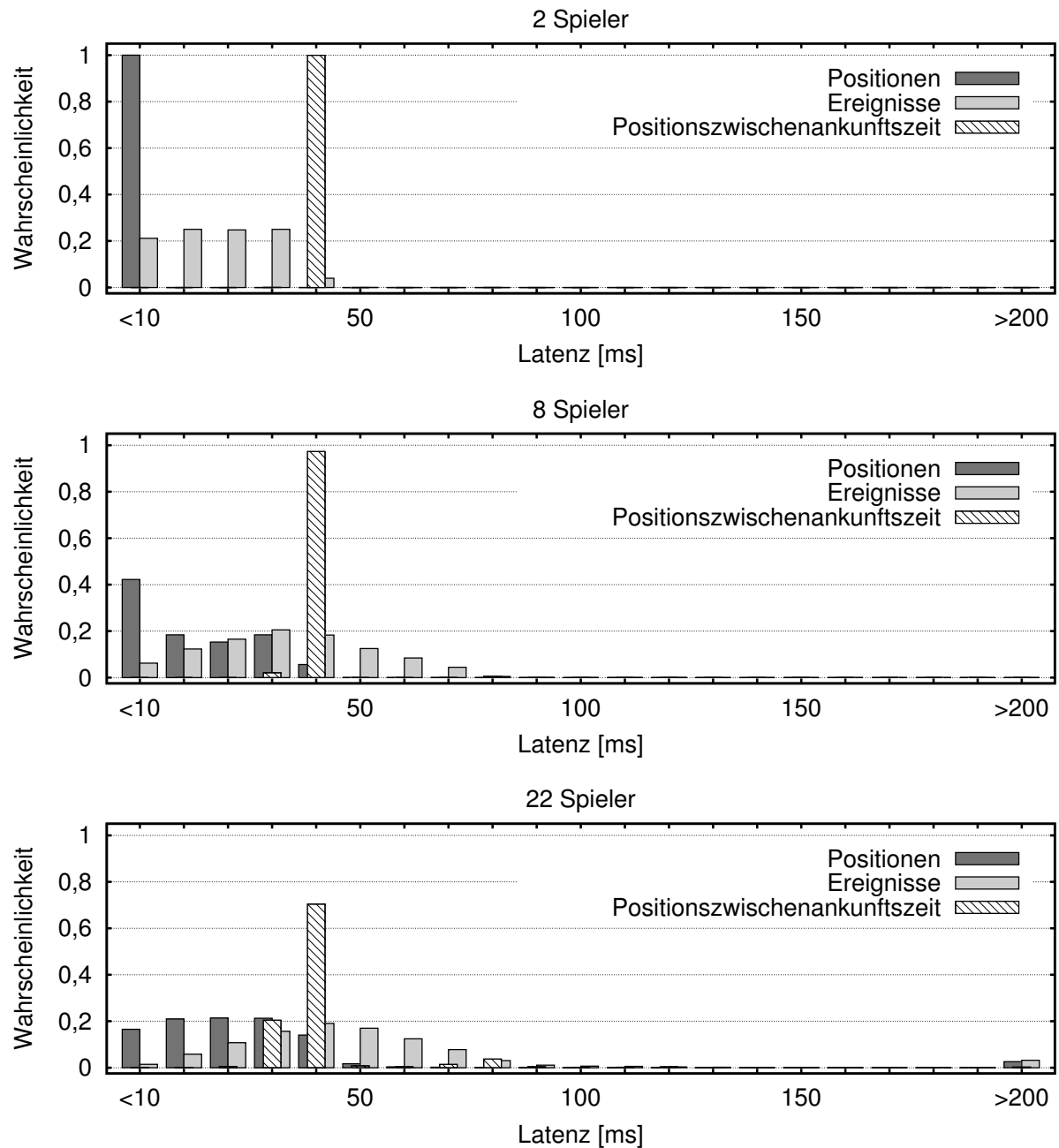


Abbildung 7.28 Latenzen für Positionsinformation und Ereignisse, welche in einer Single-Hop-Umgebung beobachtet werden können.

weise auch um mehr als 40 Millisekunden verzögert werden. Dies ist wiederum auf die bereits sehr hohe Belastung des drahtlosen Mediums zurückzuführen, aufgrund welcher zunehmend Kollisionen beim Mediengriff entstehen. So ist insbesondere erkennbar, dass ein gewisser Anteil der Positionen eine Verzögerung von mehr als 200 Millisekunden aufweist: Das drahtlose Ad-hoc-Netz stößt bei dieser Spielerzahl an seine Grenzen und der Spielfluss wird nicht mehr als angenehm empfunden.

Da Ereignisse nicht in einem festen Intervall sondern “spontan” erzeugt und von dem MPGT-Protokoll bis zum Versenden der nächsten Position verzögert werden, weisen sie bereits bei 2 Spielern eine entsprechende Verteilung der Latenz mit bis zu 40 Millisekunden auf. Bei zusätzlichen Spielern nimmt die Latenz weiter zu, da das MPGT-Protokoll die Ereignisse nicht nur im aussendenden Spieler sondern auch im weiterleitenden LBC-Repräsentanten bis zum Versenden dessen nächster Position verzögert. Dennoch sind bei beispielsweise 8 Spielern die Latenzen auf 70 Millisekunden beschränkt, so dass ein guter Spielfluss wahrgenommen werden kann. Bei 22 Spielern wiederum weist aufgrund einer zu hohen Netzlast ein nicht mehr vernachlässigbarer Anteil der Ereignisse eine Verzögerung von mehr als 200 Millisekunden auf, so dass die Mitspieler keinen angenehmen Spielfluss mehr wahrnehmen.

Die dargestellten Positionszwischenankunftszeiten geben Aufschluss darüber, in welchem zeitlichen Abstand ein Mitspieler Positionen der übrigen Mitspieler empfängt. Während dies bei 2 und 8 Mitspielern aufgrund einer geringen Netzlast alle 40 Millisekunden geschieht, sind bei 22 Spielern auch Zwischenankunftszeiten von 80 Millisekunden zu erkennen. Diese entstehen durch den Verlust einzelner Broadcast-Dateneinheiten, in welchen der LBC-Repräsentant aggregierte Positionsinformation der Mitspieler innerhalb seines LBCs weiterleitet.

In Abbildung 7.29 sind Messergebnisse für zusätzliche Spielerzahlen aufgetragen. Die einzelnen Kurven geben dabei an, welcher Anteil der Positionen und Ereignisse eine bestimmte Latenz besitzen. Während die Punktierung der Kurven drei Latenzklassen definiert ($< 40\text{ms}$, $< 100\text{ms}$ und $> 150\text{ms}$) gibt die Füllung der Messpunkte Auskunft darüber, ob es sich um Positionen (schwarze Füllung) oder Ereignisse (keine Füllung) handelt. Wie für Abbildung 7.28 bereits erläutert wurde, ist zu erkennen, dass Ereignisse aufgrund ihres zufälligen Auftretens sowie ihrer zusätzlichen Verzögerung bei der Weiterleitung höhere Latenzen als Positionen aufweisen. So nimmt der Anteil der Ereignisse, welche eine Latenz von weniger als 40 Millisekunden besitzen mit steigender Spielerzahl zügig ab. Dennoch bleibt bis ca. 20 Spieler ein guter Spielfluss erhalten: So sind erst für 21 Spieler einige wenige Ereignisse und bei 22 Spielern schließlich nahezu 5% der Ereignisse um mehr als 150 Millisekunden verzögert. Dies kann als Anzeichen dafür gesehen werden, dass bereits zu viele Medienzugriffe stattfinden und somit Kollisionen die Weiterleitung von Datenpaketen beeinträchtigen. Insbesondere kann ab 22 Spielern auch eine Vergrößerung der Konfidenzintervalle erkannt werden, welche auf die steigende Netzlast zurückzuführen ist.

7.6.2 Eine Multi-Hop-Umgebung

Um die Anzahl der Medienzugriffe innerhalb des LBCs zu reduzieren und so zusätzliche Spieler unterstützen zu können, soll in diesem Abschnitt untersucht werden, inwieweit

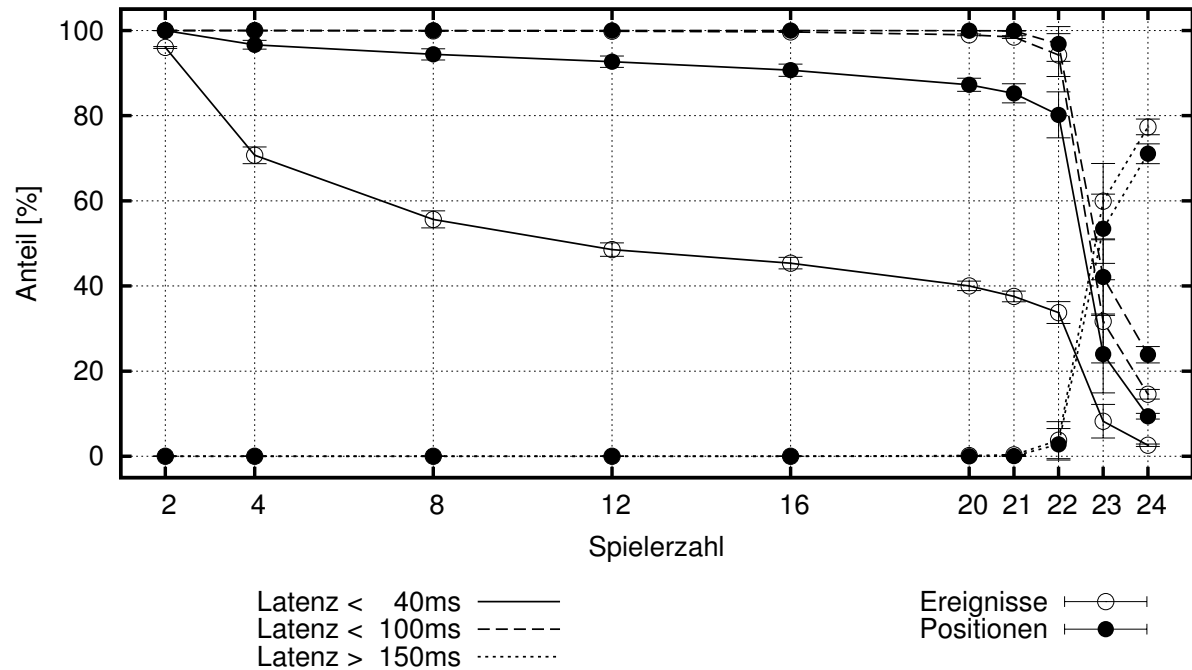


Abbildung 7.29 Abhängigkeit der Latenzen für Positionen und Ereignisse von der Mitspielerzahl in einer Single-Hop-Umgebung.

eine räumliche Aufteilung der Spieler von Vorteil ist. Die grundlegende Idee hierbei ist, dass zwei LBCs räumlich weit genug voneinander entfernt sind, um unter Berücksichtigung des Erkennungsbereichs von Funksignalen ihre Kommunikation gegenseitig nicht zu beeinflussen. Der Austausch aggregierter Spielerinformation *zwischen* den LBCs soll dann wiederum anhand einer Multi-Hop-Unicast-Verbindung erfolgen, welche von einer Overlay-Topologie aufgebaut und für die Datenvermittlung eingesetzt wird. Insbesondere soll in diesem Abschnitt also auch untersucht werden, inwieweit sich die Kommunikation entlang von Overlay-Verbindungen und innerhalb von LBCs gegenseitig beeinflussen und beeinträchtigen.

Zur Modellierung eines solchen Szenarios werden, wie in Abbildung 7.30 aufgezeigt, zwei LBC-Repräsentanten (Spieler 1 und 2) in einer Entfernung von $450m$ voneinander²⁰ platziert. Darüber hinaus werden in Abhängigkeit der Spielerzahl zusätzliche Endgeräte abwechselnd in den LBCs positioniert. Zur Überbrückung des Abstands zwischen den LBCs werden zwei gruppenfremde Endgeräte (A und B) in jeweils $150m$ Entfernung zwischen den LBC-Repräsentanten platziert: Diese Endgeräte nehmen an der Gruppenkommunikation nicht teil, sondern dienen lediglich der Weiterleitung von Datenpaketen zwischen den LBCs. Zur Verbindung der beiden LBCs wird das TrAM-Protokoll verwendet, wobei sich die zwischen den LBCs aufgebaute Overlay-Verbindung aufgrund der

²⁰Zur Erinnerung: Aufgrund der Konfigurationsparameter aus Tabelle 7.1 beträgt der Erkennungsbereich von Funksignalen $353,8m$.

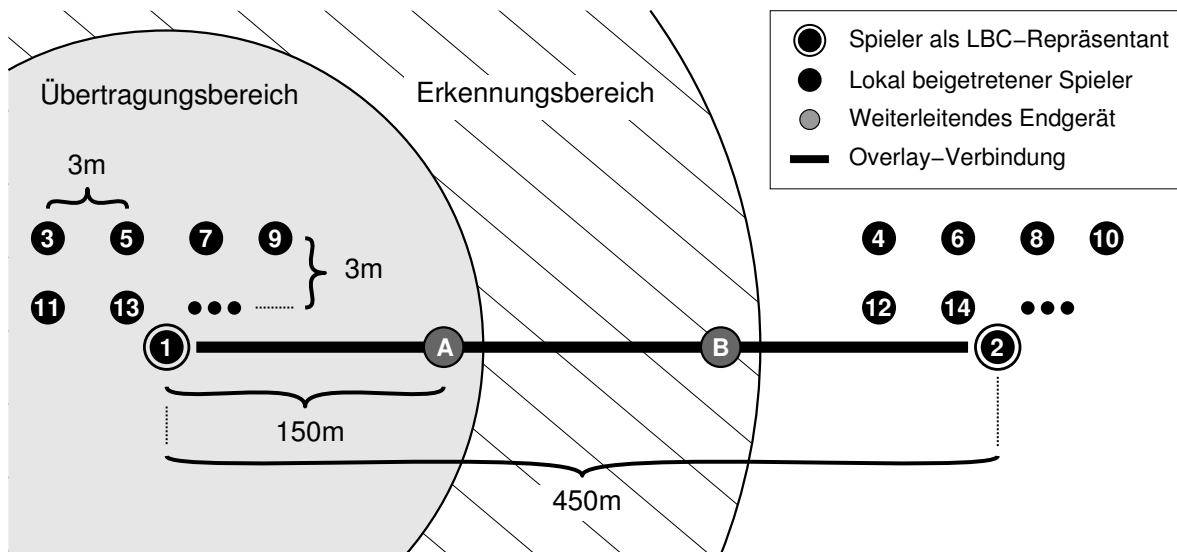


Abbildung 7.30 Szenario zur Modellierung eines räumlich verteilten Mehrbenutzerspiels.

Entfernung als 3-Hop-Verbindung ergibt. In diesem Zusammenhang ist zu bemerken, dass durch Mobilität bedingte Verbindungsabbrüche aufgrund der in Abschnitt 2.1.3 diskutierten Erkennungsdauer erhebliche Auswirkungen auf beobachtete Latenzen haben. Aus diesem Grund soll daher das gesamte Szenario als statisch bzw. der Einsatz eines Routing-Protokolls angenommen werden, welches Verbindungsabbrüche frühzeitig erkennt und durch zügiges Umschalten auf Ersatz-Routen Verzögerungen entsprechend reduziert [36, 65, 105].

In Abbildung 7.31 sind die in diesem Szenario gemessenen Ergebnisse dargestellt. Wider Erwarten ergibt sich durch die räumliche Aufteilung der Endgeräte keine Verbesserung sondern eine deutliche Verschlechterung der unterstützbaren Spielerzahl. So steigt der Anteil an Ereignissen, welche mit mehr als 150 Millisekunden Latenz zugestellt werden, bereits ab 11 Spielern an. Als Ursache hierfür kann die Tatsache genannt werden, dass die Weiterleitung der aggregierten Dateneinheiten entlang der zwischen den LBCs aufgebauten Overlay-Verbindung durch die Kommunikation innerhalb der LBCs erheblich beeinträchtigt wird. In der Tat befinden sich die zu Weiterleitungszwecken herangezogenen Endgeräte A und B zwischen den LBCs jeweils innerhalb des Erkennungsbereich der Funksignale, welche innerhalb der LBCs ausgesendet werden. Wird etwa von Spieler 1 ein aggregiertes Datenpaket über die bestehende Overlay-Verbindung an Spieler 2 gesendet, so kann Spieler 1 die Mediengriffe innerhalb des rechten LBCs nicht feststellen, obwohl diese dennoch die Übertragung zwischen 1 nach A beeinträchtigen²¹. So zeigen sich bereits innerhalb des ersten Hop der Overlay-Verbindungen Verzögerungen

²¹Dieser Sachverhalt wurde in Abschnitt 2.1.1.4 als Problematik versteckter Endgeräte eingeführt.

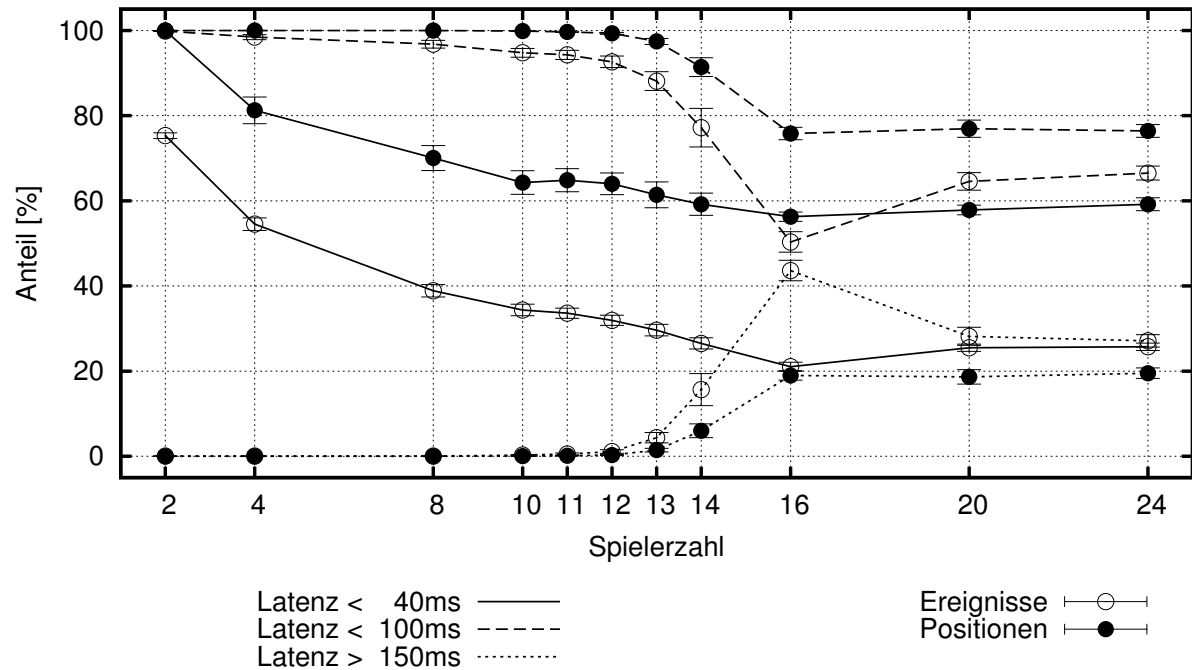


Abbildung 7.31 Abhängigkeit der Latenzen für Positionen und Ereignisse von der Mitspielerzahl in einer Multi-Hop-Umgebung.

beim Medienzugriff, wobei sich diese Verzögerungen über die einzelnen Overlay-Hops weiter aufsummieren.

Von der Verwendung der RTS/CTS-Erweiterung ist in diesem Zusammenhang keine Verbesserung zu erwarten: Dies liegt darin begründet, dass ein etwaiges von A versendetes CTS-Paket innerhalb des rechten LBCs zwar erkannt aber aufgrund zu hoher Entfernung nicht rekonstruiert werden kann. Somit ist damit zu rechnen, dass aufgrund der zusätzlich stattfindenden Medienzugriffe eine weitere Reduktion der Anzahl unterstützbarer Spieler erfolgt. Dieser Sachverhalt kann in Abbildung 7.32 beobachtet werden, in welcher zu erkennen ist, dass der Anteil der Ereignisse mit einer Latenz von mehr als 150 Millisekunden bereits bei 9 Spielern ansteigt.

Da der Grund für den Leistungseinbruch in der großen Ausdehnung des Erkennungsbereichs der Funksignale zu suchen ist, erscheint die Reduktion der Sendeleistung innerhalb von LBCs ein erfolgsversprechender Ansatz: So kann diese für lokal beigetretene Gruppenmitglieder derart gesetzt werden, dass die Übertragungreichweite lediglich bis zu dem räumlich sehr nahe gelegenen LBC-Repräsentanten reicht. Da dieser in dem betrachteten Szenario nur wenige 10 Meter entfernt ist, ergibt sich durch die Drosselung der Sendeleistung eine deutliche Verkleinerung des Erkennungsbereichs. Die Kommunikation von lokal beigetretenen Gruppenmitgliedern wirkt sich folglich nicht mehr auf die Weiterleitung der aggregierten Datenpakete über die Overlay-Verbindung aus.

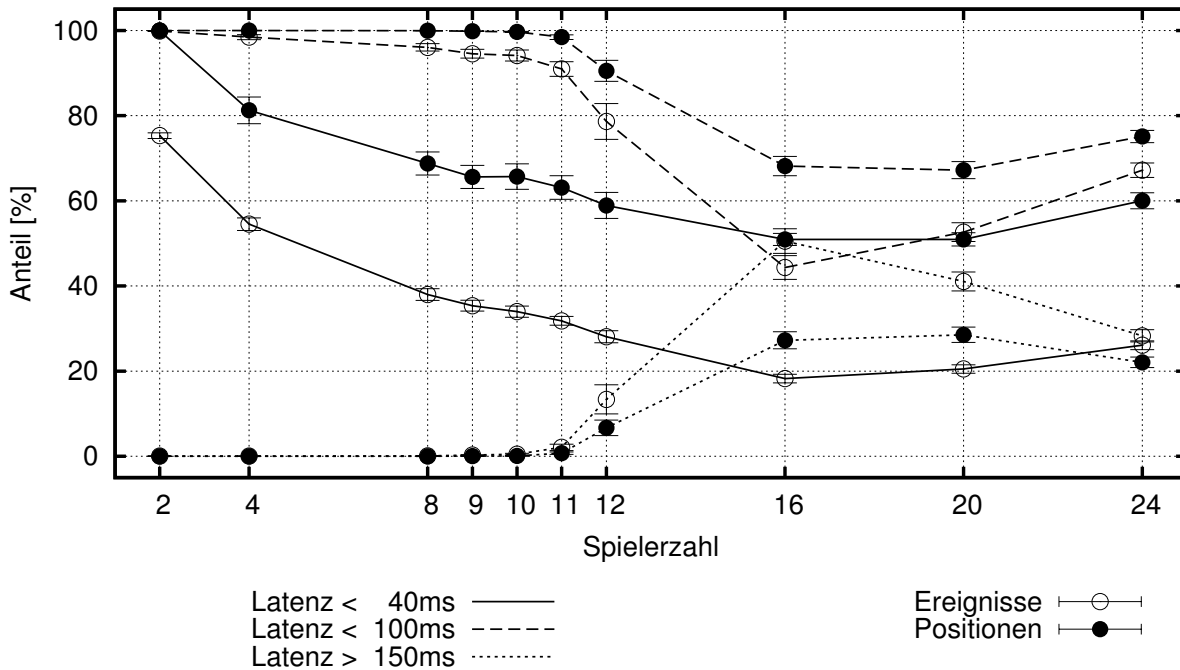


Abbildung 7.32 Latenzen für Positionsinformation und Ereignisse in einer Multi-Hop-Umgebung bei Verwendung der RTS/CTS-Erweiterung.

Die Ergebnisse, welche durch Drosselung der Sendeleistung von lokal beigetretenen Gruppenmitgliedern erreicht werden können, sind in Abbildung 7.33 dargestellt, wobei die RTS/CTS-Erweiterung wieder deaktiviert wurde. Im Vergleich zu Abbildung 7.31 können deutliche Verbesserungen erkannt werden: So erreicht bei Drosselung der Sendeleistung der Anteil an Ereignissen mit einer Latenz jenseits von 150 Millisekunden erst bei mehr als 16 Spielern nennenswerte Ausmaße. Trotz der Verbesserung bleiben die Ergebnisse von Single-Hop-Umgebungen jedoch unerreichbar. Dies ist unter anderem auf die Tatsache zurückzuführen, dass die Kommunikation innerhalb eines LBCs versteckt vor der Übertragung eines Datenpakets über den letzten Übertragungsabschnitt der Overlay-Verbindung ist: Wünscht etwa B ein aggregiertes Datenpaket an 2 weiterzuleiten, so nimmt B die Kommunikation innerhalb des rechten LBCs wegen zu geringen Sendeleistungen nicht wahr. Folglich greift B auf das Medium zu und verursacht hierbei jedoch verstärkt Kollisionen.

Aufgrund der in diesem einfachen Multi-Hop-Szenario beobachteten Ergebnisse kann gefolgert werden, dass eine noch stärkere räumliche Verteilung der Endgeräte weitere Einbußen hinsichtlich der unterstützbaren Spielerzahl nach sich ziehen wird. Wie anhand der in diesem Abschnitt durchgeführten Untersuchungen gezeigt werden konnte, sind als Gründe für den erheblichen Leistungsrückgang einerseits die Schwierigkeiten bei einem verteilten Zugriff auf das drahtlose Medium zu nennen. Andererseits erscheint das im Rahmen dieser Arbeit eingesetzte Medienzugriffsprotokoll aufgrund der fehlenden

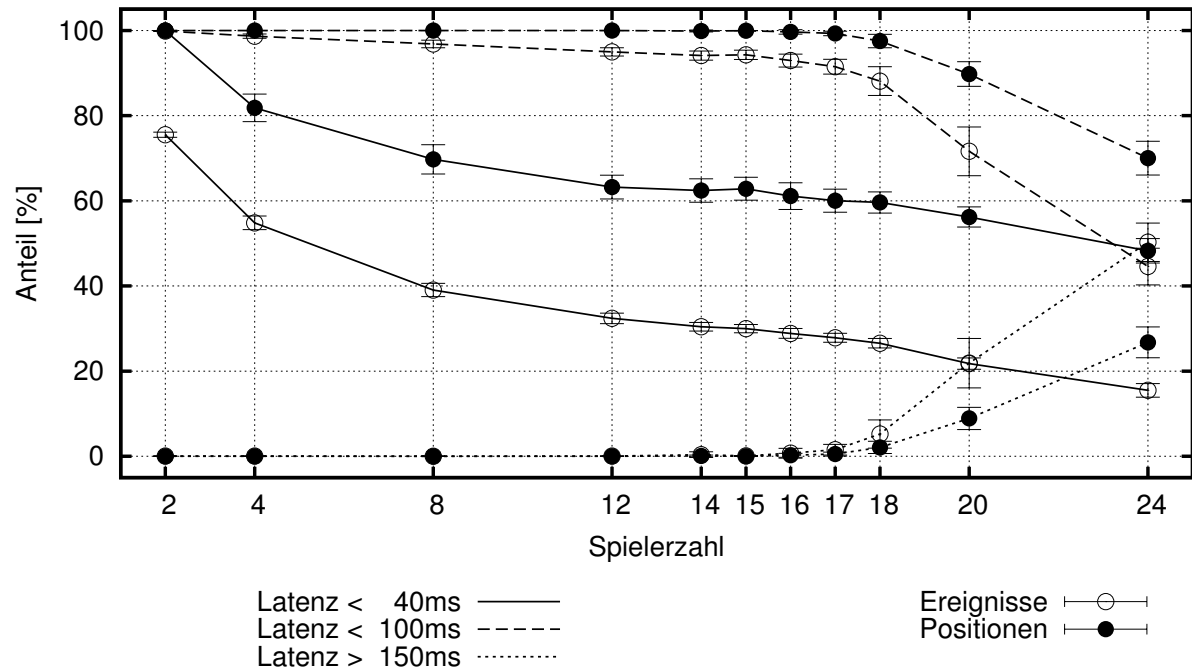


Abbildung 7.33 Latenzen für Positionen und Ereignisse in einer Multi-Hop-Umgebung bei Anpassung der Sendeleistung.

Dienstgüteunterstützungen weniger geeignet: Obgleich sich für die eingangs untersuchte Single-Hop-Umgebung bereits interessante Ergebnisse erzielen lassen, erweist sich die Berücksichtigung und Befriedigung der Echtzeit-Anforderungen im Zuge der Datenvermittlung bei einer räumlichen Verteilung der Mitspieler als unerlässlich.

8. Zusammenfassung und Ausblick

Aufgrund ihrer vorteilhaften Eigenschaften, wie ihre Selbstorganisation oder ihre Unabhängigkeit von fester Infrastruktur, sind drahtlose Ad-hoc-Netze in den letzten Jahren zu einem wichtigen Forschungsbereich in der Mobilkommunikation geworden. Dabei hat sich insbesondere die Konzeption von *Gruppenkommunikationsprotokollen* zu einem zentralen Aspekt entwickelt, da allgemein anerkannt ist, dass drahtlose Ad-hoc-Netze vor allem in Szenarien zum Einsatz kommen, in denen ein kooperatives Verhalten von Benutzern mit gleichen Interessen herrscht. Derartige Szenarien umfassen beispielsweise militärische oder Katastropheneinsätze aber auch Anwendungen für verteiltes Lernen oder Messe- und Touristenführungen.

Die Konzeption von Gruppenkommunikationsprotokollen für drahtlose Ad-hoc-Netze erweist sich als Herausforderung, da die Mobilität der Endgeräte und der Wettbewerb um die geringe Bandbreite des drahtlosen Mediums geeignet zu berücksichtigen sind. Doch auch die Heterogenität der Anforderungen einzelner Anwendungen an die Datenübertragung erschweren die Integration von Gruppenkommunikationsprotokollen: Unterschiede zeigen sich hier sowohl im Hinblick auf die zur Weiterleitung von Datenpaketen aufgebaute Verteilstruktur als auch die geforderte Zuverlässigkeit.

P2P-basierte Gruppenkommunikationsprotokolle sind relativ junge Protokollkonzepte, welche den prinzipiellen Vorteil bieten, dass sie mit geringem Aufwand in Netze integrierbar sind. Dies ergibt sich aufgrund der Tatsache, dass bei ihnen die Gruppenmitglieder selbst die Verteilstruktur, das so genannte *Overlay-Netz*, *untereinander* ausbilden und sich darüber ihre Datenpakete anhand existierender Punkt-zu-Punkt-Kommunikationsprotokolle zusenden. Da folglich nur die Gruppenmitglieder das Gruppenkommunikationsprotokoll beherrschen müssen, kann sowohl das zur Datenverteilung aufgebaute

Overlay-Netz als auch die darüber umgesetzte Zuverlässigkeit derart konzipiert werden, dass sie die Anforderungen der betriebenen Anwendung bestmöglich erfüllen.

8.1 Ergebnisse dieser Arbeit

In dieser Arbeit wurde ein *P2P-basierter Gruppenkommunikationsdienst* für drahtlose Ad-hoc-Netze vorgestellt. Um die initial in Abbildung 1.3 aufgezeigten Anforderungen zu erfüllen, basiert der Dienst auf der in dieser Arbeit entwickelten *MAAM*-Architektur. Indem diese den Dienst in einzelne Komponenten zerlegt, wie etwa eine Overlay- und eine Zuverlässigkeitskomponente, lassen sich für bestimmte Anwendungen konzipierte Protokolle flexibel in die Gruppenkommunikation integrieren. Entgegen bisheriger Arbeiten erfüllt der vorgestellte Ansatz somit die geforderte Unabhängigkeit von dem Szenario und der betriebenen Anwendung.

Um den effizienten Betrieb der Gruppenkommunikation zu gewährleisten, wurde in dieser Arbeit ein neues Overlay-Protokoll konzipiert: Das TrAM-Protokoll. Das von TrAM aufgebaute Overlay-Netz genügt wie kein anderes Overlay-Netz den in dieser Arbeit für ein effizientes Ablaufen der Gruppenkommunikation als essentiell erkannten Anforderungen. Dazu zählen ein möglichst geringer Kontrollverkehr zum Aufbau des Overlay-Netzes und eine hohe Flexibilität der Overlay-Topologie zur Reaktion auf die Mobilität der Endgeräte innerhalb des drahtlosen Ad-hoc-Netzes. Um dies zu erreichen, verzichtet das TrAM-Protokoll auf Redundanz innerhalb der Overlay-Topologie und verbindet die Gruppenmitglieder anhand einer flexiblen Baumstruktur. Während TrAMs Kontrollverkehr aus wenigen, kleinen Paketen besteht, werden diese durch Nutzung der Semi-Broadcast-Eigenschaft des drahtlosen Mediums zwischen den Gruppenmitgliedern vermittelt, so dass auch der von reaktiven Routing-Protokollen für Routen-Aufbauten erforderte Aufwand reduziert wird.

Da aus Anwendungssicht Broadcast-Übertragungen fehleranfälliger als Unicast-Übertragungen sind, erscheint ihr Einsatz nicht für jede Anwendung geeignet, so dass sie kein fester Bestandteil der Datenvermittlung eines Overlay-Netzes sein sollten. In dieser Arbeit wurde daher getrennt das LBC-Konzept entwickelt, welches einen generischen Charakter aufweist und mit *beliebigen Overlay-Netzen* kombinierbar ist. Es vermeidet den Aufbau von Overlay-Verbindungen zwischen nahe gelegenen Gruppenmitgliedern, indem es diese aus dem Overlay ausschließt. Zu diesen Gruppenmitgliedern erfolgt die Datenvermittlung anhand von Broadcast- und nicht anhand von Unicast-Übertragungen. Durch einen sorgfältigen Schnittstellenentwurf zu der Overlay-Komponente wurde das LBC-Konzept in Form einer generischen und optionalen Protokollkomponente in die MAAM-Architektur und somit in den Gruppenkommunikationsdienst integriert.

Die in dieser Arbeit durchgeführten Evaluierungen belegen nicht nur, dass im Vergleich zu bisherigen Arbeiten das TrAM-Protokoll für den Aufbau und die Verwaltung seiner

Overlay-Topologie eine in der Regel um den Faktor 3 geringere Belastung des drahtlosen Mediums verursacht. Sie zeigen darüber hinaus, dass, im Hinblick auf die Datenverteilung von CBR- und Chat-Anwendungen, vergleichbare und sogar stark überlegene Ergebnisse erzielt werden können. Dies ergibt sich direkt aus der von TrAM bei der Verwaltung der Overlay-Topologie eingesparten Bandbreite. Da diese zur Vermittlung zusätzlicher Nutzdaten verwendet werden kann, gestaltet sich die Datenübertragung bei TrAM deutlich effizienter als bei allen anderen Protokollen. Ebenso ist TrAM aufgrund seiner effizienten Overlay-Verwaltung in der Lage, größere Gruppen als andere Protokolle zu unterstützen und dabei eine leistungsfähigere Datenverteilung zu bieten.

Dennoch ist zu bemerken, dass keines der in dieser Arbeit diskutierten Overlay-Netze mit der Gruppengröße skaliert. Dies liegt in der Natur von P2P-basierten Gruppenkommunikationsprotokollen begründet, welche Datenpakete anhand von Unicast-Übertragungen den einzelnen Gruppenmitgliedern zustellen: So steigt mit der Gruppengröße die Anzahl der erforderlichen Unicast-Übertragungen und somit die für die Weiterleitung eines Datenpakets benötigte Bandbreite. Die gewonnenen Messungen belegen, dass aufgrund ihres erhöhten Verwaltungsaufwands Protokolle wie PAST-DM oder NICE bereits ab 40 Gruppenmitgliedern oftmals an ihre Grenzen stoßen.

Durch Erweiterung der Overlay-Netze mit dem generischen LBC-Konzept kann ihre Skalierbarkeit deutlich verbessert werden. So kann eine Reduktion der Mediengriffszeit um den Faktor 2 beobachtet werden, wobei die Verbesserung mit steigender Gruppengröße zunimmt. Dies ergibt sich aus dem vorgenommenen Ausschluss räumlich naher Gruppenmitglieder aus dem Overlay-Netz. Während hieraus bereits ein geringerer Overlay-Verwaltungsaufwand und somit eine Entlastung des drahtlosen Mediums folgt, wird diese Entlastung durch die Datenvermittlung mittels Broadcast-Übertragungen weiter verstärkt. So wächst bei Verwendung des LBC-Konzepts der zur Gruppenverwaltung und Datenverteilung benötigte Aufwand nicht mehr mit der Gruppengröße sondern mit dem durch die Gruppenmitglieder besetzten Areal. Die Evaluierungen belegen jedoch auch, dass sich das LBC-Konzept nicht im Kontext jeder Anwendung gewinnbringend zeigt: Insbesondere wurde für schmalbandige Chat-Anwendungen gezeigt, dass sich die im Vergleich zu Unicast-Übertragungen fehleranfälligeren Broadcast-Übertragungen negativ auswirken. Der letztendliche Einsatz des LBC-Konzepts hängt demnach nicht nur von dem Anwendungsszenario sondern auch von der darin anvisierten Gruppengröße ab. Dieses Ergebnis bestätigt einerseits die Forderung nach einer flexibel konfigurierbaren Gruppenkommunikation und untermauert andererseits den in dieser Arbeit hervorgebrachten Entwurf des LBC-Konzepts als *optionalen* Bestandteil einer Architektur zur modularen Komposition der Gruppenkommunikation.

Der in dieser Arbeit entwickelte P2P-basierte Gruppenkommunikationsdienst ist aufgrund des modularen Charakters der MAAM-Architektur nicht nur flexibel hinsicht-

lich der Integration verschiedener Overlay-Netze und Umsetzungen von Zuverlässigkeit. Durch die Kapselung System-naher Funktionen ermöglicht die MAAM-Architektur zusätzlich den Betrieb des Gruppenkommunikationsdienstes im Kontext unterschiedlicher Plattformen: Während die in dieser Arbeit durchgeführten Evaluierungen durch Integration der Architektur in eine Netzwerksimulationsumgebung gewonnen wurden, konnte die Architektur ebenfalls auf Linux- und Windows-Systeme portiert werden. In dieser Arbeit konnten daher auch prototypische Anwendungen auf Basis der MAAM-Architektur, des TrAM-Protokolls und des LBC-Konzepts implementiert und auf renommierten Konferenzen vorgeführt werden.

8.2 Ausblick und weiterführende Arbeiten

Architekturen für Kommunikationsprotokolle haben für lange Zeit einen vergleichsweise starren und unflexiblen Aufbau verfolgt. Im Zusammenhang mit der seit einigen Jahren immer schneller voranschreitenden Weiterentwicklung des Internets zur Unterstützung neuartiger Anwendungen und Dienste führen derartige Architekturkonzepte zwangsläufig zu Sackgassen, da neu aufkommende Anforderungen nur schwer berücksichtigt werden können. Forschungsinitiativen rund um das Internet der Zukunft, das *Future Internet*, bemühen sich daher um einen grundlegend neuen Entwurf einer Protokollarchitektur für das Internet. Ein wesentliches Entwurfsziel dieser Architektur ist ihre Flexibilität sowie ihr Potenzial zur Integration neuer Dienste und den damit verbundenen Anforderungen.

Im Kontext einer flexibel komponierbaren, modularen Internet-Protokollarchitektur erscheint auch der in dieser Arbeit konzipierte P2P-basierte Gruppenkommunikationsdienst sowie die dafür maßgebend herangezogene MAAM-Architektur unter einem anderen Licht. Sind im Rahmen dieser Arbeit lediglich Protokollkonzepte für drahtlose Ad-hoc-Netze unterbreitet und in die MAAM-Architektur integriert worden, erscheint ihr Einsatz im Internet oder in Infrastruktur-basierten Drahtlosnetzen sowie die Integration speziell für diese Netze konzipierter Protokollkomponenten ein interessanter Anknüpfungspunkt für weiterführende Arbeiten. Insbesondere kann jedoch auch der aufgrund von Mobilität erfolgende Übergang von einem Netztyp zu einem anderen Netztyp ein betrachtenswerter Prozess sein: Durch den *dynamischen Austausch* einzelner Protokollkomponenten innerhalb der MAAM-Architektur könnten im Falle eines detektierten Netz-Übergangs die Anforderungen einer betriebenen Anwendung durch Berücksichtigung der jeweiligen Netz-Charakteristiken bestmöglich unterstützt werden.

Ogleich in dieser Arbeit bereits erste Schritte zur Nutzung des konzipierten Gruppenkommunikationsdienstes durch reale Anwendungen unternommen wurden, weisen diese bislang lediglich einen prototypischen Charakter auf. Weiterführende Arbeiten können sich in diesem Zusammenhang mit der *Entwicklung komplexerer Gruppenkommunikati-*

onsanwendungen auf Basis der MAAM-Architektur befassen. Derartige Entwicklungen würden dann auch die Konzeption *Anwendungs-spezifischer Zuverlässigkeitsumsetzungen* umfassen. Ebenso erweist sich insbesondere in einem drahtlosen Kommunikationsumfeld die *Integration von Sicherheitsmechanismen* als kritisch, um eine Robustheit gegenüber potenziellen Angreifern zu erreichen.

A. Zusätzliche Protokolle und Mechanismen

In diesem Anhang werden zusätzliche Protokolle und Mechanismen, die in dieser Arbeit entwickelt wurden und im Zusammenhang mit der Evaluation in Kapitel 7 eingesetzt werden, kurz vorgestellt. Dabei wird lediglich das Verständnis für die Funktionsweise der Protokolle geweckt, so dass keine präzisen Spezifikationen vorgenommen werden.

A.1 Das n*Unicast-Protokoll

Das n*Unicast-Protokoll bietet eine sehr einfache Umsetzung einer Peer-to-Peer-basierenden Gruppenkommunikation. Es hebt sich durch seine vereinfachte Paketduplizierung und -weiterleitung von anderen Peer-to-Peer-basierten Gruppenkommunikationsprotokollen ab: So besteht die Grundidee des n*Unicast-Protokolls darin, dass allein die jeweilige Quelle eines Multicast-Pakets die Duplicierung des Pakets vornimmt. Die einzelnen Duplikate werden den Empfängern über Unicast-Dateneinheiten zugestellt, welche ausgehend von der Multicast-Quelle durch das Routing-Protokoll über potenziell mehrere Übertragungsabschnitte bis zu ihrem Ziel geleitet werden. Es folgt eine Beschreibung des Gruppenverwaltung sowie eine kurze Diskussion des Protokolls. Bemerkenswert soll an dieser Stelle noch, dass das n*Unicast-Protokoll in dieser Arbeit lediglich herangezogen wird, um die Auswirkungen der besonderen *Paketduplizierung* mit der von aufwendigeren Overlay-Topologien vergleichen zu können. Fortgeschrittene Protokollmechanismen, wie beispielsweise eine Ausfalldetektion von Gruppenmitgliedern, sind für das n*Unicast-Protokoll in dieser Arbeit daher ebensowenig von Interesse wie genaue Paketstrukturen für den Gruppenbeitritt oder die Gruppenverwaltung.

A.1.1 Gruppenverwaltung

Bei dem n*Unicast-Protokoll erhält das gruppengründende Endgerät die besondere Aufgabe der Gruppenmitgliedsverwaltung: Es ist einerseits dafür zuständig, einem neu beitretenden Gruppenmitglied einen Einstiegspunkt in die Gruppe zu bieten. Andererseits ist es dafür verantwortlich, die bereits beigetretenen Gruppenmitglieder mit aktuellen Listen aller Gruppenmitglieder zu versorgen, so dass im Falle eines zu versendenden Datenpakets von der Paketquelle eine korrekte Duplizierung und Weiterleitung vorgenommen werden kann.

Wünscht demnach ein Endgerät der Gruppe beizutreten, sucht es nach dem Gruppengründer anhand von wachsenden Ringsuchen (vgl. Abschnitt 3.1.1). Ist der Gruppengründer gefunden, erfolgt bei diesem eine Registrierung als neues Gruppenmitglied. Zur Bestätigung des Beitritts sendet der Gruppengründer dem neu beitretenden Gruppenmitglied eine aktuelle Liste aller derzeit beigetretenen Gruppenmitglieder zu. Nach dem Versenden der Registrierungsanfrage wartet das neu beitretende Gruppenmitglied bis zu 2 Sekunden auf das Eintreffen der durch den Gruppengründer versendeten Gruppenmitgliedsliste. Wurde die Liste nach dieser Zeitspanne nicht empfangen, geht das beitretende Gruppenmitglied von einem Paketverlust aus und sendet die Anfrage erneut. Dies wiederholt sich bis entweder die Liste empfangen wird und der Gruppenbeitritt hierdurch abgeschlossen ist oder bis nach maximal 5 Übertragungswiederholungen der Registrierung der Gruppenbeitritt aufgegeben wird.

Nach seinem Beitritt erfolgt von jedem Gruppenmitglied alle 15 Sekunden periodisch eine Auffrischung seiner Registrierung bei dem Gruppengründer. Zur Bestätigung der Registrierung sendet auch hier der Gruppengründer eine eventuell aktualisierte Liste aller Gruppenmitglieder dem sich registrierenden Gruppenmitglied zu.

A.1.2 Diskussion

Durch seine sehr einfache Gruppenverwaltung bietet das n*Unicast-Protokoll einige interessante Aspekte. Zu diesen zählt beispielsweise der sehr geringe Verwaltungsaufwand: So bezieht jedes Gruppenmitglied von einem dedizierten Gruppengründer periodisch eine Liste der aktuell beigetretenen Gruppenmitglieder, an welche ein zu sendendes Datenpaket weitergeleitet wird. Der Aufwand zur Verwaltung der Overlay-Topologie wird demnach deutlich reduziert, und der Aufwand zum Auf- und Abbau sowie zur Bewertung von Overlay-Verbindungen entfällt sogar gänzlich. Darüber hinaus ergibt sich die in Abschnitt 3.1.5 besprochene Problematik des konsistenten Routing als bedeutungslos. Dies resultiert aus den Tatsachen, dass einerseits das n*Unicast-Protokoll keine Topologieadaptionen vornimmt, und Topologieänderungen lediglich im Fall von Gruppenbeitritten erfolgen. Andererseits findet die Duplizierung von Datenpaketen nicht auf Basis

von zwischen den Gruppenmitgliedern *verteilter* Information statt sondern auf Basis der Gruppenmitgliedsliste, welche dem sendenden Gruppenmitglied selbst vorliegt.

Eben aufgrund dieser Paketduplizierung und -weiterleitung sind mit dem n*Unicast-Protokoll jedoch auch einige Nachteile verbunden. So wird beim Versenden von Datenpaketen mit steigender Gruppenmitgliedszahl das drahtlose Ad-hoc-Netz in der Umgebung der Multicast-Quelle zunehmend belastet. Dies resultiert aus der Tatsache, dass die Duplizierung eines Datenpakets, anstatt wie von anderen Overlay-Topologien auf die einzelnen Gruppenmitglieder verteilt zu werden, lediglich durch die jeweilige Quelle vorgenommen wird. Insbesondere bei Verwendung von reaktiven Routing-Protokollen kann dies zur Konsequenz haben, dass zur Weiterleitung eines Datenpakets von der Quelle ausgehend zunächst eine Route zu jedem Empfänger aufgebaut werden muss. Dies hat folglich nicht nur eine deutlichere Belastung des drahtlosen Ad-hoc-Netzes zur Folge sondern ebenfalls eine stärkere Verzögerung der weiterzuleitenden Paketduplikate.

A.2 Das MPGT-Protokoll

Bei dem *Multi-Player-Gaming-Transport-Protokoll* (*MPGT-Protokoll*) handelt es sich um ein in dieser Arbeit entwickeltes Transportprotokoll, welches speziell zur Unterstützung von Peer-to-Peer-basierten Echtzeit-Mehrbenutzerspielen in drahtlosen Ad-hoc-Netzen konzipiert wurde. Um den in Abschnitt 7.6 beschriebenen Anforderungen dieser Anwendungen gerecht zu werden, verfolgt das MPGT-Protokoll drei Ziele:

1. *Ein unzuverlässiger Transport von Positionsinformation*: Da jeder Mitspieler seine aktuelle Positionsinformation periodisch 25 mal pro Sekunde (also alle 40 Millisekunden) versendet, erscheinen Übertragungswiederholungen von als verloren erkannter Positionsinformation als nicht gerechtfertigt. In der Tat ist die Zeitspanne, welche zur Erkennung und Neuübertragung verlorener Information benötigt wird, oftmals größer als die Zeitspanne bis zum Versenden der nächsten, ohnehin aktuelleren Positionsinformation.
2. *Ein zuverlässiger Transport von Ereignissen*: Ereignisse, welche von Spielern generiert werden, kodieren oftmals Spiel-entscheidende Aktionen und werden durch das MPGT-Protokoll im Gegensatz zu Positionsinformation daher zuverlässig an die übrigen Mitspieler versendet.
3. *Eine Aggregations-basierte Reduktion von Medienzugriffen*: Um die Anzahl der Medienzugriffe und somit das Kollisionsrisiko im Laufe des Medienzugriffs zu reduzieren, aggregiert das MPGT-Protokoll Datenpakete mit gleicher Zieladresse.

Im Folgenden wird kurz auf die entscheidende Funktionalität des MPGT-Protokolls eingegangen, ohne dabei jedoch detaillierte Protokollspezifikationen vorzunehmen.

A.2.1 Die Aggregation von Datenpaketen

Zur Reduktion von Medienzugriffen aggregiert das MPGT-Protokoll die von den Spielern ausgesendeten Positionen und Ereignisse. Dabei erfolgt nicht nur die Aggregation der *eigens gesendeten* sondern auch der *von anderen Mitspielern weitergeleiteten* Daten.

Aus Sicht eines Spielers A verzögert die auf A laufende MPGT-Protokollinstanz die von A erzeugten Ereignisse, indem diese zunächst in einem Puffer eingefügt werden. Die Verzögerung erfolgt bis zum Versenden der nächsten Positionsinformation von A, also maximal 40 Millisekunden: In diesem Fall werden alle seit der letzten Positionsinformation gepufferten Ereignisse sowie die neu versendete Positionsinformation zu einem Datenpaket aggregiert. Um die Pufferung und Aggregation vornehmen zu können, muss das MPGT-Protokoll folglich in der Lage sein, die von der Anwendung ausgesendeten Ereignisse und Positionsinformation unterscheiden zu können. Um zusätzliche Medienzugriffe einzusparen, wird die von A versendete Information zusätzlich mit der von anderen Mitspielern weiterzuleitenden Information aggregiert. Hierzu werden *selbst empfangene* Positionsinformationen und Ereignisse analog zu As eigenen Ereignissen gepuffert und im Falle des Versendens eigener Positionsinformation aggregiert.

Bei der Weiterleitung von Information ist ein wichtiger Aspekt die *Zieladressen-spezifische Aggregation* von Information. So nimmt A die Aggregation derart vor, dass die von einem Mitspieler B erhaltene Information nicht an B zurückgeleitet wird. Dies ist vor allem dann von Vorteil, wenn A von B selbst wiederum für evtl. mehrere Mitspieler aggregierte Datenpakete erhält.

Aufgrund des nach Zieladressen aggregierenden Vorgehens lässt sich das MPGT-Protokoll auch in Kombination mit dem in Abschnitt 5.1.1 vorgestellten LBC-Konzept verwenden, indem die Paketaggregation über die Broadcast-Adresse erfolgt. Hierbei lässt sich jedoch nicht vermeiden, dass die von einem lokalen Gruppenmitglied L empfangene Information bei der Weiterleitung an die übrigen Mitglieder des LBCs auch an L zurückgesendet wird. Das MPGT-Protokoll nimmt deshalb eine Filterung vor, indem es im Falle des Empfangs selbst ausgesendeter Information diese nicht an die Anwendung übergibt sondern verwirft.

A.2.2 Einbettung von Verzögerungsinformation

Aufgrund der bei jeder Weiterleitung potenziell durchgeführten Verzögerung von bis zu 40 Millisekunden, können maximal 3 Weiterleitungen erfolgen, ohne dass die kritische Maximallatenz von 150 Millisekunden überschritten wird. Um die Effekte der Verzögerung zu reduzieren, versieht das MPGT-Protokoll aggregierte Ereignisse und Positionsinformation jeweils mit der Dauer, für welche das Datum vor seiner Weiterleitung gepuffert wurde. Im Zuge der Weiterleitung wird diese eingebettete Verzögerungsinformation im Anschluss an jede Pufferung um die jeweilige Pufferungsdauer erhöht.

Wichtig hierbei ist, dass das MPGT-Protokoll keine Messungen von Paketumlaufzeiten zwischen Mitspielern vornimmt. Die in Datenpaketen eingebettete Verzögerungsinformation umfasst daher lediglich die *Pufferungsdauern in Mitspielern* nicht aber die *Übertragungsdauer zwischen Mitspielern*. Dies ergibt sich aufgrund der in Abschnitt 3.2 dargestellten Problematik, aufgrund welcher Paketumlaufzeiten in drahtlosen Ad-hoc-Netzen von Paket zu Paket sehr starke Unterschiede aufweisen können und in dieser Arbeit im Rahmen des MPGT-Protokolls daher nicht zum Einsatz kommen.

Eine empfangene Verzögerungsinformation wird der Anwendung übergeben, so dass diese gegebenenfalls eine Extrapolation der empfangenen Spielzustände vornehmen kann: Damit ist jener Vorgang gemeint, bei welcher die Anwendung auf Basis der empfangenen Verzögerungs- und Positionsinformation die Bewegung eines Mitspielers vorhersagt, um eventuelle, durch die Datenübertragung entstandene Verzögerungen auszugleichen [3]. Dies ist wiederum nur dann möglich, wenn die Positionsinformation ebenfalls Geschwindigkeitsinformation des jeweiligen Mitspielers umfasst.

A.2.3 Zuverlässigkeit

Das MPGT-Protokoll versieht sowohl versendete Positionsinformation als auch Ereignisse mit getrennten, aufsteigenden Sequenznummern. Während die Übertragung von Positionsinformation unzuverlässig erfolgt, sind Sequenznummern dennoch erforderlich, um eine Reihenfolgetreue der Positionsinformation zu garantieren: So gilt es insbesondere zu vermeiden, dass eine aufgrund von Verzögerungen verspätet empfangene Positionsinformation im Anschluss an eine bereits aktuellere Positionsinformation der Anwendung übergeben wird.

Da Ereignisse im Gegensatz zu Positionsinformation zuverlässig zu übertragen sind, erweist sich nicht nur der Einsatz von Sequenznummern sondern auch von Quittungen und Übertragungswiederholungen als notwendig. Versendet ein Spieler A ein aggregiertes Datenpaket an ein Spieler B, so aggregiert er in diesem Datenpaket daher kumulierte, positive Quittungen für alle Mitspieler, von welchen A Information von B empfängt. Bei Erhalt des Datenpakets kann B anhand der Quittungen feststellen, welche Ereignisse für welche Mitspieler gegebenenfalls neu zu übertragen sind. Derartige Übertragungswiederholungen werden wiederum gepuffert und erst im Falle des Versendens einer eigenen Positionsinformation von B an A in gewohnter Weise aggregiert.

A.3 Bewertung von LBC-Repräsentanten

Das in Abschnitt 5.1.1 eingeführte LBC-Konzept sieht vor, dass lokal beigetretene Gruppenmitglieder durch den Empfang bzw. das Ausbleiben von Heartbeat-Paketen eine Bewertung von umliegenden LBC-Repräsentanten vornehmen. Diese Bewertung wird in

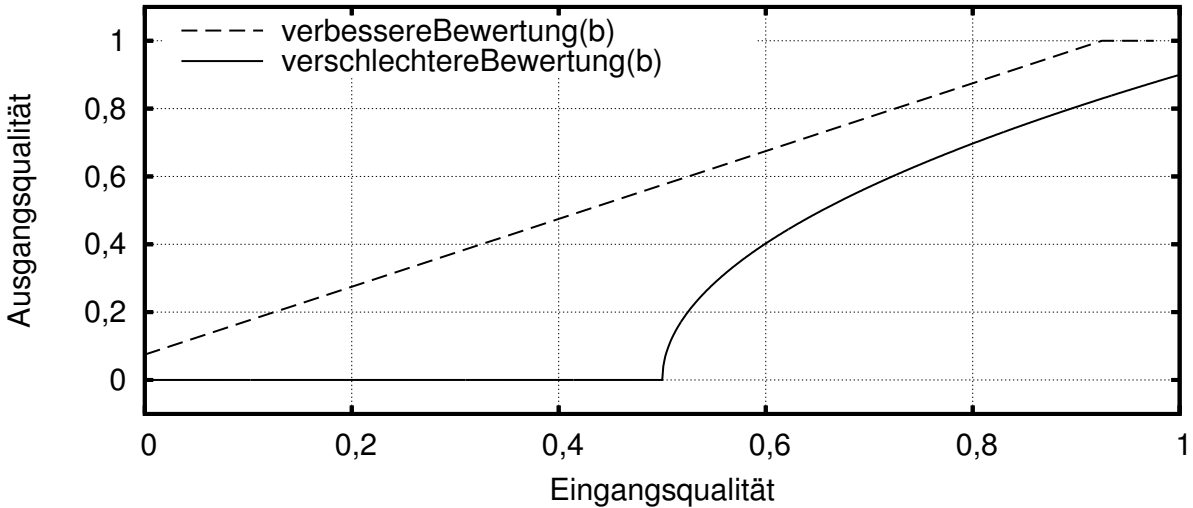


Abbildung A.1 Funktionen zur Bewertung von LBC-Repräsentanten.

dieser Arbeit aufgrund von zwei Formeln *verbessereBewertung(b)* und *verschlechtereBewertung(b)* vollzogen, welchen im Falle des Empfangs bzw. des Verlusts eines Heartbeat die bisherige Bewertung b des zugehörigen Repräsentanten als Parameter übergeben wird. Die Formeln selbst sind durch

$$\begin{aligned}
 \text{verbessereBewertung}(b) &:= \begin{cases} b + 0,075 & , \text{ falls } b + 0,075 < 1, \\ 1 & , \text{ sonst} \end{cases} \\
 \text{verschlechtereBewertung}(b) &:= \begin{cases} 0,9 \cdot \sqrt{2 \cdot (b - 0,5)} & , \text{ falls } b - 0,5 > 0, \\ 0 & , \text{ sonst} \end{cases}
 \end{aligned}$$

definiert. Wie in Abbildung A.1 verdeutlicht ist, wirkt sich der Verlust *vereinzelter* Heartbeat-Pakete kaum auf eine Bewertung aus, da bei einem Heartbeat-Empfang die Bewertung entsprechend erhöht wird. Im Falle mehrerer, aufeinander folgender Heartbeat-Verluste sinkt die Bewertung allerdings zunehmend schneller ab. Hiermit wird wiederum gewährleistet, dass der Verlust eines lokalen Repräsentanten zügig erkannt werden kann: Dies ist der Fall, sobald eine neu berechnete Bewertung den Wert 0 annimmt. Die Heartbeat-Pakete selbst werden in dieser Arbeit von LBC-Repräsentanten jede Sekunde versendet.

Literaturverzeichnis

- [1] ABAD, C. L., W. YURCIK und R. H. CAMPBELL: *A survey and comparison of end-system overlay multicast solutions suitable for network-centric warfare*. In: *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, Orlando, Florida, USA, Juli 2004.
- [2] ADJIH, C., T. CLAUSEN, P. JACQUET, A. LAOUTI, P. MUHLETHALER, A. QAYYUM und L. VIENNOT: *Optimized Link State Routing Protocol*. RFC 3626, März 2003.
- [3] AGGARWAL, S., H. BANAVAR, A. KHANDELWAL, S. MUKHERJEE und S. RANGARAJAN: *Accuracy in dead-reckoning based distributed multi-player games*. In: *Proceedings of ACM SIGCOMM 2004 workshops on NetGames '04*, Seiten 161–165, Portland, USA, September 2004.
- [4] AL HANBALI, A., E. ALTMAN und P. NAIN: *A survey of TCP over Ad Hoc Networks*. IEEE Communications Surveys & Tutorials, 7(3):22–36, Mai 2005.
- [5] BANERJEE, S. und B. BHATTACHARJEE: *A comparative study of application layer multicast protocols*. <http://www.cs.umd.edu/users/suman/publications.html>, (Überprüft im Februar 2008).
- [6] BANERJEE, S., B. BHATTACHARJEE und C. KOMMAREDDY: *Scalable Application Layer Multicast*. In: *ACM SIGCOMM 2002*, Pittsburgh, PA, USA, Juni 2002.
- [7] BANERJEE, S., S. LEE, B. BHATTACHARJEE und A. SRINIVASAN: *Resilient Multicast using Overlays*. In: *ACM Sigmetrics*, San Diego, California, USA, Juni 2003.
- [8] BAROWSKI, Y., S. BIAZ und P. AGRAWAL: *Towards the performance analysis of IEEE 802.11 in multi-hop ad-hoc networks*. In: *In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC) 2005*, Seiten 100–106, New Orleans, Louisiana, USA, März 2005.
- [9] BAUMUNG, P.: *Implementierung und Evaluation eines hierarchischen Endsystem-Multicast-Protokolls für mobile Ad-hoc-Netze*. Diplomarbeit, Universität Karlsruhe (TH), August 2003.

- [10] BAUMUNG, P.: *Stabiles Routing von Endsystem-Multicast-Strömen in Mobilien Ad-hoc-Netzen*. In: *Mobile Computing und Medienkommunikation im Internet*, Ulm, Germany, September 2004. Gesellschaft für Informatik.
- [11] BAUMUNG, P.: *Stable, Congestion-Controlled Application-Layer Multicasting in Pedestrian Ad-hoc Networks*. In: *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Seiten 57–64, Taormina, Italy, Juni 2005.
- [12] BAUMUNG, P.: *On the Modular Composition of Scalable Application-Layer Multicast Services for Mobile Ad-hoc Networks*. In: *Proceedings of the 2006 International Workshop on Wireless Ad-hoc and Sensor Networks (IWWAN 2006)*, New York, USA, Juni 2006.
- [13] BAUMUNG, P.: *TrAM: Cross-Layer Efficient Application-Layer Multicast in Mobile Ad-hoc Networks*. In: *Proceedings of the IEEE Wireless Communications and Networking Conference 2007 (WCNC 2007)*, Hong Kong, China, März 2007.
- [14] BAUMUNG, P.: *Application-Layer Multicast in MANETs: To Broadcast or not to Broadcast?* In: *Proceedings of the Fifth Annual Conference on Wireless On demand Network Systems and Services*, Seiten 133–140, Garmisch-Partenkirchen, Germany, Januar 2008.
- [15] BAUMUNG, P. et al.: *Modular P2P Multicast in Wireless Ad-hoc Networks*. <http://www.sigmobile.org/mobisys/2007/poster.html>, (Überprüft im Februar 2008).
- [16] BAUMUNG, P. et al.: *The Modular Architecture for Application-Layer Multicast*. <http://maam.pcb-net.org>, (Überprüft im Februar 2008).
- [17] BAUMUNG, P., S. PENZ und M. KLEIN: *P2P-Based Semantic Service Management in Mobile Ad-hoc Networks*. In: *Proceedings of the Seventh International Conference on Mobile Data Management (MDM 2006)*, Nara, Japan, Mai 2006.
- [18] BAUMUNG, P. und M. ZITTERBART: *Effiziente Endsystem-basierte Multicast-Dienste in mobilen Ad-hoc Netzen*. In: *Lecture Notes in Informatics*, 18. DFN-Arbeitstagung, Düsseldorf, Germany, Juni 2004.
- [19] BAUMUNG, P., M. ZITTERBART und K. KUTZNER: *Improving Delivery Ratios for Application Layer Multicast in Mobile Ad-hoc Networks*. Elsevier Special Issue on Computer Communications, 28(14):1669–1679, 2005.
- [20] BEIGBEDER, T., R. COUGHLAN, C. LUSHER, J. PLUNKETT, E. AGU und M. CLAYPOOL: *The Effects of Loss and Latency on User Performance in Unreal Tournament 2003*. In: *Proceedings of ACM Network and System Support for Games Workshop (NetGames)*, Portland, OG, USA, September 2004.

- [21] BETTSTETTER, C., G. RESTA und P. SANTI: *The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks*. IEEE Transactions on Mobile Computing, 2(3), Juli 2003.
- [22] BEURAN, R., L. T. NGUYEN, K. T. LATT, J. NAKATA und Y. SHINODA: *Wireless LAN Emulation*. Technischer Bericht S-RR-2006-015, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, Oktober 2006.
- [23] BHATTACHARJEE, S., M. H. AMMAR, E. W. ZEGURA, V. SHAH und Z. FEI: *Application-Layer Anycasting*. In: *Proceedings IEEE INFOCOM '97, The Conference on Computer Communications, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution*, Seiten 1388–1396, Kobe, Japan, April 1997.
- [24] BIANCHI, G.: *Performance analysis of the IEEE 802.11 Distributed Coordination Function*. IEEE Journal on Selected Areas in Communication, 18(3):535–547, März 2000.
- [25] BIRRER, S. und F. E. BUSTAMANTE: *Resilient peer-to-peer multicast without the cost*. In: *Proceedings of the 12th Annual Multimedia Computing and Networking Conference (MMCN'05)*, San Jose, CA, USA, Januar 2005.
- [26] BIRRER, S. und F. E. BUSTAMANTE: *Resilience in Overlay Multicast Protocols*. In: *Proceedings of the IEEE Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'06)*, Monterey, CA, USA, September 2006.
- [27] BIRRER, S., D. LU, F. E. BUSTAMANTE, Y. QIAO und P. DINDA: *FatNemo: Building a Resilient Multi-Source Multicast Fat-Tree*. In: *Proceedings of the Ninth International Workshop on Web Content Caching and Distribution (IWCW'04)*, Beijing, China, Oktober 2004.
- [28] BLÖDT, S.: *Efficient End System Multicast for Mobile Ad Hoc Networks*. In: *Proceedings of the First International Workshop on Mobile Peer-to-Peer Computing*, Seiten 75–80, Orlando, Florida, USA, März 2004.
- [29] CAMP, T., J. BOLENG und V. DAVIES: *A Survey of Mobility Models for Ad Hoc Network Research*. Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2(5):483–502, September 2002.
- [30] CARLSSON, B. und R. GUSTAVSSON: *The rise and fall of Napster: An evolutionary approach*. In: *Proceedings of the 6th International Computer Science Conference on Active Media Technology*, Hong Kong, Dezember 2001.

- [31] CASTRO, M.: *An evaluation of scalable application-level multicast built using peer-to-peer overlay networks*. In: *Proceedings IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, USA, April 2003.
- [32] CASTRO, M., P. DRUSCHEL, A. KERMARREC und A. ROWSTRON: *Scalable application-level anycast for highly dynamic groups*. In: *Proceedings of Networked Group Communication 2003*, Munich, Germany, September 2003.
- [33] CHATZIMISIOS, P., A. C. BOUCOUVALAS und V. VITSAS: *Effectiveness of RTS/CTS handshake in IEEE 802.11a Wireless LANs*. *Electronics Letters*, 40(14):915–916, Juli 2004.
- [34] CHAWATHE, Y., S. RATNASAMY, L. BRESLAU, N. LANHAM und S. SHENKER: *Making Gnutella-like P2P Systems Scalable*. In: *Proceedings of ACM SIGCOMM 2003*, Karlsruhe, Germany, August 2003.
- [35] CHEN, K., K. NAHRSTEDT und N. VAIDYA: *The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks*. In: *Proceedings of the IEEE Wireless Communications and Networking Conference 2004 (WCNC 2004)*, Atlanta, USA, März 2004.
- [36] CHEN, W.-T. und W.-T. LEE: *Wireless On-Demand Network Systems*, Band 2928/2003 der Reihe *Lecture Notes in Computer Science*, Kapitel Dynamic AODV Backup Routing in Dense Mobile Ad-Hoc Networks, Seiten 158–170. Springer Berlin / Heidelberg, 2004.
- [37] CHOI, Y., E. LEE, D. LEE, S. PARK, F. YU und S.-H. KIM: *An Implicit Cluster-based Overlay Multicast Protocol for Mobile Ad hoc Networks*. In: *Proceedings of the IEEE Wireless Communications and Networking Conference 2007 (WCNC 2007)*, Hong Kong, China, März 2007.
- [38] CHU, Y., S. G. RAO und H. ZHANG: *A Case for End System Multicast*. In: *ACM SIGMETRICS 2000*, Santa Clara, California, USA, Juni 2000.
- [39] CONRAD, M., J. DINGER, H. HARTENSTEIN, D. ROLLI, M. SCHÖLLER und M. ZITTERBART: *A Peer-to-Peer Framework for Electronic Markets*, Band 3485, *Lecture Notes on Computer Science*. Kapitel 30, Seiten 509–525. Springer, September 2005.
- [40] CONRAD, M. und H.-J. HOF: *A Generic, Self-Organizing, and Distributed Bootstrap Service for Peer-to-Peer Networks*. In: *Proceedings of New Trends in Network Architectures and Services: 2nd International Workshop on Self-Organizing Systems (IWSOS 2007)*, English Lake District, UK, September 2007.

- [41] CRAMER, C. und T. FUHRMANN: *Bootstrapping Chord in Ad hoc Networks: Not Going Anywhere for a While*. In: *Proceedings of the 3rd IEEE International Workshop on Mobile Peer-to-Peer Computing*, Pisa, Italy, März 2006.
- [42] DEERING, S.: *Multicast Routing in Internetworks and Extended LANs*. In: *ACM SIGCOMM 1988*, Seiten 55–64, Stanford, California, USA, August 1988.
- [43] DEGENER, J. und C. BORMANN: *The GSM 06.10 lossy speech compression library and its applications*. <http://kbs.cs.tu-berlin.de/~jutta/toast.html>, (Überprüft im Februar 2008).
- [44] DICK, M., O. WELLNITZ und L. WOLF: *Analysis of factors affecting players performance and perception in multiplayer games*. In: *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games (NetGames)*, New York, NY, USA, Oktober 2005.
- [45] ERGEN, M. und P. VARAIYA: *Delay Analysis of Distributed Coordination Function in IEEE 802.11*. In: *In Proceedings of IEEE GLOBECOM 2006*, San Francisco, California, USA, November 2006.
- [46] FÄRBER, J.: *Network game traffic modelling*. In: *Proceedings of the 1st workshop on Network and system support for games (NetGames)*, Braunschweig, Germany, April 2002.
- [47] GANJALI, Y. und A. KESHAVARZIAN: *Load balancing in ad hoc networks: single-path routing vs. multi-path routing*. In: *Proceedings of INFOCOM 2004, The Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, Hong Kong, China, März 2004.
- [48] GE, M., S. KRISHNAMURTHY und M. FALOUTSOS: *Overlay Multicasting for Ad Hoc Networks*. In: *Proceedings Of The Third Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Bordum, Turkey, Juni 2004.
- [49] GERLA, M., K. TANG und R. BAGRODIA: *TCP Performance in Wireless Multi-hop Networks*. In: *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, Februar 1999.
- [50] GLOMoSIM: *Global Mobile Information Systems Simulation Library*. <http://pcl.cs.ucla.edu/projects/gloMosim/>, (Überprüft im Februar 2008).
- [51] GUI, C. und P. MOHAPATRA: *Efficient Overlay Multicast for Mobile Ad hoc Networks*. In: *The Wireless Communications and Networking Conference (WCNC)*, Seiten 1118–1123, New Orleans, Louisiana, USA, März 2003.
- [52] HOLLAND, G. und N. H. VAIDYA: *Analysis of TCP Performance Over Mobile Ad Hoc Networks*. In: *Proceedings of IEEE/ACM MOBICOM '99*, Seiten 219–230, Seattle, WA, USA, August 1999.

- [53] HONG, X., M. GERLA, G. PEI und C. CHIANG: *A Group Mobility Model for Ad Hoc Wireless Networks*. In: *Proceedings of ACM International Workshop on Modelling and Simulation of Wireless and Mobile Systems*, Seiten 53–60, Seattle, USA, August 1999.
- [54] HU, Y. C., S. DAS und H. PUCHA: *Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks*. In: *Proceedings of HotOS-IX: Ninth Workshop on Hot Topics in Operating Systems*, Lihue, Kauai, Hawaii, Mai 2003.
- [55] HÜBSCH, C.: *Synchronisation von Peer-to-Peer-basierten Multiplayer-Anwendungen mit Echtzeitanforderungen in Ad-Hoc-Netzen*. Diplomarbeit, Universität Karlsruhe (TH), Oktober 2006.
- [56] HUGHES, B., M. GLEESON, M. KARPINSKI, R. CUNNINGHAM und V. CAHILL: *Real-Time Communication in IEEE 802.11 Mobile Ad hoc Networks A Feasibility Study*. Technischer Bericht, Department of Computer Science, Trinity College Dublin, Irland, September 2006.
- [57] HUNG, F.-Y. und I. MARSIC: *Effectiveness of Physical and Virtual Carrier Sensing in IEEE 802.11 Wireless Ad Hoc Networks*. In: *Proceedings of the IEEE Wireless Communications and Networking Conference 2007 (WCNC 2007)*, Hong Kong, China, März 2007.
- [58] HYYTIÄ, E., P. LASSILA und J. VIRTAMO: *Spatial node distribution of the random waypoint mobility model with applications*. *IEEE Transactions on Mobile Computing*, 5(6):680–694, Juni 2006.
- [59] JARDOSH, A., E. M. BELDING-ROYER, K. C. ALMERTH und S. SURI: *Towards realistic mobility models for mobile ad hoc networks*. In: *Proceedings of the 9th annual international conference on Mobile computing and networking*, Seiten 217–229, San Diego, CA, USA, September 2003.
- [60] JONES, E. P. C., M. KARSTEN und WARD P. A. S.: *Multipath Load Balancing in Multi-hop Wireless Networks*. In: *Proceedings of WiMob 2005 - IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Montreal, Canada, August 2005.
- [61] KAASHOEK, M. F. und D. R. KARGER: *Koorde: A simple degree-optimal distributed hash table*. In: *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA, USA, Februar 2003.
- [62] KIM, K.-I. und S.-H. KIM: *DESIRE: Density aware heterogenous overlay multicast forwarding scheme in Mobile Ad Hoc Networks*. *IEICE Transactions on Communications*, E88-B(9):3579–3587, September 2005.

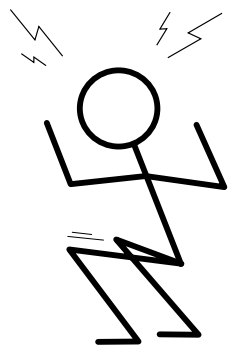
- [63] KRÄMER, B.: *Entwicklung und Evaluation einer effizienten Overlay-Struktur zur Endsystem-basierten Gruppenkommunikation in mobilen Ad-hoc Netzen*. Diplomarbeit, Universität Karlsruhe (TH), August 2005.
- [64] LANG, T., P. BRANCH und G. ARMITAGE: *A Synthetic Traffic Model for Quake3*. In: *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, New York, NY, USA, Januar 2004.
- [65] LEE, S.-J und M. GERLA: *AODV-BR: Backup Routing in Ad hoc Networks*. In: *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, USA, September 2000.
- [66] LEE, S.-J., M. GERLA und C.-C. CHIANG: *On-Demand Multicast Routing Protocol*. In: *Proceedings of IEEE WCNC 1999*, Seiten 1298–1304, New Orleans, USA, September 1999.
- [67] LEE, S.-J., W. SU, J. HSU, M. GERLA und R. BAGRODIA: *A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols*. In: *Proceedings of IEEE INFOCOM 2000*, Seiten 565–574, Tel Aviv, Israel, März 2000.
- [68] LEUCHTNER, M.: *Entwicklung und Evaluation einer Endsystem-basierten Chat-Anwendung für mobile ad-hoc Netze*. Diplomarbeit, Universität Karlsruhe (TH), Januar 2005.
- [69] LI, J., J. STRIBLING, R. MORRIS, M. F. KAASHOEK und T. M. GIL: *A performance vs. cost framework for evaluating DHT design tradeoffs under churn*. In: *In Proceedings of the 24th Infocom*, Miami, Florida, USA, März 2005.
- [70] LOCHERT, C., B. SCHEUERMANN und M. MAUVE: *A Survey on Congestion Control for Mobile Ad-Hoc Networks*. Wiley Wireless Communications and Mobile Computing, 7:655–676, April 2007.
- [71] OIKARINEN, J. und D. REED: *Internet Relay Chat Protocol*. RFC 1459 (Experimental), Mai 1993. Updated by RFCs 2810, 2811, 2812, 2813.
- [72] ONG, H.-L., T. C. WAN, ONG H.-K. und S. RAMADASS: *Node Density Effects on Reactive Routing Protocols for Mobile Ad Hoc Wireless Networks*. In: *AINTEC*, Seiten 206–221, Pathumthani, Thailand, November 2006.
- [73] PANTEL, L. und L. WOLF: *On the impact of delay on real-time multiplayer games*. In: *Proceedings of the 12th international Workshop on Network and Operation Systems Support for Digital Audio and Video (NOSSDAV)*, Miami Beach, FL, USA, Mai 2002.
- [74] PEI, G., M. GERLA und T.-W. CHEN: *Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks*. In: *In Proceedings of the IEEE International Conference on Communications (ICC)*, Seiten 70–74, New Orleans, Louisiana, USA, Juni 2000.

- [75] PENDARAKIS, D., S. SHI, D. VERMA und M. WALDVOGEL: *ALMI: An application level multicast infrastructure*. In: *3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, Seiten 49–60, San Francisco, USA, März 2001.
- [76] PERKINS, C. E., E. M. BELDING-ROYER und S. R. DAS: *Ad hoc On-Demand Distance Vector (AODV) Routing*, Februar 2003.
- [77] PLEISCH, S., M. BALAKRISHNAN, K. BIRMAN und R. VAN RENESSE: *Mistral: Efficient Flooding in Mobile Ad-hoc Networks*. In: *In Proceedings of the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc 2006)*, Florence, Italy, Mai 2006.
- [78] POSTEL, J.: *Transmission Control Protocol*. RFC 761, Januar 1980.
- [79] QAYYUM, A., L. VIENNOT und A. LAOUITI: *Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks (RR-3898)*. Technischer Bericht, INRIA - Rocquencourt, Equipe: HIPERCOM, März 2000.
- [80] QUAX, P., P. MONSIEURS, W. LAMOTTE, D. DE VLEESCHAUWER und N. DEGRANDE: *Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game*. In: *Proceedings of Workshop on Network and System Support for Games (NetGames)*, Portland, OG, USA, September 2004.
- [81] RAJENDRAN, V., K. OBRACZKA, Y. YI, S.-J. LEE, K. TANG und M. GERLA: *Combining Source- and Localized Recovery to Achieve Reliable Multicast in Multi-Hop Ad Hoc Networks*. In: *Proceedings of the Networking 2004*, Athens, Greece, Mai 2004.
- [82] RAJENDRAN, V., Y. YI, K. OBRACZKA, S. J. LEE, K. TANG und M. GERLA: *Reliable, Adaptive, Congestion-Controlled Adhoc Multicast Transport Protocol: Combining Source-based and Local Recovery*. Technischer Bericht, University of California, Santa Cruz, 2003.
- [83] RAPPAPORT, T. S.: *Wireless Communications: Principles & Practice*. Prentice Hall, 2002.
- [84] RATNASAMY, S., P. FRANCIS, M. HANDLEY, R. KARP und S. SHENKER: *A Scalable Content-Addressable Network*. In: *Proceedings of ACM SIGCOMM 2001*, San Diego, CA, USA, August 2001.
- [85] RATNASAMY, S., M. HANDLEY, R. KARP und S. SHENKER: *Application-level Multicast using Content-Addressable Networks*. In: *Proceedings of the Third International COST264 Workshop on Networked Group Communication*, London, UK, November 2001.

- [86] REKHTER, Y. und T. LI: *A Border Gateway Protocol 4 (BGP-4)*. RFC 1771, März 1995.
- [87] RIPEANU, M.: *Peer-to-Peer Architecture Case Study: Gnutella Network*. In: *Proceedings of IEEE 1st International Conference on Peer-to-peer Computing*, Linköping Sweden, August 2001.
- [88] ROWSTRON, A. und P. DRUSCHEL: *Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems*. In: *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, Heidelberg, November 2001.
- [89] ROYER, E. M. und C. E. PERKINS: *Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol*. In: *Proceedings of MobiCom '99, Seattle, WA, USA*, Seiten 207–218, August 1999.
- [90] SCALABLE NETWORK TECHNOLOGIES, INC.: *QualNet*. <http://www.scalable-networks.com/>, (Überprüft im Februar 2008).
- [91] SCHEUERMANN, B., C. LOCHERT und M. MAUVE: *Implicit Hop-by-Hop Congestion Control in Wireless Multihop Networks*. *Ad Hoc Networks*, 6:260–286, April 2008.
- [92] SCHEUERMANN, B., M. TRANSIER, C. LOCHERT, M. MAUVE und W. EFFELBERG: *Backpressure Multicast Congestion Control in Mobile Ad-Hoc Networks*. In: *Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, New York, NY, USA, Dezember 2007.
- [93] SCHOLLMEIER, R., I. GRUBER und M. FINKENZELLER: *Routing in Mobile Ad-hoc and Peer-to-Peer Networks: A Comparison*. In: *Networking 2002: International Workshop on Peer-to-Peer Computing*, Mai 2002.
- [94] SOBEIH, A., W. YURCIK und J. C. HOU: *VRing: A Case for Building Application-Layer Multicast Rings (Rather Than Trees)*. In: *Proceedings of the 12th Annual Meeting of the IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MAS-COTS)*, Volendam, The Netherlands, Oktober 2004.
- [95] SOCIETY, THE INTERNET: *iLBCfreeware.org Project Homepage*. <http://www.ilbcfreeware.org/>, (Überprüft im Februar 2008).
- [96] STEVENS, W.: *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*. RFC 2001, Januar 1997.
- [97] STEVENS, W. R.: *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

- [98] STOICA, I., R. MORRIS, D. KARGER, M. F. KAASHOEK und H. BALAKRISHNAN: *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*. In: *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.
- [99] SZPIRO, G. G.: *Kepler's conjecture*. Wiley, 2003.
- [100] TAKAHASHI, H. und A. MATSUYAMA: *An approximate solution for the steiner problem in Graphs*,. *Mathematica Japonica*, 24(6):573–577, 1980.
- [101] TANG, H., M. JANIC und X. ZHOU: *Hopcount in the NICE Application Layer Multicast Protocol*. In: *IEEE/SMC the multiconference on Computational Engineering in Systems Applications*, Beijing, China, Oktober 2006.
- [102] TANG, K., K. OBRACZKA, S.-J. LEE und M. GERLA: *Congestion Controlled Adaptive Lightweight Multicast in Wireless Mobile Ad Hoc Networks*. In: *Proceedings of the Seventh IEEE Symposium on Computers and Communications*, Taormina, Italy, Juli 2002.
- [103] TANG, K., K. OBRACZKA, S.-J. LEE und M. GERLA: *A Reliable, Congestion-Controlled Multicast Transport Protocol in Multimedia Multi-hop Networks*. In: *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, Honolulu, Hawaii, Oktober 2002.
- [104] TANG, K., K. OBRACZKA, S.-J. LEE und M. GERLA: *Reliable Adaptive Lightweight Multicast Protocol*. In: *Proceedings of the International Conference on Communications*, Anchorage, Alaska, USA, Mai 2003.
- [105] TANG, S. und B. ZHANG: *A robust AODV protocol with local update*. In: *Proceedings of the 10th Asia-Pacific Conference on Communications*, Seiten 418–422, Beijing, China, September 2004.
- [106] TINNIRELLO, I., S. CHOI und Y. KIM: *Revisit of RTS/CTS Exchange in High-Speed IEEE 802.11 Networks*. In: *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Taormina, Italy, Juni 2005.
- [107] TORRENT-MORENO, M., S. CORROY, F. SCHMIDT-EISENLOHR und H. HARTENSTEIN: *IEEE 802.11-Based One-Hop Broadcast Communications: Understanding Transmission Success and Failure under Different Radio Propagation Environments*. In: *Proceedings of The 9-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, Seiten 68–77, Torremolinos, Malaga, Spain, Oktober 2006.
- [108] TÓTH, G. F., P. GRITZMANN und J. M. WILLS: *Finite sphere packing and sphere covering*. *Discrete and Computational Geometry*, 4(1):19–40, Dezember 1989.

-
- [109] VALIN, J.-M. et al.: *Speex: A Free Codec For Free Speech*. <http://www.speex.org/>, (Überprüft im Februar 2008).
- [110] WAITZMAN, D., C. PARTRIDGE und S. DEERING: *Distance Vector Multicast Routing Protocol*. RFC 1075, November 1988.
- [111] WENIGER, KILIAN: *IP-Autokonfiguration in mobilen Ad-hoc-Netzwerken*. Shaker Verlag Ed., Aachen, September 2004.
- [112] WENIGER, KILIAN: *PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks*. IEEE Journal on Selected Areas in Communications (JSAC) Special Issue, 23(3):507–519, März 2005.
- [113] WITTMANN, R. und M. ZITTERBART: *Multicast Communication: Protocols and Applications*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.
- [114] XIE, J., R. TALPADE, A. MCAULEY und M. LIU: *AMRoute: Adhoc Multicast Routing Protocol*. Mobile Networks and Applications, 7(6):429–439, Dezember 2002.
- [115] XU, K., M. GERLA und S. BAE: *Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks*. Ad Hoc Networks, 1(1):107–123, Juli 2003.
- [116] YI, Y.: *Efficient Flooding in Ad hoc Networks: a Comparative Performance Study*. In: *In proceedings of the IEEE International Conference on Communications (ICC)*, Anchorage, Alaska, USA, Mai 2003.
- [117] YI, Y., M. GERLA und T. KWON: *Efficient Flooding in Ad Hoc Networks using On-Demand Passive Cluster Formation*. In: *In proceedings of the Second Annual Mediterranean Ad Hoc Networking Workshop (Med-hoc-Net 2003)*, Mahdia, Tunisia, Juni 2003.
- [118] ZENG, X., R. BAGRODIA und M. GERLA: *GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks*. In: *Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, Banff, Alberta, Canada, Mai 1999.
- [119] ZIMMERMANN, H.: *OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection*. IEEE Transactions on Communications, 28(4):425–432, April 1980.
- [120] ZITTERBART, M., B. STILLER und A. N. TANTAWY: *A Model for Flexible High-Performance Communication Subsystems*. IEEE Journal on Selected Areas in Communications, 11(4):507–518, Mai 1993.



Universität Karlsruhe (TH)
Institut für Telematik



ISBN: 978-3-86644-268-9

www.uvka.de