Institut für Parallele und Verteilte Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 2929

Geographic Routing Protocols for Wireless Mesh Networks

Geographische Vermittlungsprotokolle für drahtlose vermaschte Netze

Damian Cassius Philipp

Studiengang:	Informatik
Prüfer:	Prof. Dr. Kurt Rothermel
Betreuer:	Dipl.-Inf. Frank Dürr
begonnen am:	18. Mai 2009
beendet am:	17. November 2009
CR-Klassifikation:	C.2.1, C.2.3, C.2.5

Inhaltsverzeichnis

1 Einleitung **11**
1.1 Motivation . . . 11
1.1.1 Mesh-Netzwerke . . . 11
1.1.2 Geocast . . . 12
1.1.3 Lokalitätseigenschaften von Geocast in Mesh-Netzwerken . . . 13
1.2 Zielsetzung . . . 13
1.3 Gliederung der Arbeit . . . 14

2 Grundlagen und verwandte Arbeiten **15**
2.1 Grundlagen zum Routing . . . 15
2.2 Einordnung in das Schichtenmodell . . . 15
2.3 Zugrunde liegende Routingprotokolle . . . 16
2.3.1 Ad-Hoc On-Demand Distance Vector Protokoll . . . 16
2.3.2 Optimized Link State Routing . . . 17
2.3.3 Better Approach To Mobile Ad-Hoc Networking (B.A.T.M.A.N.) . . . 19
2.4 Multicast-Routing . . . 19
2.4.1 Multicast mit AODV . . . 20
2.4.2 Multicast Optimized Link State Routing . . . 21
2.5 Geocast-Kommunikation . . . 21
2.5.1 Adressierungskonzepte & Lokationsmodelle . . . 22
2.5.2 Vermittlungsalgorithmen . . . 24
2.6 Metriken für Routing-Algorithmen . . . 25
2.6.1 Hopcount . . . 25
2.6.2 Linkqualität . . . 26
2.6.3 Implizite Metriken . . . 26

3 Systemmodell & Anforderungen **27**
3.1 Systemmodell . . . 27
3.1.1 Konnektivität . . . 27
3.1.2 Fehlerfälle . . . 28
3.1.3 Lokationsinformationen . . . 29
3.2 Anforderungen . . . 30
3.2.1 Grundlegende Anforderungen . . . 31
3.2.2 Effektivität . . . 31
3.2.3 Effizienz . . . 32

4 Entwurf **33**

4.1 Übersicht und Taxonomie der Routingverfahren . . . 33
4.2 Lokationsinformationen . . . 36
4.3 Multicastbasierende Implementierungen . . . 37
4.3.1 Beziehung zwischen Geocast und Multicast . . . 37
4.3.2 Adressierung . . . 38
4.3.3 Geo-Multicast-Routing bei AODV . . . 38
4.3.4 Geo-Multicast-Routing bei OLSR . . . 39
4.3.5 Nachteile einer Multicast-Lösung . . . 40
4.4 Multi-Path-Routing . . . 41
4.4.1 Das Netz als Vektorfeld . . . 41
4.4.2 AODV . . . 42
4.4.3 OLSR . . . 45
4.4.4 Fazit . . . 48
4.5 Single-Path-Routing . . . 48
4.5.1 Geolimitierter Broadcast . . . 50
4.5.2 Reaktiver Geocast . . . 52
4.5.3 Hierarchischer Geocast . . . 57
4.5.4 Fazit . . . 63

5 Implementierung 65
5.1 Gemeinsame Funktionalitäten . . . 65
5.1.1 Geolimitierter Broadcast . . . 65
5.1.2 Handhabung von Lokationsinformationen . . . 66
5.1.3 TCL-Schnittstelle . . . 68
5.1.4 Logging . . . 68
5.2 Reaktiver Geocast auf Basis von AODV . . . 69
5.2.1 Duplikaterkennung . . . 69
5.2.2 Parameter . . . 70
5.2.3 Konstanten . . . 71
5.2.4 Paketformat . . . 71
5.2.5 TCL-Schnittstelle . . . 73
5.2.6 Utility-Klassen . . . 73
5.2.7 Geocast-Routingtabelle . . . 75
5.2.8 Abfrage der Lokation . . . 75
5.2.9 Logging . . . 75
5.2.10 Befehls- und Paketverarbeitung . . . 76
5.3 Hierarchischer Geocast auf Basis von OLSR . . . 78
5.3.1 Architektur des Routers . . . 78
5.3.2 Duplikaterkennung . . . 79
5.3.3 Parameter . . . 79
5.3.4 Konstanten . . . 80
5.3.5 Paketformat . . . 80
5.3.6 TCL-Schnittstelle . . . 83
5.3.7 Utility-Klassen . . . 83
5.3.8 Abfrage der Lokation . . . 84

5.3.9 Logging . . . 84
5.3.10 Befehls- und Paketverarbeitung . . . 85

6 Evaluierung **91**
6.1 Evaluierungsumgebung . . . 91
6.1.1 Netz-Infrastruktur . . . 91
6.1.2 Mobilität und Knotenplatzierung . . . 91
6.1.3 Lokationsinformationen . . . 94
6.1.4 Verkehrsmodell Geocast . . . 94
6.1.5 Verkehrsmodell Unicast . . . 94
6.2 Untersuchte Größen . . . 95
6.3 Ungenauigkeiten der Simulation . . . 95
6.4 Ergebnisse . . . 96
6.4.1 Routingfehler und Nachrichtenverlust . . . 96
6.4.2 Verzögerung von Nachrichten . . . 105
6.4.3 Abdeckung des Zielgebietes . . . 106
6.4.4 Fairness . . . 107
6.4.5 Pfadlänge . . . 108
6.5 Fazit . . . 108

7 Zusammenfassung und Ausblick **111**
7.1 Zusammenfassung . . . 111
7.2 Offene Fragen . . . 112

Literaturverzeichnis **115**

Abbildungsverzeichnis

2.1 Beispiel für Lokationen, die sich nicht als Baum anordnen lassen. 23

4.1 Single-Path- und Multi-Path-Routing im Mesh-Netz 35
4.2 Illustration des Vektorfeldes der Routen für ein Zielgebiet 41
4.3 Routenfindung mittels ERS 43
4.4 Abdeckung des Zielgebietes bei Wiederverwendung bestehender Routen . . . 44
4.5 Catch-All-Problem bei aggregierten Routen 46
4.6 Propagieren von Routing-Nachrichten bei OLSR: Routingschleifen 48
4.7 Rautenprobleme bei erweitertem Horizont 49
4.8 Notwendigkeit für limitiertes Fluten bei geolimitiertem Broadcast 51
4.9 Problematik der suboptimalen Routen („Eckenproblem“) 54
4.10 Optimierung von Geocast-Routen 55

5.1 Überblick über die Komponenten eines Knotens bei reaktivem Geocast 69
5.2 Überblick über die Komponenten eines Knotens bei hierarchischem Geocast . 78

6.1 Raumplan des Informatik-Gebäudes der Universität Stuttgart, Vaihingen, 2. Stock 92
6.2 Informatik-Gebäude der Universität Stuttgart, Vaihingen 93
6.3 Nachrichtenverlustraten bei AODV, 30 Knoten 98
6.4 Nachrichtenverlustraten bei AODV, 50 Knoten 99
6.5 Nachrichtenverlustraten bei AODV, 100 Knoten 100
6.6 Nachrichtenverlustraten bei OLSR, 30 Knoten 102
6.7 Nachrichtenverlustraten bei OLSR, 50 Knoten 103
6.8 Nachrichtenverlustraten bei OLSR, 100 Knoten 104
6.9 Kumulierte Verzögerung der Auslieferung von Nachrichten 106
6.10 Kumulierte Abdeckung des Zielgebietes 107
6.11 Kumulierte Verteilung von Sendeereignissen im Netz 108
6.12 Kumulierte Pfadlänge von ausgelieferten Nachrichten 109

Tabellenverzeichnis

4.1 Klassifikation von Geocast-Routingalgorithmen 34

Verzeichnis der Listings

5.1 Typdefinition eines `GEO-RREQ`-Paketes 72
5.2 Typdefinition eines `GEO-RREP`-Paketes 72
5.3 Typdefinition eines `GEO-DATA`-Paketes 73
5.4 Typdefinition eines `GEO-RDEL`-Paketes 73
5.5 Typdefinition einer `DRA`-Nachricht 81
5.6 Typdefinition einer `DRS`-Nachricht 81
5.7 Typdefinition einer `GEO-DATA`-Nachricht 82
5.8 Typdefinition einer `WRAPPER`-Nachricht 82
5.9 Typdefinition einer `DRSS`-Nachricht 83

Verzeichnis der Algorithmen

5.1 Geolimitierter Broadcast 66
5.2 Präfixprüfung der Lokation 67
5.3 Erkennung von Duplikaten beim Empfang 70

Abstract

Wireless mesh networks allow building cheap scalable networks to serve mobile users in large areas. Geocast is a kind of group communication, whereby the group of recievers of a message is defined by their presence in a given geographic area. The correlation of geographical and topological distance between nodes in wireless mesh networks enables new optimizations for geocast routing protocols.

This work presents the performance of existing multicast routing protocols built for wireless mesh networks when used to transport geocast messages. It also includes upcoming problems in newly developed multi-path routing protocols and possible single-path routing protocols for the dissemination of geocast messages. Moreover, two single-path routing protocols based on the unicast routing protocols AODV and OLSR have been developed.

Both these protocols have been analyzed for their performance in simulations using the network simulator ns2. The 2nd floor of the computer science building at the University of Stuttgart has been used as an environmental model. Geocast-routing based on AODV networks proved to work well with a rather constrained set of network parameters only. Geocast-routing based on OLSR networks showed worse performance. This, however, might be due to the fact that an outdated version of OLSR had to be used for the simulation. This could have distorted the results.

Zusammenfassung

Mit Hilfe von drahtlosen vermaschten Netzwerken (Mesh-Netzen) lassen sich kostengünstig skalierbare Netzwerke aufbauen, die mobile Anwender in großen Gebieten versorgen können. Geocast ist eine Art der Nachrichtenverteilung, bei der die Gruppe der Empfänger einer Nachricht durch die Anwesenheit in einem bestimmten Zielgebiet definiert wird. In Mesh-Netzwerken eröffnet die Korrelation von geographischer und topologischer Nähe von Knoten neue Möglichkeiten zur Optimierung von Geocast-Routingprotokollen.

In dieser Arbeit wird untersucht, in wie weit sich bereits bestehende Multicast-Routingprotokolle für Mesh-Netze zur Verteilung von Geocast-Nachrichten eignen, welche Probleme sich bei neuentwickelten Multi-Path-Routingprotokollen für Geocast ergeben und wie sich Single-Path-Routingprotokolle für Geocast entwickeln lassen. Auf Basis der Mesh-Unicast-Routingprotokolle AODV und OLSR werden zwei Single-Path-Geocast-Routingprotokolle entworfen.

Die entworfenen Protokolle werden in Simulationen mit dem Netzwerksimulator ns2 auf ihre Performanz hin untersucht. Als Umgebungsmodell für die Simulation dient der 2. Stock des Informatik-Gebäudes an der Universität Stuttgart. Es zeigt sich, dass Geocast-Routing auf Basis von AODV nur innerhalb enger Netzwerkparameter gut funktioniert. Geocast-Routing auf Basis von OLSR zeigt eine durchgängig geringere Performanz. Jedoch konnte nur eine veraltete Variante von OLSR für die Simulationen eingesetzt werden, die die Simulationsergebnisse möglicherweise verfälscht.

1 Einleitung

1.1 Motivation

Mit Hilfe von drahtlosen vermaschten Netzwerken (Mesh-Netze) lassen sich kostengünstig skalierbare Netzwerke aufbauen, die mobile Anwender in großen Gebieten versorgen können. Einsatzszenarien für solche Netze sind vielfältig. Ein häufig angeführtes Szenario ist der Katastrophenschutz. Ohne eine vorher bestehende Infrastruktur können mit Mesh-Netzen schnell Kommunikationsverbindungen zwischen Einsatzkräften und Leitstellen hergestellt werden. Es sind bereits einige Mesh-Projekte im praktischen Einsatz. Google Inc. bietet in Mountain View, CA, USA einen Internetzugang über ein eigens dafür aufgebautes Mesh-Netz an ([Inc08]). Im deutschsprachigen Raum existieren in vielen Großstädten Freifunk-Netzwerke ([Pro09]). Dabei stellen Privatpersonen einen meshfähigen Router auf. Ist die Dichte der Router groß genug, ergibt sich ein WLAN, das sich über die gesamte Stadt erstreckt.

1.1.1 Mesh-Netzwerke

Als Mesh-Netzwerke bezeichnet man Funknetze, bei denen Netzknoten Daten für andere Knoten, die sich nicht in direkter Funkreichweite befinden, weiterleiten. Mesh-Netze können sich spontan bilden, wenn sich meshfähige Geräte in gegenseitiger Funkreichweite befinden. Genauso ist es auch möglich, dass ein Betreiber oder eine Gruppe von Betreibern meshfähige Geräte an bestimmten Punkten installiert, die dann sowohl meshfähigen als auch nicht-meshfähigen Knoten den Zugang zum Netzwerk ermöglichen. In beiden Fällen können meshfähige Knoten am Rande des Netzwerkes die Reichweite erhöhen.

Typen von Mesh-Knoten Es gibt zwei Arten von Geräten in Mesh-Netzen: Infrastrukturgeräte (Mesh-Router) und Anwendergeräte (Mesh-Clients).

Mesh-Router sind Geräte ohne Benutzerschnittstelle, die nur für den Netzbetrieb arbeiten. Als solche werden sie ortsfest installiert. Mesh-Netze können wie bereits erwähnt auch ohne eine solche Infrastruktur funktionieren. Jedoch wird durch solche Knoten sichergestellt, dass das Netz in dem von ihnen abgedeckten Bereich verfügbar ist und entfernte Partitionen eines Netzes kommunizieren können.

Mesh-Clients sind dagegen Geräte mit Benutzerschnittstelle. Diese können, etwa als Desktop-Rechner, fest installiert sein. Zumeist handelt es sich aber um mobile Geräte wie Laptops oder Smartphones.

Da nach Erfahrungen mit bisherigen Infrastruktur-WLAN-Netzen jeder Infrastruktur-Knoten (analog zum Mesh-Router) mehrere Infrastruktur-Clients (analog zum Mesh-Client) versorgen kann und die Anschaffung und Aufstellung von Mesh-Routern mit Kosten verbunden ist, ist davon auszugehen, dass in produktiven Mesh-Netzen die Anzahl der Clients die der Router deutlich überwiegt.

Kommunikation benachbarter Knoten Mesh-Knoten werden als benachbart angesehen, sofern sie sich in direkter Funkreichweite befinden bzw. wenn sie ohne die Mitwirkung anderer Knoten miteinander kommunizieren können. Dabei ist zu unterscheiden, ob eine Verbindung uni- oder bidirektional genutzt werden kann. Aufgrund von unterschiedlicher Abstrahlcharakteristiken von Antennen, Umgebungseinflüssen oder technischen Unterschieden der Knoten kann es passieren, dass ein Knoten Daten direkt an einen anderen Knoten senden kann, in der umgekehrten Richtung jedoch keine Datenübertragung möglich ist. Unidirektionale Verbindungen werden von Mesh-Routingalgorithmen in der Regel nicht berücksichtigt. Die Kommunikation zwischen benachbarten Knoten wird als Single-Hop-Kommunikation bezeichnet.

Kommunikation nicht benachbarter Knoten Wie bereits beschrieben, assistieren Knoten auch bei der Kommunikation zwischen nicht direkt benachbarten Knoten. Dabei wird eine Nachricht von Knoten zu Knoten weitergegeben, bis sie ihr Ziel erreicht hat. Jede Zwischenstation, die die Nachricht bearbeitet, wird dabei als ein Hop angesehen; jeder Knoten leitet die Nachricht zum nächsten Nachbar in Richtung des Zieles, dem Next-Hop der Route, weiter. Im Gegensatz zur Kommunikation zwischen direkt benachbarten Knoten wird diese Art der Datenübertragung als Multi-Hop-Kommunikation bezeichnet.

Fluktuationen Aufgrund mobiler Knoten oder sich verändernder Umgebungsbedingungen kann sich die Topologie eines Mesh-Netzes zu jeder Zeit ändern. Die Änderungsrate ist hinsichtlich der Mobilität durch die Bewegungsgeschwindigkeit der Knoten begrenzt. Änderungen aufgrund der Umgebungsbedingungen können zwar mit beliebiger Geschwindigkeit erfolgen, treten aber typischerweise nur sehr selten auf. Die Anpassung von Routingalgorithmen an diese Topologieänderungen ist eine der großen Herausforderungen bei der Entwicklung von Mesh-Netzwerken.

1.1.2 Geocast

Geocast ist eine Art der Nachrichtenverteilung, bei der die Gruppe der Empfänger einer Nachricht durch die Anwesenheit in einem bestimmten Zielgebiet definiert wird. Es gibt mehrere Beispiele von Nachrichten, die nur für bestimmte Gebiete von Interesse sind. So sind etwa Erdbebenwarnungen außerhalb des betroffenen Gebietes relativ bedeutungslos. Werden jedoch alle Personen innerhalb des betroffenen Gebietes schnell benachrichtigt, können möglicherweise Menschenleben gerettet werden.

Geocast ist jedoch auch für kommerzielle Anwendungen von Interesse. So können etwa Nachrichten über die neuesten Modetrends an die Kunden eines Bekleidungsgeschäftes gesendet werden, während die Besucher des auf dem Platz vor dem Ladengeschäft stattfindenden Wochenmarktes mit Rezeptideen versorgt werden.

1.1.3 Lokalitätseigenschaften von Geocast in Mesh-Netzwerken

In Mesh-Netzwerken definiert sich der topologische Abstand zweier Knoten daraus, wie viele andere Knoten notwendig sind, um die geographische Entfernung der beiden Knoten zu überbrücken. Daraus ergibt sich wiederum, dass in Mesh-Netzen geographische und topologische Nähe korrelieren, d.h. dass die Knoten innerhalb eines per Geocast adressierten Zielgebietes mit hoher Wahrscheinlichkeit topologisch sehr nahe beieinander liegen.

1.2 Zielsetzung

Mit Multicast existiert bereits eine Möglichkeit, Daten an Mitglieder einer bestimmten Gruppe zu senden. Jedoch erfordert Multicast zumeist ein aktives Gruppenmanagement. Im Entwurf von Multicast-Algorithmen werden desweiteren sowohl dichte Gruppen von topologisch sehr nahen Knoten als auch dünne, sehr weit verteilte Gruppen berücksichtigt. Mit dem Entwurf eines eigenen Geocast-Algorithmus können möglicherweise durch die Idee von Geocast gegebene implizite Lokalitätseigenschaften ausgenutzt werden.

In der vorliegenden Arbeit sollen mögliche Geocast-Routingalgorithmen für große Mesh-Netze erforscht werden. Zielgröße sind dabei Metropolitan Area Networks, also Netze, die ein ganzes Stadtgebiet abdecken können. Diese Netze umfassen dabei sowohl Indoor- als auch Outdoor-Versorgung. Die entworfenen Routingalgorithmen sollen unter Verwendung der zu erwartenden Lokalitätseigenschaften optimiert werden. Das Hauptziel der Optimierungen von Algorithmen für die Anwendung in Mesh-Netzwerken ist die Verringerung des Nachrichtenoverhead, um so die knappste Ressource in drahtlosen Netzen, die verfügbare Bandbreite zur Übertragung von Nutzdaten, zu schonen.

Als zugrunde liegende Mesh-Netze werden Netze mit zwei verschiedenen Unicast-Routingalgorithmen betrachtet werden. Netze mit Optimized Link State Routing (OLSR) repräsentieren die aktuell in einigen Großstädten bestehenden freien Mesh-Netze. Auf Ad-Hoc On-Demand Distance Vector Routing (AODV) aufsetzende Netze sind ein Prototyp für noch im Standardisierungsprozess befindliche IEEE 802.11s Mesh-Netze.

Neben der Entwicklung von Geocast-Algorithmen werden mit den aussichtsreichsten Ansätzen auch Simulationen zur Vorhersage der Leistungsfähigkeit solcher Netze durchgeführt.

1.3 Gliederung der Arbeit

In Kapitel 2 wird grundlegendes Wissen über das Themengebiet zusammengefasst sowie ein Überblick über die bisherige Forschung und aktuelle Entwicklungen gegeben. Das Systemmodell, in dessen Kontext die weiteren Überlegungen stattfinden, wird in Kapitel 3 aufgestellt. Die im Rahmen dieser Arbeit entwickelten Ideen führt Kapitel 4 aus, während Kapitel 5 auf die konkrete Umsetzung in einer Simulation eingeht. Schließlich werden Simulationsergebnisse in Kapitel 6 aufgezeigt und analysiert. Kapitel 7 bietet eine abschließende Zusammenfassung dieser Arbeit und Ausblick auf noch zu bearbeitende Themen.

2 Grundlagen und verwandte Arbeiten

Dieses Kapitel gibt einen Überblick über die der weiteren Arbeit zugrunde liegenden Protokolle und Technologien. Dies umfasst eine Zusammenstellung der verwendeten Unicast-Algorithmen und ihrer jeweiligen Multicast-Erweiterungen. Ebenso werden bereits bestehende Geocast-Varianten und -Routingprotokolle untersucht. Ein Überblick über verschiedene Routing-Metriken schließt das Kapitel ab.

2.1 Grundlagen zum Routing

Um automatisiert Routen in Netzwerken zu bestimmen, gibt es verschiedene Ansätze. Die meistgenutzten sind Link-State- und Distanz-Vektor-Verfahren.

Bei Link-State-Verfahren wird die Topologie des gesamten Netzes bekannt gemacht, d.h. eine globale Sicht auf das Netz wird hergestellt. Daraus können für alle Knoten die besten Routen zu jedem Ziel errechnet werden.

Bei Distanz-Vektor-Verfahren arbeitet jeder Knoten nur mit einer lokalen Sicht. Jeder Knoten teilt seinen direkten Nachbarn mit, welche Ziele über ihn zu erreichen sind. Anhand der übermittelten Daten über die Qualität der Weiterleitung nach einer gewählten Metrik kann ein Knoten entscheiden, über welchen seiner Nachbarn er ein Ziel kontaktieren möchte.

In festverkabelten Netzwerken werden üblicherweise Adressen hierarchisch entsprechend der Netztopologie verteilt, um durch Präfix-Routing die Routingtabellen klein halten zu können und die auszutauschenden Informationen zu minimieren. In Mesh-Netzen ist dies aufgrund der Fluktuationen im Netz nicht möglich. Eine Adresse, die an einem Ende des Netzes vergeben wurde, kann von einem mobilen Knoten ans andere Ende des Netzes mitgenommen werden. Routingprotokolle in Mesh-Netzen müssen daher jeden einzelnen Knoten im gesamten Netz auffinden können.

2.2 Einordnung in das Schichtenmodell

Geocast-Routing ist genau wie Unicast- oder Multicast-Routing ein Ansatz, der gemäß dem OSI-Schichtenmodell auf der Vermittlungsschicht (Schicht 3) stattfinden sollte. In Mesh-Netzen ergibt sich jedoch die Besonderheit, dass das Routing teilweise auch auf der Sicherungsschicht (Schicht 2) stattfindet, um für die Vermittlungsschicht das Konzept einer großen Broadcast-Domäne bereitzustellen. In einem verkabelten Netzwerk entspricht dies

dem Einsatz von Switchen zum Anbinden neuer Netzwerksegmente im Gegensatz zum Aufstellen von Routern.

Die Entscheidung, ob ein Mesh-Netz das Routing auf Schicht 2 oder 3 durchführt, liegt letztlich beim Hersteller der Komponenten oder auch beim Administrator des Netzwerkes. Die Mesh-Routing-Protokolle AODV und OLSR (siehe Abschnitt 2.3.1, Ad-Hoc On-Demand Distance Vector Protokoll und Abschnitt 2.3.2, Optimized Link State Routing) wurden bereits in beiden Varianten präsentiert [Bah06]. IEEE 802.11s-Netzwerke werden voraussichtlich auf einer Schicht-2-Variante von AODV aufsetzen [Wik09], während Freifunk-Netzwerke derzeit Routing mit OLSR auf Schicht 3 betreiben [Pro09].

Ebenso wie Unicast-Routing kann auch Geocast-Routing in Mesh-Netzwerken auf Schicht 2 oder 3 durchgeführt werden. Für den Entwurf von Geocast-Routingprotokollen ist es unerheblich, auf welcher Schicht das Routing später stattfindet. Implementierungen sollten jedoch auf der gleichen Schicht wie das zugrunde liegende Unicast-Routing arbeiten. Da die Ur-Versionen von AODV und OLSR als Schicht-3-Varianten vorgestellt wurden, kann im Folgenden davon ausgegangen werden, dass auch das Geocast-Routing auf Schicht 3 stattfindet.

2.3 Zugrunde liegende Routingprotokolle

In der Vergangenheit wurden mehrere Protokolle zum Austausch von Routinginformationen entwickelt. In dieser Arbeit sollen allerdings nur einige wenige Protokolle betrachtet werden. Die Auswahl wurde durch den IEEE 802.11s-Standard für Mesh-Netze und das Freifunk Projekt [Pro09] beeinflusst.

2.3.1 Ad-Hoc On-Demand Distance Vector Protokoll

Im aktuellen Entwurf zum IEEE-Standard 802.11s für Mesh-Netze ist für alle Knoten das Hybrid Wireless Mesh Protocol (HWMP) vorgesehen [Wik09]. HWMP ist ein Routingprotokoll auf der Sicherungsschicht und basiert auf dem Ad-Hoc On-Demand Distance Vector Protokoll (AODV, [PR99]). Da der Standard kurz vor der Verabschiedung steht, ist damit zu rechnen, dass sich Implementierungen in einigen Jahren in kommerziell erhältlichen Geräten finden werden. Untersuchungen in dieser Arbeit auf Basis von AODV können daher als Grundlage verwendet werden, um später diese Geräte um Geocast-Funktionen zu erweitern.

Nach der Spezifikation von AODV werden Routing-Informationen ausschließlich bei Bedarf ausgetauscht und errechnet. Falls keine Nutzdaten durch das Netz geleitet werden sollen, werden lediglich Nachrichten zum Erkennen der direkten Nachbarschaft ausgetauscht. AODV geht von ausschließlich bidirektionalen Verbindungen zwischen den Knoten aus.

Die Erkennung der Nachbarschaft funktioniert durch das periodische Versenden von `hello`-Nachrichten. Diese werden von empfangenden Knoten nicht weitergeleitet, jedoch erkennen

Knoten somit, wer sich in ihrer direkten Nachbarschaft befindet. Bleibt eine `hello`-Nachricht über mehrere Periodendauern aus, wird davon ausgegangen, dass sich der betroffene Knoten nicht mehr in der direkten Nachbarschaft befindet.

Soll ein Paket verschickt werden, sendet der Quellknoten zunächst einen Route-Request (`RREQ`) per Broadcast aus. Empfängt ein Knoten einen `RREQ` und ist er nicht der Zielknoten, so versendet er den `RREQ` per Unicast an den Zielknoten, falls er eine Route dorthin kennt. Andernfalls wird der `RREQ` per Broadcast weitergeleitet. Ist der Empfänger der Zielknoten, so sendet er ein Route-Reply (`RREP`) per Unicast an den Absender zurück. Jeder Knoten, der ein `RREQ` oder `RREP` empfängt, merkt sich denjenigen Knoten, von dem er das Paket empfangen hat, als nächsten Hop auf der Unicast-Route zur Quelle des `RREQ` bzw. `RREP`. Routen zu entfernten Zielen werden somit als „Reverse Pointer" zu benachbarten Knoten gelernt.

Sequenznummern in allen Paketen sorgen dafür, dass errechnete Routing-Informationen aktuell sind und keine Routing-Schleifen entstehen können. Errechnete Routen und Reverse Pointer, die nicht verwendet werden, werden nach dem Soft-State-Prinzip nach einiger Zeit wieder gelöscht.

Um mit Topologieänderungen umgehen zu können, wird bei einem Verbindungsabbruch der Quellknoten (der Absender des initialen `RREQ`) mit einem speziellen `RREP` darüber informiert, dass eine von ihm verwendete Route nicht länger verfügbar ist. Der Quellknoten kann daraufhin einen neuen `RREQ` senden um eine neue Route errechnen zu lassen.

In Simulationen zeigte sich, dass sich die Leistungswerte von AODV zwar mit der Netzwerkgröße verschlechtern, insgesamt jedoch auch bei mehreren hundert Knoten noch ein funktionierendes Netzwerk existiert. Die auf der AODV-Varainte HWMP aufbauenden IEEE 802.11s-Netze sind dagegen nur für eine Zielgröße von etwa 60 Knoten entworfen.

2.3.2 Optimized Link State Routing

Optimized Link State Routing (OLSR) wurde in einer Schicht-2-Variante (RA-OLSR) in früheren Entwürfen des 802.11s-Standards als optionales weiteres Routing-Protokoll vorgeschlagen, wurde jedoch in neueren Entwürfen aus dem Standard entfernt [Wik09]. Die Funktionsweise des ursprünglichen OLSR ist in RFC 3626 [CJ03] und in [JMC+01] definiert.

OLSR-Knoten senden periodisch eine `hello`-Nachricht an alle direkten Nachbarn. In dieser Nachricht ist die Liste der bisher bekannten direkten Nachbarn enthalten. Somit kennt jeder Knoten, der eine `hello`-Nachricht von jedem seiner Nachbarn empfangen hat, alle anderen Knoten, die maximal zwei Hops weit entfernt sind. Gleichzeitig kann über die `hello`-Nachrichten geprüft werden, ob eine bidirektionale Verbindung zwischen zwei direkten Nachbarn existert. Findet ein Knoten seine eigene Adresse in der Nachbarschaftsliste der `hello`-Nachricht eines anderen Knotens wieder, existiert zwischen diesen beiden Knoten eine bidirektionale Verbindung.

Mit diesem Wissen über seine Nachbarschaft wählt ein Knoten seine Multipoint-Relays (MPR) aus. MPRs werden aus den direkt benachbarten Knoten so ausgewählt, dass eine Nachricht, die vom Knoten an alle MPRs gesendet und von diesen wiederum an alle ihre direkten

Nachbarn verteilt wird, alle Knoten in einer Entfernung von zwei Hops erreicht. Jeder Knoten, der als MPR ausgewählt wurde, sendet periodisch Topology Control Nachrichten (`TC`). Diese beinhalten die Adressen aller Knoten, die diesen Knoten als MPR ausgewählt haben, und werden durch das Netz geflutet. Die Tupel (Knotenadresse, MPR-Adresse) definieren die Verbindungen, die im Netz zur Verfügung stehen. Jeder Knoten bewahrt die neuesten `TC`-Nachrichten jedes anderen Knotens auf. Wird eine Route zu einer Zieladresse gesucht, kann ein Knoten mit einem Shortest-Path-Algorithmus (z.B. Dijkstra) den kürzesten Weg zwischen sich und dem Ziel errechnen und damit bestimmen, an welchen seiner direkten Nachbarn ein Paket weitergeleitet werden soll.

Die Aktualität von Nachrichten wird durch in jedem Paket enthaltene, vom Paketerzeuger vergebene Sequenznummern sichergestellt. Nach dem Soft-State-Prinzip werden Nachbarknoten und Routen nach einiger Zeit vergessen, wenn sie nicht mehr in `hello`- bzw. `TC`-Nachrichten erscheinen.

Im Freifunk-Projekt [Pro09] konnten praktische Erfahrungen mit OLSR gesammelt werden. Im Praxiseinsatz erwies sich jedoch die hier vorgestellte RFC-konforme Version von OLSR als völlig untauglich. OLSR leidet unter zwei Arten von Problemen:

- OLSR nach RFC3626 leidet unter konzeptionellen Schwächen, die Verbindungen durch ein Mesh-Netz sehr instabil machen [Aic]. Multipoint-Relays (MPR) senken die Redundanz. Der eigentlich zur Stabilisierung von Routen eingesetzte Hysterese-Mechanismus reichte nicht aus, um oszillierende Routen zu vermeiden. Häufige Routing-Schleifen waren die Folge.
- Die ersten Implementierungen des Routing-Daemons OLSRd hatten massive Performance-Probleme, die insbesondere die Skalierbarkeit auf Access Points stark beeinträchtigt haben – die maximal erreichbaren Netzgrößen lagen bei etwa 100 Knoten. Im Rahmen des OLSR-NG-Projektes [TLG+08] konnte jedoch durch einen Rewrite des Routing-Daemons die Performance derart verbessert werden, dass nun auch Routen in größeren Netzen auf Geräten mit geringer Rechenleistung, wie z.B. Access Points, schnell berechnet werden können.

Diese Entdeckungen beruhen auf subjektiven Beobachtungen der aufgebauten Netze und sind bisher nicht durch wissenschaftliche Forschungen untermauert worden. Verbesserungen an den Protokollen unterliegen keinem wissenschaftlichen Entwicklungsprozess mehr, sondern werden sehr pragmatisch von den OLSRd-Entwicklern erstellt und in realen Systemen getestet.

In der weiteren Entwicklung des OLSR-Routing-Daemons wurden konzeptionelle Änderungen eingebracht. So können nun MPRs und Hysterese deaktiviert werden. Die Hop-Count-Metrik kann durch eine ETX-Metrik (Expected Transmission Count, siehe Abschnitt 2.6.2) ersetzt werden. Das Fluten des Netzes kann eingeschränkt werden: Mit dem Fish-Eye-Mechanismus werden häufig `TC`-Nachrichten in der näheren Umgebung eines Knotens ausgetauscht, während nur relativ selten `TC`-Nachrichten eines Knotens durch das gesamte Netz propagiert werden.

Diese Anpassungen gingen später im OLSR-NG-Projekt, dem Rewrite des OLSR-Routing-Daemons, auf. Welche Variante des Algorithmus verwendet werden soll, kann durch Konfigurationsoptionen eingestellt werden. Dabei kann je nach Wahl der Parameter auch die Kompatibilität zu RFC3626 verloren gehen. Es existiert jedoch bereits ein neuer Entwurf für einen Standard [CD09], der keinen Algorithmus zur Auswahl der Routen mehr spezifiziert und den Fish-Eye-Mechanismus erlaubt. Durch diese zusätzliche Freiheit wird es möglich, dass Implementierungen verbesserte Algorithmen zur Routenberechnung verwenden können. Am Konzept der Multi-Point-Relays und der Hop-Count-Metrik wird jedoch weiter festgehalten.

2.3.3 Better Approach To Mobile Ad-Hoc Networking (B.A.T.M.A.N.)

Better Approach To Mobile Ad-Hoc Networking (B.A.T.M.A.N.) wurde entwickelt, um OLSR in praktischen Anwendungen zu ersetzen. Analog zum RFC-konformem Verhalten, das Fluten des Netzes mit Hilfe von MPRs zu optimieren, versucht auch B.A.T.M.A.N. eine solche Optimierung.

Das Protokoll basiert auf dem Fluten des Netzes mit Originator Messages (`OGM`, [NAL07, NALS08]). Jeder Knoten speichert sich, von welchem Nachbarn er wieviele `OGM`s gesehen hat. Derjenige Nachbar, von dem die meisten `OGM`s empfangen wurden, zählt als „bester Nachbar“ und wird zum Routing von Paketen an andere Knoten verwendet. Ebenso werden nur `OGM`s, die von diesem „besten Nachbarn“ empfangen wurden, erneut verteilt.

Bisher gibt es keine wissenschaftlichen Untersuchungen zur Performanz dieses Protokolls. Die Entwickler selbst haben lediglich von stark schwankenden subjektiven Eindrücken der Nutzer berichtet. Da B.A.T.M.A.N. sich noch in einer relativ frühen Entwicklungsphase befindet, ist noch mit einigen Änderungen am Protokoll zu rechnen. Während die Operationsweise im Kern bisher unangetastet blieb, hat sich eine Reihe von Optimierungen ergeben. Solche zukünftigen Änderungen können durchaus noch so gravierend werden, dass sie eine neue Entwicklung des Geocast-Protokolles notwendig machen. Daher wird B.A.T.M.A.N. in dieser Arbeit keine Beachtung finden.

2.4 Multicast-Routing

Ebenso wie Unicast-Routingprotokolle müssen Multicast-Verfahren an die Eigenheiten von Mesh-Netzen, insbesondere die häufigen Topologieänderungen, angepasst werden. Aufgrund der unterschiedlichen vorgeschlagenen Unicast-Routingkonzepte ist es kaum möglich, generische Multicast-Routingprotokolle zu entwickeln, die mit allen Unicast-Routingprotokollen effizient zusammenarbeiten. Daher werden im folgenden zwei Multicast-Protokolle vorgestellt, die vollständig an AODV bzw. OLSR angepasst sind.

2.4.1 Multicast mit AODV

AODV bietet direkte Unterstützung für Shared-Tree-Multicast [RP99, RP00]. Zu jeder aktiven Multicast-Gruppe existiert ein Gruppenführer, der als Rendezvous-Punkt dient. Eine Multicast-Gruppe ist aktiv, wenn gerade Nachrichten an die Gruppe gesendet werden. Jeder Knoten kennt den Rendezvous-Punkt jeder aktiven Gruppe sowie eine Route und die Entfernung dorthin. Weiterhin weiß jeder Knoten, welche seiner Nachbarn an Daten dieser Multicast-Gruppe interessiert sind.

Empfängt ein Knoten Multicast-Nachrichten für eine Gruppe, so leitet er die Nachricht an alle seine direkten Nachbarn, die ebenfalls Mitglied der Gruppe sind, weiter. Dabei wird die Route zum Rendezvous-Punkt auch berücksichtigt; jedoch wird die Nachricht nicht zu demjenigen Nachbarn zurückgeschickt, von dem der Knoten die Nachricht erhalten hat.

Um eine Multicast-Gruppe zu eröffnen, ihr beizutreten oder Nachrichten an die Gruppe schicken zu können, versendet ein Knoten einen `RREQ` (Route Request) an die Gruppenadresse. Dieser wird durch das Netz geflutet, bis ein Mitglied der Gruppe diesen Request empfängt. Jedes Gruppenmitglied, das den `RREQ` empfängt, wird einen `RREP` (Route Reply) als Antwort darauf generieren und diesen an den anfragenden Knoten zurücksenden. Jeder Knoten, der einen `RREQ` weiterleitet, notiert sich die Route zum anfragenden Knoten. Jeder Knoten, der einen `RREP` weiterleitet, notiert sich die Route zum nächsten Knoten der Multicast-Gruppe und markiert diese zunächst als inaktiv. Im Allgemeinen wird ein anfragender Knoten mehrere `RREP`s erhalten. Daher wird dieser Knoten eine Route, typischerweise die mit der geringsten Anzahl an Hops, auswählen und ein Multicast-Activation-Paket (`MACT`) an den nächsten Hop auf dieser Route senden. Wenn ein Knoten ein `MACT` empfängt, aktiviert er die Route zur in der Nachricht angegebenen Multicast-Gruppe und versendet seinerseits ein `MACT` an den nächsten Knoten auf seiner Route zur Gruppe. Sobald das `MACT` einen Knoten erreicht, der bereits Mitglied der gewünschten Gruppe ist, werden Daten weitergeleitet.

Falls die Gruppe noch nicht existiert, wird der `RREP` unbeantwortet bleiben und der anfragende Knoten kann zum Gruppenführer der neu erstellten Gruppe werden.

Knoten können die Multicast-Gruppe verlassen, indem sie eine `MACT`-Nachricht mit gesetztem *prune*-Flag an ihren nächsten Hop in Richtung Gruppenführer schicken. Es können jedoch nur Blattknoten die Gruppe verlassen, da die Gruppe sonst partitioniert werden würde. Falls der Knoten, bei dem man sich abgemeldet hat, nun ein Blattknoten geworden ist und auch selbst nicht an einer Gruppenmitgliedschaft interessiert ist, kann dieser Knoten ebenfalls auf die gleiche Weise die Gruppe verlassen. Dies kann sich rekursiv fortsetzen, bis der Gruppenführer oder ein Nicht-Blattknoten erreicht wird.

Um die Aktualität von Nachrichten zu gewährleisten, werden dieselben Sequenznummern für `RREQ` und `RREP` eingesetzt wie beim Unicast-Routing. Jeder Multicast-Gruppe ist weiterhin ebenfalls eine Sequenznummer zugeordnet, die vom Gruppenführer verwaltet wird.

2.4.2 Multicast Optimized Link State Routing

Um Multicast in Mesh-Netzen auf Basis von OLSR zu ermöglichen, wurde das Multicast Optimized Link State Routing Protokoll (MOLSR) entworfen [LJM+03]. MOLSR erzeugt quellenbasierte Multicast-Bäume.

Für die Funktion von MOLSR ist es erforderlich, dass zumindest ein Teil der Knoten in einem Mesh-Netz als MOLSR-Router arbeitet. Es ist jedoch nicht notwendig, dass alle Knoten multicastfähig sind. Multicast-Router fluten das Netz periodisch mit Benachrichtigungen über ihre Existenz. Zusammen mit der von OLSR erzeugten globalen Sicht kann so jeder Multicast-Router ständig die optimalen Routen zu allen anderen Multicast-Knoten über andere Multicast-Router berechnen.

Möchte ein Knoten an eine Multicast-Gruppe senden, flutet er eine `SOURCE_CLAIM`-Nachricht, die ihn als neue Quelle für die gewünschte Multicast-Gruppe identifiziert, durch das Netz. Alle Multicast-Knoten, die an Nachrichten für die angegebene Gruppe interessiert sind, senden eine `PARENT_CONFIRM`-Nachricht zu ihrem nächsten Multicast-Router in Richtung der Quelle. Dieser Router wird dann wiederum eine Nachricht zu seinem nächsten Multicast-Router in Richtung der Quelle senden, falls er nicht schon Mitglied dieses Multicast-Baumes ist. Der Prozess setzt sich bis zur Quelle rekursiv fort. Jeder Router führt Buch über alle Empfänger, die sich bei ihm angemeldet haben.

Wiederum durch die globale Sicht von OLSR können Topologieänderungen im Netz bemerkt werden. In solchen Fällen werden auch die Multicast-Routen neu berechnet. Gegebenenfalls melden sich Multicast-Knoten bei anderen Elternknoten an, um den Multicast-Baum optimal zu halten.

Neueintritte in den Baum werden dadurch ermöglicht, dass Multicast-Quellen ihre `SOURCE_CLAIM`-Nachrichten regelmässig wiederholen und neue Multicast-Empfänger so feststellen können, welche Multicast-Bäume zu einer Gruppe bestehen. Mit dieser Information kann der Empfänger allen notwendigen Multicast-Bäumen beitreten.

Zum Verlassen eines Multicast-Baumes schickt ein Empfänger eine `LEAVE`-Nachricht aus, die dem zur Quelle nächsten Router anzeigt, dass der absendende Knoten keine weiteren Nachrichten aus diesem Baum wünscht. Der Router wird daraufhin diesen Empfänger bei sich abmelden. Falls keine weiteren Interessenten für Nachrichten aus dem fraglichen Multicast-Baum bekannt sind, wird sich der Router ebenfalls beim nächsten Router in Richtung Quelle abmelden. Dieser Prozess kann sich rekursiv fortsetzen.

2.5 Geocast-Kommunikation

Um Geocast-Nachrichten zu adressieren gibt es im wesentlichen zwei unterschiedliche Ansätze: Geographische Koordinaten oder symbolische Adressen. Die folgenden Abschnitte stellen beide Adressierungsarten vor. Im Anschluss daran werden für beide Adressierungsarten bereits existierende Algorithmen zum Geocast-Routing im Internet präsentiert.

2.5.1 Adressierungskonzepte & Lokationsmodelle

Geocast mit geographischen Koordinaten

Mit Hilfe von Geopositionierungssystemen wie GPS ist es zumindest außerhalb geschlossener Räume für mobile Geräte sehr leicht möglich, ihre aktuellen geographischen Koordinaten auf wenige Meter genau zu bestimmen. In Gebäuden kann eine raumgenaue Bestimmung der Koordinaten mit Hilfe von Kurzstreckenfunksystemen stattfinden. Diese Koordinaten können für eine Geocast-Adressierung bereits ausreichend sein: Es gibt Modelle, in denen ein Anwender auf einer Karte einen beliebigen Bereich markiert. Eine Geocast-Nachricht wird dann an alle Knoten innerhalb dieses Bereiches gesendet. Meist kommen dabei Kreise und Polygone als Primitive zum Einsatz [IN96, IN99, IN97, IN98, Nav01]. So kann mit ein paar geometrischen Rechenoperationen festgestellt werden, ob sich ein Gerät im Zielgebiet befindet oder nicht.

Geocast mit symbolischen Adressen

Im Gegensatz zu geometrischen Koordinaten, die eine sehr feingranulare Sicht der Welt bieten und eine Menge an geometrischen Rechenoperationen benötigen, wurden in [DR03, DR06, DR08] symbolische Geocast-Adressen definiert. Symbolische Geocast-Adressen bestehen aus einer hierarchischen Darstellung sich verfeinernder Ortsangaben. So adressiert beispielsweise `/earth/europe/de/bw/stuttgart/vaihingen/universitaetsstrasse/38/0.353` ein Computerlabor des Instituts für parallele und verteilte Systeme an der Universität Stuttgart. Ein Präfix kann einer Lokation z.B. über die Reichweite eines Funksenders oder einem RFID-Tag der von geocastfähigen Knoten ausgelesen wird, zugeordnet werden.

Die Granularitätsstufen, in denen ein Zielgebiet adressiert werden kann, hängen dabei allein von den vergebenen Namen ab. Im vorhergehenden Beispiel kann man so Nachrichten entweder an alle Clients richten, die sich derzeit im Gebäude Universitätsstrasse 38 aufhalten, oder man kann den Empfängerkreis auf einen einzelnen Raum einschränken. Es ist jedoch mit diesem Präfix nicht möglich, Nachrichten an alle Clients in einem Stockwerk zu senden. Dies ist kein Schwachpunkt des Adressformats, sondern der vergebenen Adressen. Wird der Präfix um Stockwerksangaben erweitert, z.B. `/earth/europe/de/bw/stuttgart/vaihingen/universitaetsstrasse/38/ground_floor/0.353`, so sind auch Adressierungen an einzelne Stockwerke möglich.

Lokationen ohne weitere untergeordnete Lokationen, d.h. Lokationen der feinsten Granularität, werden als Basislokation bezeichnet. Knoten befinden sich stets an Basislokationen. Übergeordnete Lokationen werden lediglich konzeptionell aus den Anfangsstücken der symbolischen Adressen der Basislokationen gebildet. Untergeordnete Lokationen sind stets vollständig in übergeordneten Lokationen enthalten. Daher werden die symbolischen Geocast-Adressen von Lokationen im Folgenden auch als Präfixe der jeweiligen (Basis-)Lokationen bezeichnet.

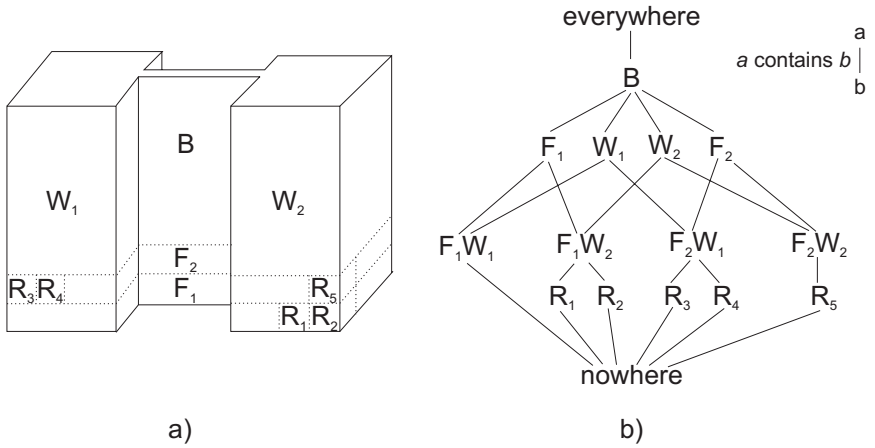


Abbildung 2.1: Beispiel für Lokationen, die sich nicht als Baum anordnen lassen. Links: Gebäude mit Stockwerken F_1, F_2, Flügeln W_1, W_2 und Räumen R_1 bis R_5. Rechts: Beziehungen zwischen den einzelnen Teilen in Netzform. Quelle: [DR03]

Alle geographischen Bereiche, die von einem gewissen Präfix abgedeckt werden, bilden gemeinsam die Geocast-Domäne dieses Präfixes. Anders gespochen ist die Geocast-Domäne eines Präfixes bzw. einer Lokation die Menge aller Knoten, die an Nachrichten für diesen Präfix interessiert sind.

In diesem Schema von Präfix-Adressen ergibt sich jedoch das Problem, dass Gebiete existieren, die sich nur teilweise überschneiden, sich jedoch nicht gegenseitig enthalten. Ein klassisches Beispiel ist ein Bürogebäude mit mehreren Flügeln und mehreren Stockwerken. Jedes Stockwerk erstreckt sich über mehrere Flügel, jedoch beinhaltet ein Flügel wiederum mehrere Stockwerke.

Um diesem Problem zu entgegnen kann die Hierarchie der Lokationen als Netz statt als Baum betrachtet werden. Dabei kann, wie in Abbildung 2.1 dargestellt, jede Lokation in mehr als einer anderen Lokation enthalten sein. Es ergibt sich jedoch das Problem der Notation. Prinzipiell lassen sich zwar auch bei der Betrachtungsweise als Netz Präfixe als hierarchische Pfade durch das Netz angeben. Jedoch sind dann einzelnen Orten gegebenenfalls mehrere Pfade zugeordnet, z.B. `/earth/europe/de/bw/stuttgart/vaihingen/universitaetsstrasse/38/ground_floor/0.353` und `/earth/europe/de/bw/stuttgart/vaihingen/universitaetsstrasse/38/south_side/0.353`. Wenn mehrere Komponenten austauschbar sind, ergibt sich eine kombinatorische Explosion der Anzahl von Präfixen einer einzigen Lokation. Neben möglicherweise zusätzlichem Rechenzeit- und Speicheraufwand beim Routing ist auch die Fragestellung zu lösen, wie Clients alle diese Namen bekannt gemacht werden können.

In [DR06] wird eine weitere Adressschicht eingeführt. Orten wird eine eindeutige binäre Adresse so zugewiesen, dass Präfixe der binären Adresse tatsächlich vollständig umgebende

Bereiche bilden. Über einen Verzeichnisdienst, z.B. DNS, werden die allgemeinverständlichen Namen in binäre Adressen umgesetzt. Dieses Verfahren ist jedoch nur bedingt geeignet, um das obige Problem zu lösen. Um der Bedingung zu genügen, dass Anfangsstücke von Präfixen Lokationen bezeichnen, die das fragliche Gebiet vollständig einschließen, muss die Lokationshierarchie wieder als Baum betrachtet werden. Damit muss ein symbolischer Name auf mehrere binäre Adressen abgebildet werden, d.h. der Sender muss mehrere Geocasts an mehrere binäre Adressen versenden, um einen Geocast an eine symbolische Adresse zu versenden.

2.5.2 Vermittlungsalgorithmen

Geographisches Routing

Für die Realisierung von Routing in Geocast-Systemen mit geographischen Koordinaten wurde ein zweistufiges Routing vorgeschlagen. Die Stufen sind hierbei das Routing im Backbone-Netz und das Routing zum Endgerät. Im Backbone-Netz existiert eine baumförmige Hierarchie aus Geocast-Routern, die über Unicast miteinander kommunizieren. Die Blätter des Baumes sind GeoNodes: Knoten, die die Übertragung zu den Endgeräten in einem begrenzten geographischen Gebiet vornehmen können. Mit Hilfe von erweiterten Unicast-Routingprotokollen (z.B. GeoRIP oder GeoOSPF) werden außer Unicast-Routinginformationen auch Informationen über die von jedem Knoten versorgten Gebiete verteilt. So können auch Geocast-Router errechnen, für welches geographische Gebiet ihre Kinder zuständig sind. Durch Vergleich der Zuständigkeitsgebiete mit dem Zielgebiet einer Nachricht kann jeder Router feststellen, ob er die Nachricht an seinen Vater oder an eines oder mehrere seiner Kinder weiterleiten muss.

Kommt eine Nachricht bei einer GeoNode an, wird sie dort zwischengelagert und der Versand vorbereitet. Für jedes in einer Nachricht adressierte Gebiet reserviert die GeoNode eine Multicast-Gruppe. Die Information über den bevorstehenden Geocast und sein Zielgebiet wird zusammen mit der vorgesehenen Multicast-Gruppe per Broadcast an alle Knoten in Reichweite des Georouters verteilt. Die Knoten treten daraufhin, so sie sich im Zielgebiet befinden, der Multicast-Gruppe bei. Anschließend wird die Geocast-Nachricht per Multicast lokal verteilt. Zuletzt verlassen alle Knoten die Multicast-Gruppe wieder.

Routing von Geocast mit symbolischen Adressen

Um symbolisch adressierte Geocast-Nachrichten im Internet effizient routen zu können, eignet sich ebenfalls ein baumbasierter Ansatz [DR06, DR08]. Dabei wird zunächst jeder Lokation ein Designated Router (DR) zugewiesen. DRs kennen jeweils alle Router, deren Präfix um eine Komponente genauer oder ungenauer ist. D.h. ein DR kennt den DR der direkt übergeordneten Lokation und alle DRs der direkt untergeordneten Lokationen. Weiterhin kann jeder Blatt-DR alle Knoten in seinem Zuständigkeitsbereicht, d.h. an seiner Lokation, erreichen.

Geocast-Nachrichten werden über diesen Baum durch das Netz geroutet. Sie werden zunächst zum DR des gesamten Zielgebietes geleitet. Dieser gibt die Nachricht an seine untergeordneten DRs weiter, die die Nachricht wiederum bis zu den untersten DRs propagieren. Dort wird die Nachricht an die Endknoten verteilt.

In dieser baumförmigen Architektur müssen potentiell sehr viele Nachrichten über die Wurzel geleitet werden. Um die Wurzel zu entlasten und die Gesamtperformance des Systems zu verbessern, bauen DRs Abkürzungen zu anderen DRs in anderen Zweigen des Baumes auf, sodass sich der Baum zum Graph ausweitet. Damit ein schleifenfreies Routing gewährleistet wird, muss das Weiterleiten von Nachrichten in zwei Phasen aufgeteilt werden. In Phase 1 steigt die Nachricht im Baum und über die gewählten Abkürzungen auf, bis der DR des Zielgebietes oder die Wurzel des Baumes erreicht wurden. In Phase 2 sinkt die Nachricht durch den Baum zu den Endknoten ab. Abkürzungen dürfen in Phase 2 nicht verwendet werden.

2.6 Metriken für Routing-Algorithmen

Verschiedene Routing-Ansätze verwenden unterschiedliche Metriken zur Bewertung von Routen. Je nach Einsatzszenario eignen sich auch für einen Routingansatz unterschiedliche Metriken. Im Folgenden soll ein kurzer Überblick über die in den verschiedenen Routingprotokollen verwendeten Metriken gegeben werden.

2.6.1 Hopcount

Die Hopcount-Metrik ist die einfachste Metrik für Netzwerke aller Art. Die Anzahl der Hops (Hopcount) zwischen zwei Netzwerkknoten A und B ist hierbei die Anzahl der Verbindungen zwischen benachbarten Knoten, die ein Paket von A nach B passiert. Ein kleinerer Hopcount gilt als besserer Wert.

Wird zum Bestimmen von guten Routen durch ein Netzwerk die Hopcount-Metrik angewendet, wird nur auf einen kurzen Pfad hin optimiert. Eigenschaften wie Latenz und Bandbreite oder aktuelle Auslastung einer Leitung gehen nicht in die Routingentscheidung mit ein. Wenn also beispielsweise eine Verbindung mit geringer Bandbreite und hoher Latenz zwischen A und B besteht, parallel dazu jedoch eine Verbindung mit sehr hoher Bandbreite und geringerer Latenz über einen Zwischenknoten C, so wird ein Routingprotokoll mit Hopcount-Metrik stets die schmalbandigere Verbindung bevorzugen.

Diese Metrik wurde in der ursprünglichen Variante von Optimized Link State Routing vorgeschlagen, hat sich jedoch in der Praxis nicht bewährt (siehe Abschnitt 2.3.2). AODV verwendet noch immer die Hopcount-Metrik.

2.6.2 Linkqualität

Um auch schnellere, aber längere Verbindungen nutzen zu können, kann eine Metrik verwendet werden, die die Qualität jeder einzelnen Verbindung berücksichtigt.

Expected Transmission Count (ETX, [DC04]) versucht abzuschätzen, wie viele Neuübertragungen notwendig sind, bis ein Paket erfolgreich über eine Verbindung übertragen wurde. ETX bewertet Routen gut, die einen hohen Durchsatz ermöglichen. Die Latenz von Übertragungen spielt in dieser Metrik keine Rolle.

Um den ETX-Wert zu bestimmen, muss zunächst die Qualität jeder Verbindung zwischen zwei direkt benachbarten Knoten bestimmt werden. Dazu versendet jeder Knoten mit global konstanter Periode eine `Probe`-Nachricht per Broadcast. Alle empfangenen `Probe`-Nachrichten werden gespeichert. Aufgrund der global konstanten Periode und der Anzahl der empfangenen `Probe`-Nachrichten kann nun jeder Knoten ausrechnen, wie wiele Nachrichten er von jedem anderen Knoten hätte empfangen müssen und wieviele verloren gegangen sind. Damit ist die Qualität bzw. die Verlustwahrscheinlichkeit für jede Verbindung in jeder Richtung bekannt. Der Austausch dieser Informationen zwischen den Knoten, um so jedem Knoten Wissen über seine Verbindungen in beide Richtungen zukommen zu lassen, ist Sache des Routing-Protokolles.

Falls ein Schicht-2-Protokoll mit zuverlässiger Übertragung eingesetzt wird, können auch die dafür notwendigen Acknowledgement-Nachrichten verwendet werden um die Linkqualität direkt zu messen. Diese Variante steht auch als Erweiterung für das Optimized Link State Routing zur Verfügung (siehe Abschnitt 2.3.2, [Lop04]).

Praktische Erfahrungen in Mesh-Netzen haben gezeigt, dass Linkqualität als Metrik in solchen Netzen nur bedingt geeignet ist [Aic]. In einem unbelasteten Netz ergeben sich sehr gute Routen. Unter Last treten jedoch vermehrt Paketverluste und -verzögerungen durch Kollisionen mit Nutzdatenpaketen auf, sodass eine eigentlich sehr gute Route plötzlich viele Links mit schlechter Linkqualität enthält. Die führt dazu, dass eigentlich schlechtere Routen plötzlich bevorzugt werden und der Datenverkehr umgeleitet wird. Dann wird wiederum auf der neuen Route eine schlechtere Qualität gemessen und die Route wird zurückverlegt. Eine Verschlechterung der Verfügbarkeit und des Datendurchsatzes ist die Folge.

2.6.3 Implizite Metriken

Für das B.A.T.M.A.N.-Protokoll (siehe Abschnitt 2.3.3) wurde eine neue, implizite Metrik entwickelt. Jeder Knoten flutet periodisch Originator Messages (`OGM`) durch das Netz. Weiterhin speichert sich jeder Knoten, wie viele `OGM`s von welchem Knoten er über welchen seiner direkten Nachbarn empfangen hat – dabei wird jeweils nur der erste Empfang einer `OGM` gewertet, d.h. Duplikate werden ignoriert. Der Nachbar, der in einem gewissen Zeitfenster die meisten `OGM`s weitergeleitet hat, wird als Route zu diesem Knoten gewählt. Die Idee dahinter ist, dass sich `OGM`s auf „guten“ Routen schneller ausbreiten werden als auf „schlechten“ und damit immer die erste ankommende `OGM` die derzeit beste Route markiert.

3 Systemmodell & Anforderungen

In diesem Kapitel wird der Rahmen für die zu entwickelnden Routingprotokolle gesetzt. Dazu untersucht Abschnitt 3.1 die technischen und konzeptionellen Eigenschaften der zugrundeliegenden Netzwerke. Abschnitt 3.2 stellt die Anforderungen und Optimierungsmetriken zusammen, auf die bei der Entwicklung der Protokolle ein besonderes Augenmerk gelegt werden soll.

3.1 Systemmodell

In dieser Arbeit werden größere Mesh-Netze betrachtet, die etwa einen Universitätscampus oder auch eine ganze Stadt abdecken können. Solche Netze bestehen typischerweise aus mehreren hundert bis mehreren tausend Knoten, die sich sowohl innerhalb als auch außerhalb von Gebäuden befinden können. Somit werden Eigenschaften von Indoor- und Outdoor-Anwendungen vereint.

Mesh-Knoten sind sowohl Mesh-Router (im folgenden auch einfach als „Router“ bezeichnet) als auch Mesh-Clients (im folgenden „Clients“). Mesh-Router sind ortsfest aufgestellte Geräte mit einer Energieversorgung über das Stromnetz. Sie bilden ein verbundenes Netz, d.h. falls kein Mesh-Router ausfällt, kann jeder Router jeden anderen im Netz ohne weitere Hilfsmittel erreichen. Mesh-Clients hingegen sind mobile Geräte mit lediglich begrenzter, jedoch unabhängiger Energieversorgung. Die Clients können auch Routing-Funktionen im Mesh-Netz übernehmen und so z.B. zeitweise die Reichweite des Netzes erhöhen, den Durchsatz des Netzes in ihrer Umgebung verbessern oder auch Ausfälle von Routern kompensieren.

Ein oder mehrere Mesh-Router können an einen Upstream-Provider zum Internet angeschlossen sein.

Als Vorbild für das Systemmodell dienen das an der Universität Stuttgart im Fachbereich Informatik installierte Mesh-Netzwerk sowie das von Google Inc. betriebene Mesh-Netz zur Internetversorgung von Mountain View, CA, USA.

3.1.1 Konnektivität

Die Konnektivität von Mesh-Routern untereinander ändert sich nur selten. Da die Mesh-Router ortsfest sind, können nur Änderungen in der Umgebung Verbindungsänderungen hervorrufen. Solche Änderungen könnten sein:

- Bau neuer oder Abriss alter Gebäude
- Öffnen oder Schließen von Fenstern und Türen
- Vorübergehende Abschattung durch Fahrzeuge oder Personen
- Permanente Abschattung durch Bäume
- Störeinflüsse durch Schlechtwetterlagen
- Störeinflüsse durch andere Geräte (z.B. Radar, Mikrowellen, drahtlose Gebäudeautomatisierung, RC-Steuerungen, Bluetooth-Geräte etc.)

Die Konnektivität zwischen Mesh-Routern und -Clients oder von Mesh-Clients untereinander ist dagegen sehr viel instabiler. Da die Clients potentiell mobil sind, kann sich ihre Nachbarschaft ständig ändern. Ebenso können Clients das Netz jederzeit und an jeder Stelle betreten oder verlassen.

3.1.2 Fehlerfälle

In Mesh-Netzwerken gibt es eine Vielzahl an möglichen Fehlerfällen wie z.B. Kollisionen von Nachrichten oder andere Störeinflüsse. Diese werden jedoch zumeist schon auf den unteren Schichten des Netzwerkes behandelt. Speziell für das Geocast-Routing sind lediglich Knotenausfälle relevant.

Ausfälle von Clients sind zwar möglich, werden jedoch bereits durch deren Beitreten und Verlassen des Netzes sowie deren Mobilität abgedeckt. Da davon ausgegangen wird, dass die Mesh-Router selbst bereits ein zusammenhängendes Netz bilden, kann ein Client eine Netzpartitionierung nicht verursachen.

Knotenfehler von Mesh-Routern sind stets transient: Falls ein Router ausfällt, ist damit zu rechnen, dass der Betreiber des Mesh-Netzes in endlicher Zeit für Ersatz sorgen wird, um die volle Funktionalität des Netzes wieder herzustellen. Bei einem Ausfall kann die Erreichbarkeit bestimmter Orte eingeschränkt werden. In Extremfällen kann es zu einer Partitionierung des Netzes kommen.

Netzpartitionierungen können nicht mit Mitteln des Netzwerkes selbst behoben werden. Es müssen stets von Benutzern oder Betreibern Knoten so platziert werden, dass die Kommunikation zwischen den Paritionen wieder möglich wird. Weiterhin kann es vorkommen, dass Geocast-Domänen partitioniert werden, ohne dass das gesamte Netz partitioniert wurde. Aufgrund einer angenommenen hohen Dichte von Knoten in Gebieten, in denen sich typischerweise Empfänger von Geocast-Nachrichten aufhalten, ist davon auszugehen, dass dieser Fall nur äusserst selten auftritt.

Durch die Erhöhung der Reichweite des Netzes durch Clients können sich Arme aus dem durch die Infrastruktur abgedeckten Bereich herausbilden, in denen noch immer Zugang zum betrachteten Mesh-Netz möglich ist. Ein mögliches Beispiel hierfür ist eine nahe an der Stadt vorbeiführende Autobahn, auf der viele meshfähige Fahrzeuge verkehren. Verlassen genügend Knoten an der Wurzel des Armes das Netz, entsteht ein durch Knoten

partitioniertes Netz. Dies ist jedoch für die vorliegende Arbeit ohne Bedeutung: Da im Normalfall nur ein begrenzter Bereich durch das Netz abgedeckt wird, ist nicht zu erwarten, dass Orte außerhalb dieses Bereiches per Geocast direkt durch das Netz angesprochen werden sollen. Geocast-Nachrichten an solche Orte müssten typischerweise durch einen Upstream-Provider an ein anderes Netz geschickt werden, das die Verteilung am gewünschten Ort übernimmt.

3.1.3 Lokationsinformationen

Adressformat

Es wird von einem baumförmigen Modell ausgegangen. Lokationsinformationen werden hierarchisch dargestellt (siehe [DR03, DR08, DR06] und Abschnitt 2.5.1). Dabei schließt jeder Präfix einer Lokation die Lokation vollständig ein. Die Namenskomponenten sind prinzipiell Freitext und werden durch Slashes („/“) getrennt. Präfixe werden als Strings abgelegt. Beispielsweise bezeichnet `/earth/europe/de/bw/stuttgart/vaihingen/universitaetsstrasse/38/0.353` einen der Laborräume im Informatikgebäude der Universität Stuttgart.

Ein geeignetes Verfahren zur Allokation konsistenter Namen wird vorausgesetzt. Dies kann beispielsweise erreicht werden, indem das Schema `/earth/iso-landeskürzel/nationaler_bezeichner` gewählt wird, wobei `nationaler_bezeichner` von einer Institution des jeweiligen Landes vergeben wird. So wird vermieden, dass die Gebäude Universitätsstrasse 38 und 39 korrekte, aber inkompatible Namen wie z.B. `/earth/europe/de/bw/stuttgart/vaihingen/universitaetsstrasse/38` und `/earth/europe/de/bw/stuttgart/universitaetsstrasse/39` verwenden. Es wird jedoch nicht davon ausgegangen, dass Komponenten, die in Adressen an der gleichen Stelle stehen, einen ähnlichen Bereich beschreiben. Eine Adresse wie `/earth/north_america/us/san_francisco/UCSF` ist ebenso möglich wie die angegebenen Adressen, die noch ein Bundesland enthalten.

Ortsbestimmung

Lokationsinformationen werden unabhängig vom Mesh-Netz über räumlich beschränkte Kommunikation zwischen Mesh-Clients und Markern bekannt gemacht. Solche Marker können z.B. RFID-Tags, Funkbojen mit extrem geringer Reichweite oder per Kamera zu erfassende Barcodes sein.

Die Marker können in Räumen (z.B. an Türrahmen, in der Decke) oder Objekten in der Umgebung (Briefkästen, Lichtmasten,...) angebracht werden. Alle mit Geocast adressierbaren Clients haben die Möglichkeit, solche Marker in ihrer Umgebung zu erkennen und die darin gespeicherten Lokationsinformationen auszulesen.

Unter Normalbedingungen haben Marker eine Reichweite von nur wenigen Metern. Dies ist ausreichend für eine raumgenaue Bestimmung des Standortes. Die Speicherkapazität

der Marker wird als groß genug angenommen, um den kompletten Präfix der Lokation zu speichern. Angesichts der Verfügbarkeit von RFID-Tags bzw. 2D-Barcodes mit Speicherplatz in der Größenordnung von mehreren hundert Bytes erscheint diese Annahme realistisch.

Mesh-Router haben keine Möglichkeit, Marker auszulesen. Jedoch können sie bei ihrer Installation mit einer oder mehreren Lokationen, die sich in ihrer Funkreichweite befinden, vorkonfiguriert werden, falls das verwendete Geocast-Protokoll Lokationsinformationen auf Mesh-Routern erfordert. Ebenso ist es möglich, dass Mesh-Router ihre Lokation(en) durch Beobachten des Netzes lernen können; genauere Überlegungen hierzu finden sich in Abschnitt 4.2.

Verkehrsmodell

Einen wesentlichen Einfluss auf die Qualität eines Geocast-Protokolls hat ein der Verkehrscharakteristik angepasstes Design. Wenn Geocasts nur sporadisch versendet werden, ist die zusätzliche Latenz durch ein reaktives Protokoll wie AODV problemlos zu verkraften. Falls jedoch häufig Geocasts mit wechselnden Quellen versendet werden, wird das Mesh-Netz möglicherweise durch häufiges Fluten von `RREQ`s lahmgelegt. In diesem Falle lohnt es sich, eher auf ein proaktives Verfahren analog zu AODV-Multicast oder OLSR umzusteigen.

Da jedoch bisher alle Versuche mit Geocast nur in Simulationen stattfanden, liegen keine Informationen zu realen Verkehrscharakteristiken von Geocast-Nachrichten vor. Für die im Folgenden entwickelten Protokolle und die anschließenden Simulationen wird daher folgendes Modell angenommen:

Sender Es wird erwartet, dass es einige wenige Knoten gibt, die als eine Art Nachrichtenzentralen sehr häufig sehr viele Geocast-Nachrichten versenden. Beispielsweise kann ein Knoten in einem Verkehrslagezentrum Staumeldungen in betroffene Gebiete verteilen. Das Gros der Knoten wird lediglich vereinzelt Nachrichten senden.

Empfangsbereiche Geocast-Nachrichten werden voraussichtlich hauptsächlich an eine begrenzte Menge von Präfixen gesendet, z.B. in bestimmte Räume oder Gebäude. In den meisten Gebieten ergibt sich nur ein sehr schwacher Geocast-Traffic.

3.2 Anforderungen

Im Rahmen dieser Arbeit sollen Geocast-Routingprotokolle für das oben genannte Systemmodell entworfen und analysiert werden. Die entworfenen Protokolle sollen effizient arbeiten, müssen jedoch lediglich einen Best-Effort-Dienst zur Verfügung stellen. Die folgenden Abschnitte stellen diese Anforderungen genauer zusammen.

3.2.1 Grundlegende Anforderungen

Es sind Geocast-Routingprotokolle zu entwerfen, die es Knoten im Mesh-Netz zu jeder Zeit ermöglichen, Geocast-Nachrichten zu beliebigen Geocast-Domänen zu senden, an denen sich Knoten befinden, die die Nachrichten empfangen können. Sind keine Empfänger vorhanden, muss die Nachricht nicht versendet werden. Es ist ein hierarchisches Lokationsmodell mit symbolischen Adressen zu unterstützen.

Als Grundlage dienen Mesh-Netze nach dem in Abschnitt 3.1 vorgestellten Systemmodell. Die Zielgröße der Netze sind Metropolitan Area Networks, d.h. Netze, die ein Stadtgebiet mit mehreren tausenden Knoten abdecken. Die entwickelten Protokolle sollen optimal an die Konzepte der zugrunde liegenen Unicast-Routingprotokolle AODV und OLSR angepasst werden.

3.2.2 Effektivität

Mesh-Netze bieten lediglich einen Best-Effort-Dienst. Daher wird auch von auf solchen Netzen aufsetzenden Geocast-Routingprotokollen keine bessere Dienstgüte gefordert. In nicht-partitionierten und nicht-überlasteten stabilen Netzen wird somit erwartet, dass eine Geocast-Nachricht nach wenigen Sekunden an alle Zielknoten ausgeliefert wird. Im Falle von Netzpartitionierungen, Netzüberlastungen oder Rekonfigurationen ist auch eine verzögerte Zustellung an nur einen Teil der Zielknoten akzeptabel. Da keine sichere Übertragung gefordert wird, kann es in solchen Fällen auch dazu kommen, dass eine Nachricht verworfen wird.

Für die Beurteilung der praktischen Einsetzbarkeit entwickelter Protokolle wird jedoch festgelegt, dass Nachrichten, bei deren Versand mehr als 60 Sekunden zwischen Sendeanforderung und Auslieferung vergehen, als verloren angesehen werden sollen. Ebenso sollte die Nachrichtenverlustrate in nicht-überlasteten Netzen 5% nicht übersteigen. Andernfalls wird ein menschlicher Benutzer den Dienst mit hoher Wahrscheinlichkeit als zu fehlerbehaftet ansehen und von einer weiteren Verwendung absehen.

Nachrichten, die ihre Zieldomäne erreichen, müssen an alle Knoten innerhalb der Geocast-Domäne ausgeliefert werden. Eine Ausnahme bilden partitionierte Geocast-Domänen. Da davon ausgegangen wird, dass Partitionierungen nur selten auftreten, ist es aufgrund des Best-Effort-Gedankens ausreichend, eine Nachricht nur allen Knoten innerhalb einer einzigen Partition zuzustellen. Eine Abdeckung von mehreren oder sogar allen Partitionen erhöht jedoch die für einen Benutzer erkennbare Dienstgüte und ist daher, wenn möglich, anzustreben.

Die Mobilität der Clients stellt besondere Herausforderungen für mögliche optimierte Geocast-Routingprotokolle dar. Für die beste Verbreitung von Nachrichten müssen Clients zum Routing eingesetzt werden. Andererseits darf jedoch nicht ignoriert werden, dass ein Client möglicherweise durch seine Mobilität schon nach kurzer Zeit nicht mehr dort als Router zur Verfügung steht, wo er bei der Routenfindung eingeplant war. Abreißende Routen

dürfen also nicht als exotischer Sonderfall behandelt werden, sondern müssen bei allen Entwürfen und Optimierungen als Regelfall miteinbezogen werden.

3.2.3 Effizienz

Um eine effiziente Nachrichtenverteilung zu ermöglichen, soll die Nachrichtenmenge minimiert werden. Routingtabellen mit mehreren hundert Einträgen haben keinen Einfluss auf die Funktion oder Performance moderner Router. Falls jedoch unnötige Nachrichten versendet oder Nachrichten in falsche Bereiche des Netzes geleitet werden, beeinträchtigt dies aufgrund des gemeinsamen Übertragungsmediums direkt jegliche andere parallel stattfindende Kommunikation.

Falls Geocast-Routingprotokolle Nachrichten per Broadcast verteilen bzw. fluten müssen, muss der Radius dieser Flutung so klein wie möglich gehalten werden, damit nicht unnötigerweise Netzbereiche berührt werden, in denen die Nachricht keine Relevanz besitzt.

Da der Großteil der Netzknoten mobile Mesh-Clients sind, die von ihren Benutzern auch zu anderen Zwecken als dem Geocast-Routing genutzt werden, soll die Last pro Knoten möglichst gering gehalten werden, d.h. die durch das Geocast-Routing erzeugte Last soll gleichmäßig auf alle Netzknoten verteilt werden. Falls es unumgänglich ist, dass sich Hot-Spots, d.h. Knoten, die einen übermäßigen Anteil der Last zu tragen haben, herausbilden, müssen diese Rollen soweit möglich von Mesh-Routern übernommen werden.

Routen sollen möglichst kurz gehalten werden. Dies führt zu weniger Übertragungen je Nachricht und damit auch zu einer geringeren Netzlast.

Weitere Optimierungen wie Energieeffizienz oder die auf den einzelnen Knoten benötigte Rechenleistung sind nicht Gegenstand dieser Arbeit.

4 Entwurf

In diesem Kapitel werden mehrere Möglichkeiten dargelegt, wie Geocast-Routing in Mesh-Netzwerken realisiert werden kann. Durch eingehende Betrachtung der Vor- und Nachteile jedes Ansatzes wird argumentiert, warum der Ansatz mehr oder weniger gut geeignet ist. Schließlich werden zwei geeignete Ansätze für AODV und OLSR als zugrunde liegende Unicast-Routingprotokolle ausgewählt und ein detaillierter Entwurf passender Geocast-Routingprotokolle vorgestellt.

4.1 Übersicht und Taxonomie der Routingverfahren

Um Geocast-Routingverfahren zu klassifizieren, müssen verschiedene Eigenschaften untersucht werden. Allen Geocast-Verfahren ist gemein, dass es sich um Punkt-zu-Multipunkt-Kommunikation handelt, d.h. dass Nachrichten an eine Gruppe von Zielen ausgeliefert werden müssen.

Die erste wichtige Eigenschaft betrifft die Bestimmung der Gruppenzugehörigkeit. Geographische Routingprotokolle verwenden direkt die Lokationen der einzelnen Knoten, während abbildungsbasierte Verfahren die Gruppenzugehörigkeit auf andere Weise bestimmen, z.B. durch eine Abbildung von Lokationen auf Multicast-Gruppen.

Weiter unterteilen lassen sich Geocast-Routingprotokolle danach, ob das Routing in einer oder in mehreren Stufen erfolgt. Einstufige Routingverfahren bilden Routen vom Sender zu jedem Zielknoten aus, während sich mehrstufige Routingverfahren die Korrelation zwischen geographischer und topologischer Nähe zu Nutze machen: Es werden unterschiedliche Routingprotokolle eingesetzt um eine Nachricht zur Zielgruppe zu transportieren und um die Nachricht innerhalb der Zielgruppe zu verteilen.

Eng mit der Ein- oder Mehrstufigkeit verknüpft ist die Eigenschaft, ob ein Routingprotokoll eine (Single-Path-Routing) oder mehrere (Multi-Path-Routing) Routen zur Zielgruppe aufbaut. Multi-Path-Routingansätze versuchen zu jedem Ziel die kürzeste Route zu bestimmen und müssen daher überall dort, wo sich eine Route aufteilt, eine Nachricht vervielfältigen, um sie korrekt zustellen zu können. Single-Path-Routingprotokolle bestimmen dagegen nur eine einzige Route zur Zielgruppe, so dass Single-Path-Routing ähnlich zu normalem Unicast-Routing abläuft.

Die Verknüpfung zur Ein- oder Mehrstufigkeit ist leicht ersichtlich und wird in Abbildung 4.1 illustriert. Single-Path-Routingprotokolle müssen immer mehrstufige Protokolle sein. Da

(a)

Routing-Protokoll	Gruppenzugehörigkeit	Routingstufen	Single-/Multi-Path
Geo-Multicast AODV	Abbildung	einstufig	Multi-Path
Geo-Multicast OLSR	Abbildung	einstufig	Multi-Path
Multi-Path AODV	geographisch	einstufig	Multi-Path
Multi-Path OLSR	geographisch	einstufig	Multi-Path
Single-Path AODV	geographisch	mehrstufig	Single-Path
Single-Path OLSR	geographisch	mehrstufig	Single-Path

(b)

Routing-Protokoll	Genauigkeit	Pro-/Reaktiv
Geo-Multicast AODV	genau	proaktiv
Geo-Multicast OLSR	genau	proaktiv
Multi-Path AODV	ungenau	reaktiv
Multi-Path OLSR	genau	proaktiv
Single-Path AODV	ungenau	reaktiv
Single-Path OLSR	genau	proaktiv

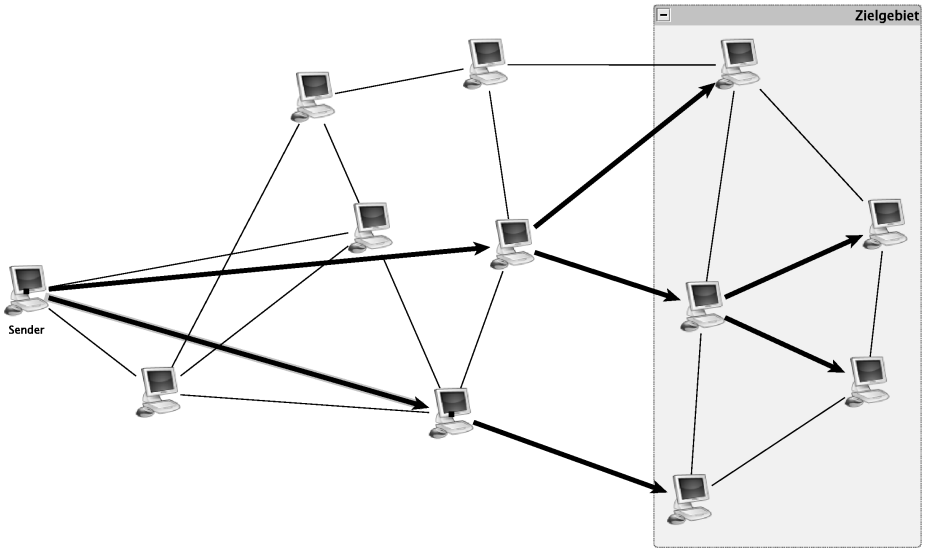
Tabelle 4.1: Klassifikation von Geocast-Routingalgorithmen

Pakete nicht vervielfältigt werden, muss ein anderes nachgelagertes Routingprotokoll die Verteilung an die Gruppe übernehmen. Single-Path-Routingprotokolle stellen damit eine Art Anycast-Dienst bereit: Es wird nur ein beliebiges Mitglied der Zielgruppe erreicht, was dann wiederum selbst zuständig für die Verteilung an den Rest der Gruppe ist.

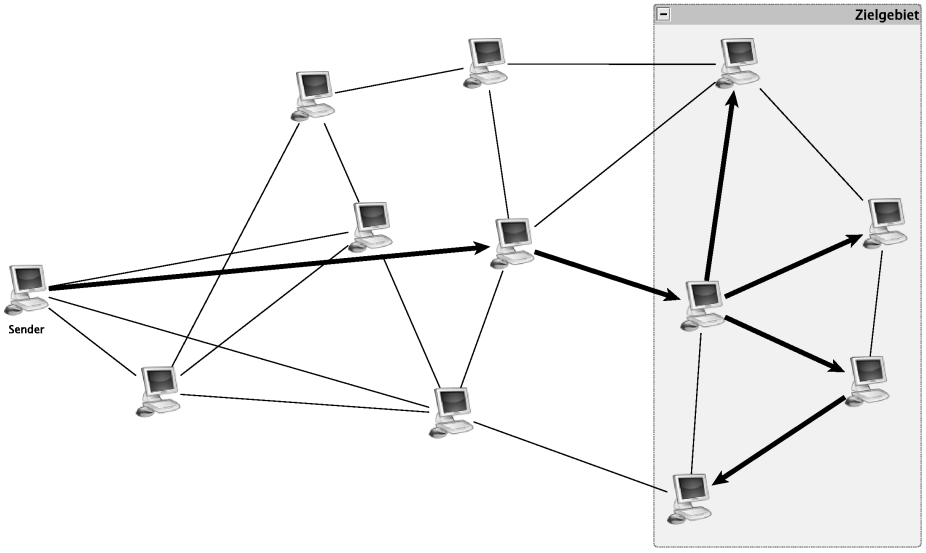
Multi-Path-Protokolle dürfen Nachrichten uneingeschränkt vervielfältigen und können daher wahlweise als ein- oder mehrstufige Protokolle auftreten. Mehrstufige Multi-Path-Routingprotokolle haben gegenüber einstufigen Protokollen den Vorteil, dass die Routenbestimmung darauf optimiert werden kann, netztopologisch weit entfernte Sender und Empfangsgruppen zu verbinden. Die Verteilung zwischen den topologisch dicht beieinander liegenden Zielknoten wird von der nachfolgenden Stufe übernommen und muss daher nicht berücksichtigt werden.

Angesichts dessen, dass kein zuverlässiger, sondern lediglich ein Best-Effort-Dienst gefordert ist, verliert die Genauigkeit von Routingprotokollen an Bedeutung. Nichtsdestoweniger handelt es sich um ein Merkmal von Geocast-Routingverfahren. Ungenaue Verfahren nehmen in Kauf, dass nur ein Teil der Zielknoten erreicht wird, falls z.B. partitionierte Geocast-Domänen auftreten. Genaue Protokolle versuchen dagegen partitionierte Geocast-Domänen zu kompensieren und im Rahmen ihrer Möglichkeiten alle Zielknoten zu erreichen. Bei schwerer wiegenden Fehlern wie Überlastungen oder Netzpartitionierungen können jedoch auch genaue Protokolle nicht alle Zielknoten erreichen.

Eine letzte Eigenschaft ist die proaktive oder reaktive Arbeitsweise des Protokolls. Diese Eigenschaft eignet sich weniger dazu Geocast-Routingprotokolle zu klassifizieren, beschreibt



(a) Ein- oder mehrstufiges Multi-Path-Routing im Mesh-Netz: Vervielfältigung von Nachrichten auf dem Weg ins Zielgebiet



(b) Mehrstufiges Single-Path-Routing im Mesh-Netz: Vervielfältigung von Nachrichten nur im Zielgebiet

Abbildung 4.1: Dünne Linien sind verfügbare Links, dicke Linien sind Links, über die die Geocast-Nachricht ins Zielgebiet geleitet wird.

aber in einem gewissen Maße, wie gut das Geocast-Protokoll an das zugrunde liegende Unicast-Routingprotokoll angepasst ist.

Im Folgenden werden sechs Varianten von Geocast-Routingalgorithmen analysiert. Tabelle 4.1 zeigt, wie sich die einzelnen Verfahren nach den vorgestellten Eigenschaften einordnen lassen.

Abhängig von der genauen Arbeitsweise der Geocast-Protokolle müssen alle Mesh-Knoten ihre Lokation kennen oder können Mesh-Knoten ohne Kenntnis ihrer Lokation toleriert werden. Es lässt sich jedoch nur vordergründig eine klare Unterscheidung zwischen diesen beiden Fällen ziehen. Tatsächlich ist es möglich, auch in einem Netz, in dem nicht alle Knoten Lokationsinformationen besitzen, Routingverfahren einzusetzen, die vollständig bekannte Lokationsinformationen voraussetzen. Wie sichergestellt werden kann, dass Lokationsinformationen auf allen Knoten vorhanden sind, beschreibt der folgende Abschnitt.

4.2 Lokationsinformationen

Verschiedene Geocast-Routingalgorithmen haben unterschiedliche Anforderungen daran, welche Knoten ihre Lokation kennen müssen. Wenn das Routing ausschließlich auf lokalen Lokationsinformationen beruht, müssen alle Knoten – Clients wie Router – ihre Lokation kennen. Andernfalls kann es ausreichend sein, wenn nicht alle Knoten Lokationsinformationen besitzen. Solche Knoten können zwar am Routing teilnehmen, treten jedoch nie als Empfänger von Geocast-Nachrichten auf.

Wird vorausgesetzt, dass jeder Netzknoten seine Lokation kennt, ist zusätzlicher Aufwand bei der Aufstellung neuer Router notwendig, da Mesh-Router nicht wie Clients die in der Umgebung verteilten Lokationsinformationen auslesen können und daher manuell mit einem Präfix konfiguriert werden müssen. Ebenso müssen, wenn sich der Präfix einer Lokation ändert, alle Router umkonfiguriert werden, die sich an der zugehörigen Lokation befinden.

Eine Alternativlösung wäre, dass Router ihre Lokation von Clients in der Nähe automatisch lernen. Dabei muss jedoch jeder Router mehrere Lokationen als seine eigene ansehen können, um Konflikte zu vermeiden: Es können sich mehrere Clients in Funkreichweite eines Routers befinden, wobei die Clients sich auf zwei oder mehr Lokationen aufteilen. Ein Router kann in diesem Fall nicht entscheiden, an welcher der zur Auswahl stehenden Lokationen er sich tatsächlich befindet. Um Anpassungen an geänderte Lokationen oder veränderte Reichweiten zu ermöglichen, müssen alle gelernten Lokationen mit einem Timeout versehen, d.h. als Soft-State gehalten werden. Da Änderungen der Lokationen relativ selten zu erwarten sind, erscheint ein Timeout von mehreren Tagen sinnvoll.

Die einfachste Variante des Lernens von Lokationen ist das direkte Lernen: Die Lokation wird von einem direkt verbundenen Client übernommen. Wenn jedoch nicht gerade eine sehr hohe Client-Dichte vorhanden ist, kann ein Knoten durchaus längere Zeit ohne gelernte Lokation bleiben. Dies kann dazu führen, dass Geocast-Domänen partitioniert werden oder das Nachrichten über Gebühr weit aus dem Zielgebiet herausgeleitet werden (siehe Abschnitt 4.5.1).

Um diesem Problem zu begegnen können Lokationen auch über größere Entfernungen gelernt werden. Dabei befragen alle Knoten in einem Gebiet, in dem keine Lokationsinformationen vorliegen, ihre Nachbarschaft. Per Broadcast wird die Anfrage von jedem Knoten weitergegeben, bis Knoten erreicht werden, die ihre Lokation kennen. Diese leiten die Anfrage nicht weiter, sondern antworten mit ihrer Lokation. Somit lernen alle Knoten im fraglichen Gebiet alle Lokationen, die am Rand dieses Gebietes auftreten. Das gesamte Gebiet wird dann von außen gesehen zu einem Meta-Knoten, der sich in Reichweite aller dieser Lokationen befindet.

Ein Knoten, der seine Lokation nur indirekt gelernt hat, darf seine Lokation nicht an andere Knoten weitergeben. Andernfalls können durch unglückliche Ketten Lokationen an entfernte Enden des Netzes wandern und so unnötige partitionierte Geocast-Domänen erzeugen.

Falls eine Lokation später direkt gelernt werden kann, werden alle indirekt gelernten Lokationen vergessen. Falls das bisher lokationslose Gebiet dadurch partitioniert wird, dass ein Knoten eine Lokation direkt gelernt hat, müssen die Lokationen in den beiden neuen Gebieten erneuert werden. Dies kann unter anderem dadurch erreicht werden, dass indirekt gelernte Lokationen nur eine geringe Gültigkeitsdauer haben und somit häufig nach indirekt zu lernenden Lokationen gesucht werden muss.

4.3 Multicastbasierende Implementierungen

Multicast stellt bereits eine Punkt-zu-Multipunkt-Kommunikation zur Verfügung. In diesem Abschnitt wird untersucht, inwieweit sich bereits verfügbare Multicast-Protokolle zur Auslieferung von Geocast-Nachrichten eignen.

4.3.1 Beziehung zwischen Geocast und Multicast

Geocast und Multicast haben gemeinsam, dass die Routing-Protokolle dafür sorgen sollen, dass möglichst wenig Kopien einer Nutznachricht je zur Verfügung stehender Leitung übertragen werden müssen. Geocast-Router müssen wie Multicast-Router in der Lage sein, Pakete zu vervielfältigen.

Multicast erfordert ein Gruppenmanagement, d.h. es muss überwacht werden, wo im Netz sich interessierte Knoten befinden. Clients müssen sich also an Multicast-Gruppen an- und abmelden, wenn sie sich durch das Netz bewegen. Der Nachrichtenaufwand hierfür kann, abhängig vom verwendeten Multicast-Protokoll, gering gehalten werden, da sich Gruppenmitglieder im Falle von Geocast in der Regel topologisch nahe sind. Falls eine Geocast-Domäne partitioniert ist, existieren nur wenige lange Strecken durch das umgebende Netz.

Multicast zur Implementierung von Geocast wurde in infrastrukturbasierten Systemen bereits erforscht [IN96]. Jedoch sind bisher alle Versuche an dem Problem gescheitert, dass

nicht genug Multicast-Gruppen zur Verfügung stehen um eine hinreichend feingranulare Adressierung bei gleichzeitig effizienter Nachrichtenverteilung zu ermöglichen.

4.3.2 Adressierung

Wie in Abschnitt 4.3.1 bereits diskutiert, scheitert das Abbilden von Geocast auf Multicast-Adressen bisher auf globaler Ebene an der vergleichsweise geringen Anzahl von Multicast-Adressen. Auch in den in dieser Arbeit betrachteten Metropolitan Area Networks ist damit zu rechnen, dass die Anzahl der abzubildenenden Lokationen die Anzahl der verfügbaren Multicast-Gruppen übersteigt. Erst mit IPv6-Multicast wird der Adressraum für Multicast-Adressen derart erweitert, dass ein Mapping von Lokationen auf Multicast-Adressen wieder möglich wird. In jedem Fall ist Geocast jedoch nicht die einzige Multicast-Anwendung. Den Großteil der verfügbaren Multicast-Adressen für Geocast zu reservieren erscheint daher nicht als realistisch.

Um das Problem des begrenzten Adressraumes zu umgehen muss der Präfix der Lokation als Multicast-Gruppenadresse verwendet werden. Dabei können zwar bestehende Multicast-Algorithmen weiterverwendet werden, jedoch sind umfangreiche Anpassungen an Implementierungen notwendig. So muss zum einen der Paketheader um ausreichenden Platz für die Lokation erweitert werden. Zum anderen treffen bestehende IP-Stacks ihre Routingentscheidungen im wesentlichen nach der Zieladresse. Hier muss die eventuell angehängte Lokation in die Routingentscheidungen einbezogen werden. Ein Umschalten zwischen regulärem zieladressbasiertem Multicast und lokationsbasiertem Geocast kann z.B. durch eine reservierte Zieladresse für Geocast-Pakete erfolgen.

Abgesehen von dieser Modifikation werden die Multicast-Protokolle ausgeführt, wie es in den Abschnitten 2.4.1 „Multicast mit AODV“ und 2.4.2 „Multicast Optimized Link State Routing“ beschrieben ist.

4.3.3 Geo-Multicast-Routing bei AODV

Die Multicast-Erweiterungen für AODV bieten Shared-Tree-Multicast mit einem Gruppenführer als Rendezvous-Punkt (siehe Abschnitt 2.4.1). Der Gruppenführer ist derjenige Knoten, der als erstes Nachrichten an die Multicast-Gruppe senden möchte. Weitere Gruppenmitglieder werden gefunden, indem ein `RREQ` für die Gruppenadresse durch das Netz geflutet wird. `RREP`-Nachrichten werden wie im Unicast-Fall auch über die Rückroute zum anfragenden Knoten geleitet.

Nachrichten werden innerhalb der Multicast-Gruppe von jedem Knoten an alle benachbarten Knoten im Baum weitergeleitet – mit Ausnahme des Knotens, von dem die Nachricht erhalten wurde.

Ein klarer Nachteil von Geo-Multicast-Routing bei AODV ist, dass die Lokalitätseigenschaften von Geocast in Mesh-Netzwerken völlig außer Acht gelassen werden. So wird der erste Knoten, der Nachrichten an eine Gruppe senden möchte zum Gruppenführer und kann die

Gruppe danach nicht mehr verlassen, bis alle anderen Knoten die Gruppe verlassen haben. Da ein Sender von Geocast-Nachrichten typischerweise außerhalb des Zielgebietes liegt und Multicast-Gruppen erst dann aus dem Netz entfernt werden, wenn keine Empfänger für diese Gruppe mehr existieren, muss also für jede angefragte Multicast-Gruppe ein Pfad zu einem weit außerhalb gelegenen Knoten aufrechterhalten werden.

Lässt man die Forderung fallen zu einem reaktiven Unicast-Routingprotokoll auch ein reaktives Geocast-Routingprotokoll verwenden zu wollen, kann das Problem des entfernten Gruppenführers gelöst werden, indem Multicast-Gruppen für jede Lokation proaktiv aufgebaut werden. Dies erfordert jedoch einen enormen Mehraufwand an Nachrichten: Für jede Gruppe, d.h. einmal für jeden im Netz existierenden Präfix, muss zur Aufrechterhaltung der Gruppe das Netz periodisch geflutet werden. Dieser Mehraufwand lässt sich mit einer weiteren Modifikation am Protokoll beschränken: Knoten im Zielgebiet müssen nicht das gesamte Netzwerk fluten. Für diese Knoten ist es ausreichend, eine Nachricht an ihre Nachbarn zu senden, um ihre Mitgliedschaft in der zur Lokation gehörigen Multicast-Gruppe bekannt zu geben. Damit ließe sich der Faktor Gruppenanzahl aus dem periodischen Flutungsaufwand entfernen.

Auch entfernte Sender müssen in einem modifizierten Protokoll nicht zwingend das gesamte Netz fluten. Der Einsatz von Expanding Ring Search (ERS) kann den Aufwand abmildern. Bei ERS werden Anfragen mehrfach in zeitlichem Abstand per limitierter Flutung versendet. Der Flutungsradius steigt bei jeder Neuübertragung einer Anfrage an, bis eine Antwort eingeht oder ein festgelegtes Abbruchkriterium, z.B. ein maximaler Flutungsradius, erreicht wird. Dabei geht jedoch eine wichtige Eigenschaft einer multicastbasierenden Implementierung verloren: Es werden nicht mehr in allen Fällen alle Knoten der Gruppe erreicht, d.h. falls partitionierte Geocast-Domänen auftreten wird eine Geocast-Nachricht mit hoher Wahrscheinlichkeit nur eine oder wenige Partitionen erreichen.

4.3.4 Geo-Multicast-Routing bei OLSR

MOLSR bietet Source-Based Trees für Multicast (siehe Abschnitt 2.4.2). Es ist für die Funktion des Protokolles erforderlich, dass im Netz Multicast-Router existieren, die sich mit periodisch durch das Netz gefluteten Nachrichten gegenseitig bekannt machen. Der Overhead zur Bekanntmachung der Multicast-Router ist ähnlich dem zur Verteilung der zur Berechnung von Unicast-Routen verwendeten `TC`-Nachrichten.

Ein Knoten, der an eine Multicast-Gruppe senden möchte, flutet eine Nachricht durch das Netz um alle potentiellen Empfänger über die bevorstehende Übertragung zu informieren. Die Nachricht wird periodisch wiederholt, um auch Knoten zu erreichen, die während einer laufenden Übertragung der Multicast-Gruppe beitreten.

Alle Gruppenmitglieder senden als Reaktion auf die Benachrichtigung eine Nachricht, die sie als Empfänger identifiziert, über ihre nächstgelegenen Multicast-Router zum neuen Sender. Die Pfade, die von diesen Nachrichten genommen werden, definieren den Verteilbaum für diese Kombination aus Quelle und Multicast-Gruppe.

Auch für Geo-Multicast-Routing bei OLSR ergibt sich ein zur Verteilung von Geocast-Nachrichten unnötig hoher Flutungsaufwand. Die Bekanntmachungen der Multicast-Router amortisieren sich zwar, wenn Multicast ohnehin genutzt wird. Was sich jedoch nicht vermeiden lässt, sind die Flutungen um Multicast-Gruppen bekannt zu geben. Es muss mindestens einmal für jeden Sender geflutet werden, im schlimmsten Fall sogar für jedes versendete Geocast-Paket.

Optimierungen an diesem Protokoll sind deutlich komplexer als bei AODV-basiertem Multicast. Bei OLSR-basiertem Multicast werden Source-Based-Trees aufgebaut, d.h. die Multicast-Verteilbäume existieren nur genau so lange, wie der Sender Nachrichten an die Gruppe senden möchte. Ein proaktiver Aufbau des Verteilbaumes im Zielgebiet ist somit nicht möglich.

Auch der Einsatz von ERS zur Einschränkung der zu flutenden Bereiche des Netzes ist nicht unproblematisch. Um den Verteilbaum korrekt aufzubauen, müssen ausnahmslos alle Knoten im Zielgebiet erreicht werden. ERS erreicht jedoch potentiell nur einen einzigen Knoten im Zielgebiet.

Um dieses Problem zu umgehen und ERS trotzdem einsetzen zu können muss sichergestellt werden, dass eine SOURCE_CLAIM-Nachricht im Zielgebiet von Knoten zu Knoten weitergegeben wird, auch wenn der Suchradius bereits überschritten wurde.

Da periodisches Fluten nur notwendig ist, solange ein Multicast-Verteilbaum aufrecht erhalten werden soll, d.h. solange Geocast-Nachrichten gesendet werden sollen, reduziert sich die Anzahl der Flutungen von der Anzahl der exitierenden Geocast-Präfixe auf die Anzahl der adressierten Präfixe. Mittels ERS kann der Durchmesser des gefluteten Bereiches eingeschränkt werden, was jedoch wie bei AODV-basiertem Multicast dazu führt, dass partitionierte Geocast-Domänen nicht mehr vollständig abgedeckt werden können.

4.3.5 Nachteile einer Multicast-Lösung

Sowohl die Multicast-Protokolle für AODV als auch für OLSR müssen modifiziert werden, um effizienten Geocast zu ermöglichen. Auf einen unmodifizierten Multicast-Algorithmus aufzusetzen und lediglich das Mapping bereitzustellen hat sich nicht als praktikabel erwiesen, da Multicast-Protokolle häufig das gesamte Netz fluten müssen.

Wird ein Multicast-Protokoll modifiziert um es besser an die Eigenheiten von Geocast in Mesh-Netzen anzupassen, gehen die beiden wesentlichen Vorteile einer multicastbasierenden Implementierung, nämlich die Einfachheit durch Wiederverwendung bestehender Protokolle und die vollständige Abdeckung des Zielgebietes, verloren. Unter diesen Umständen erscheint es geeigneter, neue Protokolle für die Verteilung von Geocast-Nachrichten zu entwickeln. Tatsächlich sind die meisten der im Folgenden vorgestellten Routing-Algorithmen von den hier vorgestellten Multicast-Algorithmen und ihren Optimierungen inspiriert.

4.4 Multi-Path-Routing

Ähnlich dem multicastbasierten Geocast werden bei Multi-Path-Routing mehrere Routen zwischen Sender und Zielgebiet aufgebaut. In diesem Abschnitt soll erforscht werden, wie Multi-Path-Routingalgorithmen besser an die Eigenschaften von AODV-basierten bzw. OLSR-basierten Mesh-Netzen angepasst werden können.

4.4.1 Das Netz als Vektorfeld

Beim Multi-Path-Routing soll eine Nachricht über die optimalen Routen vom Sender zu jedem Knoten im Zielgebiet übertragen werden. Dies erfordert, dass Knoten mit mehreren Routingeinträgen zu einem Ziel umgehen können, da sich die optimalen Routen möglicherweise noch vor dem Zielgebiet teilen (siehe Abbildung 4.1(a)).

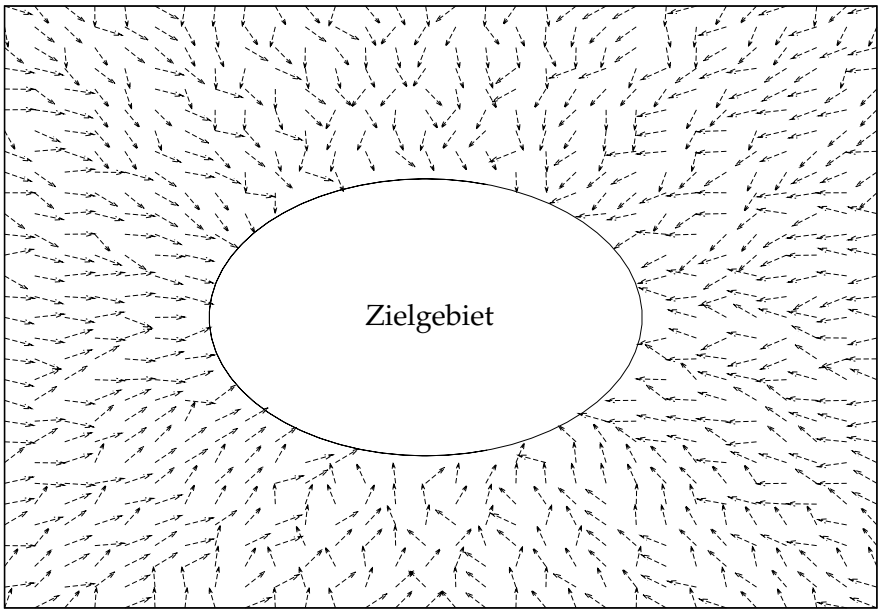


Abbildung 4.2: Illustration des Vektorfeldes der Routen für ein Zielgebiet (Oval in der Mitte). Jeder Pfeil symbolisiert einen Knoten außerhalb des Zielgebietes. Die Pfeilrichtung gibt an, wohin ein Knoten eine an das Zielgebiet adressierte Nachricht weiterleitet.

Betrachtet man ein Mesh-Netz „von oben“ und visualisiert für ein Zielgebiet alle Routen im Netz als Pfeile, so ergibt sich ein Vektorfeld, in dem alle Vektoren weit außerhalb des Zielgebietes auf diese zeigen (siehe Abbildung 4.2). Erst in der Nähe werden mehrere Vektoren pro Knoten, die in unterschiedliche Richtungen zeigen, sichtbar. Innerhalb des Zielgebietes würde schließlich jeder Knoten Vektoren in jede Richtung besitzen. Die Lokation,

bzw. im Falle von partitionierten Geocast-Domänen die Partitionen der Lokation wirken in der Draufsicht als Attraktoren. Diesen Zustand herzustellen scheint aber nur auf den ersten Blick einfach, wie in den folgenden Abschnitten erläutert wird.

4.4.2 AODV

Bei AODV werden aufgrund der reaktiven Routenbestimmung keine Routing-Informationen vorgehalten. Das Vektorfeld entsteht lediglich konzeptionell als Überlagerung aller Routen für eine Lokation, die über eine gewisse Zeit aufgebaut wurden.

Um die Zielknoten für einen Geocast zu finden, ist bei AODV ein Fluten des Netzwerkes notwendig. Da jedoch die Netzlast minimiert werden soll, muss vorsichtiger nach Informationen über das Zielgebiet gesucht werden – ERS bietet sich an, um die Zielknoten für einen Geocast im Netz zu finden, ohne notwendigerweise das gesamte Netz zu belasten.

Alle Knoten, die Anfragen zum Aufbau von Geocast-Routen weiterleiten, richten eine Unicast-Route zur Quelle der Anfrage über den letzten Absender der Anfrage ein. Erreicht eine Anfrage einen Knoten im Zielgebiet, gibt dieser die Anfrage auch dann an seine Nachbarknoten im Zielgebiet weiter, wenn der Radius des ERS bereits überschritten wurde. Somit werden alle Zielknoten erreicht. Um die optimalen Geocast-Routen aufzubauen müssen nun lediglich alle Zielknoten eine Antwortnachricht über die bereits aufgebauten Unicast-Routen zur Quelle der Anfrage senden. Jeder Knoten, der eine Antwort weiterleitet, richtet eine Geocast-Route ins Zielgebiet über den letzten Absender der Antwort ein.

Es ergibt sich ein Source-Based-Tree zur Verteilung von Geocast-Nachrichten. Bei nicht-partitionierten Geocast-Domänen entspricht der Verteilbaum einem minimalen Spannbaum, d.h. es werden nur die nach Hop-Count kürzesten Routen verwendet. Im Fall von partitionierten Geocast-Domänen werden nur Partitionen erreicht, deren Entfernung zum Sender der minimalen Entfernung zwischen dem Sender und irgend einer Partition entspricht. Dieses Problem ist auch in Abbildung 4.3 dargestellt.

Das offensichtlich auftretende Implosionsproblem beim Sender lässt sich leicht dadurch beheben, dass Knoten, die bereits eine Antwort weitergeleitet haben, beim Erhalt einer weiteren Antwort eines anderen Zielknotens zwar einen Routingtabelleneintrag generieren, die Antwort aber nicht weiterleiten.

Im Wesentlichen wird der Aufwand für dieses Geocast-Protokoll dem von einfachem Fluten entsprechen. Daher sind noch weitere Optimierungen notwendig. Analog zu Unicast-AODV dürfen Knoten, die bereits eine Route ins Zielgebiet kennen, das ERS abbrechen und die Anfrage direkt per Geocast zu den Zielknoten leiten. Auf den ersten Blick führt dies dazu, dass der Source-Based-Tree zu einem Shared-Tree erweitert wird, ohne dass Mechanismen zur Gruppenverwaltung notwendig sind. Tatsächlich ergeben sich bei näherer Betrachtung jedoch zwei gravierende Probleme.

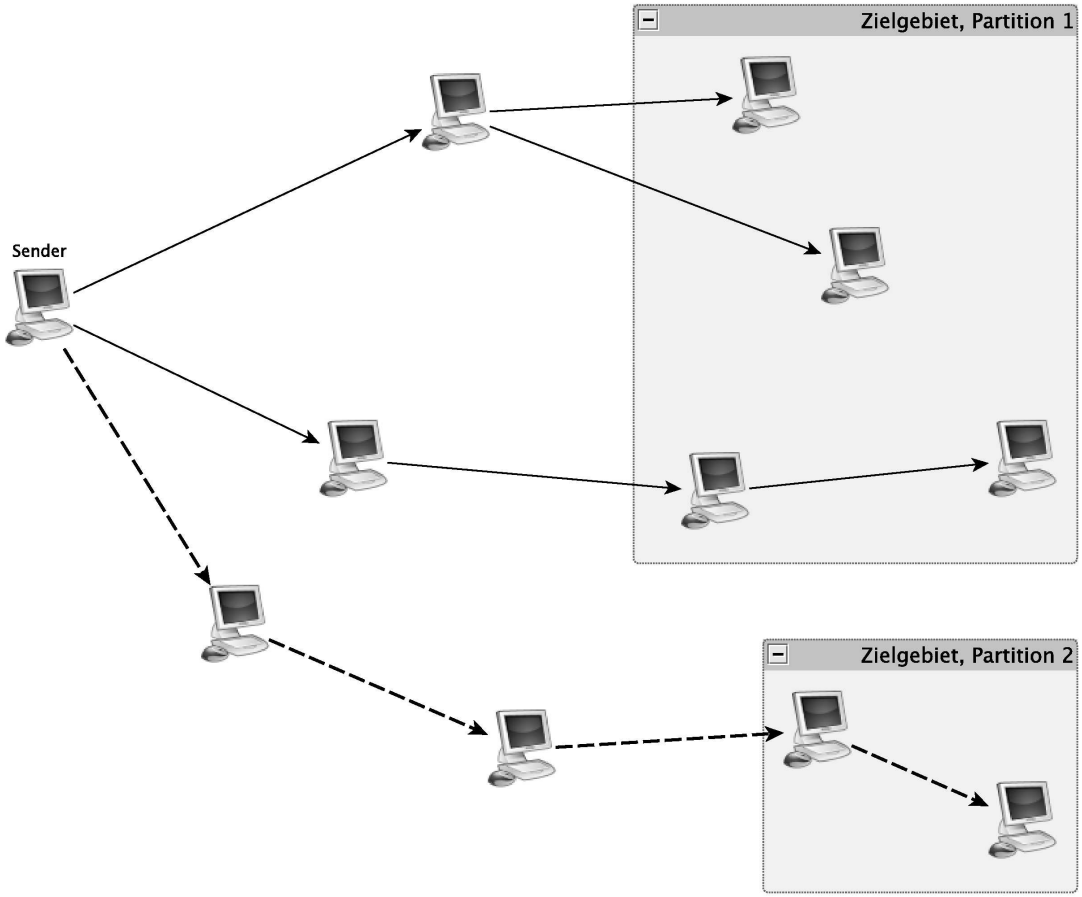


Abbildung 4.3: Routenfindung mittels ERS: Der Sender hat zwei Routen zu Partition 1 mit Hopcount 2 gefunden. Die gestrichelte Route zu Partition 2 wurde wegen des höheren Hopcounts von 3 nicht gefunden.

Das erste Problem betrifft die Rückmeldung der Zielknoten an die Quelle einer Anfrage. Bei der Wiederverwendung bestehender Routen ist die resultierende Route nicht notwendigerweise optimal. Um also optimale Geocast-Routen aufzubauen, müssten zunächst die optimalen Unicast-Routen von allen Zielknoten zum Sender bestimmt werden. Das Auffinden einer Unicast-Route erfordert bei AODV jedoch ein Fluten des gesamten Netzes von jedem Zielknoten aus. Statt den Nachrichtenaufwand für Geocast zu optimieren, vervielfacht dieses Protokoll den Aufwand. Als einzige Lösung bleibt, die Antworten über die umgekehrten Routen zu senden, die die Anfrage genommen hat. Dabei muss in Kauf genommen werden, dass die Geocast-Routen nicht mehr optimal sind.

Wird mit dem ERS ein Knoten erreicht, der bereits eine Geocast-Route für das angefragte Zielgebiet kennt, befindet sich dieser im Allgemeinen in der Mitte eines bestehenden Verteilbaums. Dies führt zum zweiten Problem: Falls der bestehende Verteilbaum zwischen der Wurzel und dem erreichten Knoten bereits eine Verzweigung enthält, kann nur noch der Teil

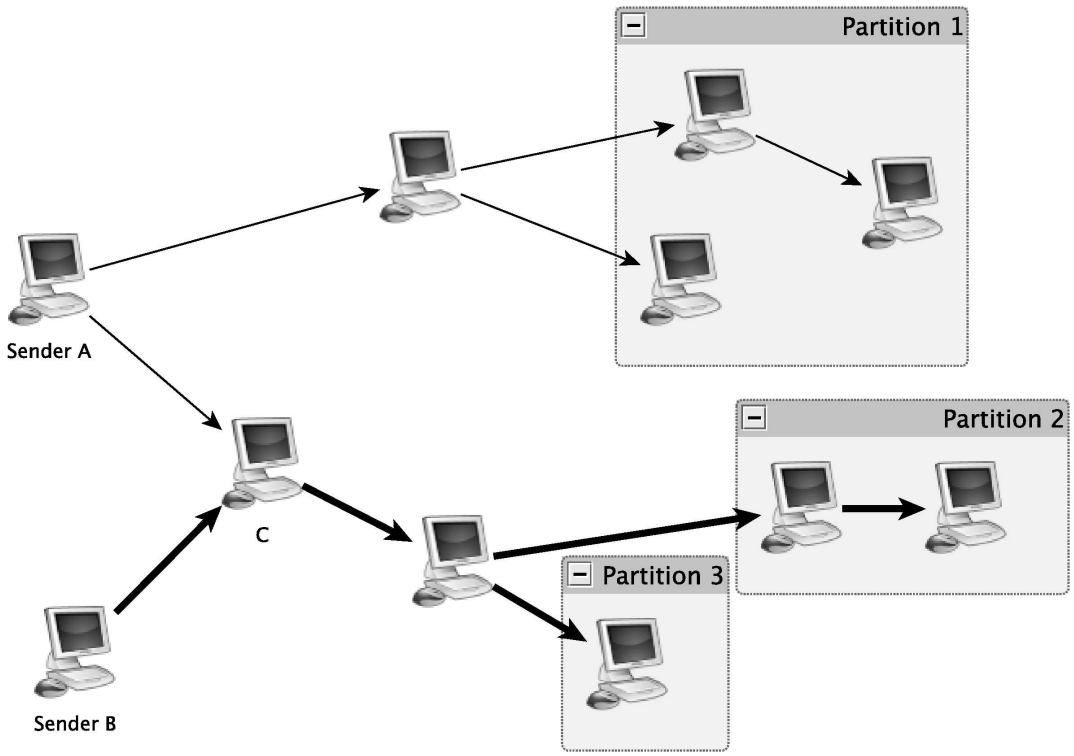


Abbildung 4.4: Abdeckung des Zielgebietes bei Wiederverwendung bestehender Routen: Sender A hat einen Verteilbaum zu allen Partitionen des Zielgebietes aufgebaut. Sender B verwendet alte Routen wieder (dicke Pfeile) und steigt bei C in den bestehenden Verteilbaum ein. B kann nun Partition 1 nicht mehr erreichen.

des Zielgebietes erreicht werden, der sich unterhalb des erreichten Knotens befindet (siehe Abbildung 4.4). Dieses Verhalten ist jedoch nach den Anforderungen aus Abschnitt 3.2 nur erlaubt, falls das Zielgebiet partitioniert ist.

Um auch die Teile des Zielgebietes zu erreichen, die sich in anderen Ästen des Verteilbaumes befinden, müsste eine Anfrage auch über die Wurzel des Verteilbaumes geleitet werden, von der aus dann alle Äste erreichbar sind. Da die Unicast-Antworten aber den umgekehrten Weg nehmen, folgt daraus, dass spätere Geocast-Nachrichten des neuen Senders ebenso die Wurzel des alten Verteilbaumes passieren müssten, was zu äußerst schlechten Routen führen kann.

Eine neue Route hätte also beim Wiederverwenden von Geocast-Routen sehr wahrscheinlich eine schlechtere Abdeckung des Zielgebietes als die zuerst aufgebaute Route. Doch wenn man dieses Problem weiter analysiert, stellt man fest, dass es in keinem Fall so weit kommt:

Durch den Einsatz von ERS wird häufig ohnehin nur eine Partition der Geocast-Domäne erreicht.

Da hierfür ein Protokoll mit der Komplexität eines Multi-Path-Routingprotokolles entworfen werden muss, das zumeist lediglich die Funktionalität eines Single-Path-Routingprotokolles bereit stellt, d.h. nur eine einzelne Partition einer Geocast-Domäne erreicht, wird von weiteren Überlegungen in dieser Richtung abgesehen. Der als Single-Path-Routingprotokoll konzipierte Ansatz wird in Abschnitt 4.5.2 entworfen.

4.4.3 OLSR

OLSR kann durch in `TC`-Nachrichten mitgesendete Informationen über die Präfixe von Knoten eine globale Sicht auf die Geocast-Domänen herstellen. Die daraus errechneten Routen für jeden Präfix ergeben direkt das Vektorfeld. Jedoch bedeutet dies durch die vergrößerten Nachrichten einen kontinuierlichen zusätzlichen Nachrichtenaufwand.

Die einzige Möglichkeit diesen Aufwand zu minimieren ist, die Genauigkeit der Nachrichten mit der Entfernung zu reduzieren und somit die globale Sicht einzuschränken. Das heißt, in der Nähe eines Zielgebietes werden mehrere exakte Routen vorgehalten, weiter entfernt wird dies aber auf einzelne bis, bei partitionierten Geocast-Domänen, wenige grobe Präfixe eingeschränkt.

Dieses Verhalten wird dadurch erreicht, dass Knoten auswählen können, für welche der Präfixe, zu denen sie bereits Geocast-Routen kennen, sie sich bei ihren Nachbarn als Next-Hop bekannt machen wollen. Für jeden bekannten Präfix muss entweder dieser Präfix oder ein Anfangsstück desselben beworben werden.

Auf jedem Knoten enthält die Geocast-Routingtabelle alle von Nachbarn beworbenen Präfixe zusammen mit der Adresse des jeweiligen Nachbarn.

Komprimierte Routingtabellen Eines der aus der Reduzierung der Genauigkeit resultierenden Probleme betrifft die Verwendung von Routingtabellen. Bei exakten Routingtabellen sind alle Einträge zutreffend, bei denen die Adresse der Geocast-Nachricht ein Präfix der Adresse in der Routingtabelle ist.

Wenn jedoch Routingtabelleneinträge zu kürzeren Präfixen aggregiert werden, müssen Nachrichten auch über Routen gesendet werden, bei denen der neue Tabelleneintrag nur ein Anfangsstück der Adresse aus der Nachricht ist.

Abbildung 4.5 illustriert das daraus resultierende Catch-All-Problem. Auf Knoten B eintreffende Nachrichten an `/earth/stuttgart/uni/informatik` müssen korrekterweise an Knoten A und D weitergeleitet werden. Genauso müssen nach den Routing-Regeln Nachrichten an `/earth/stuttgart/uni/maschinenbau` an Knoten C und – diesmal fälschlicherweise – an Knoten A weitergeleitet werden, da nicht bekannt ist, ob nicht noch weitere Lokationen unterhalb von `/earth/stuttgart/uni/maschinenbau` über Knoten A erreichbar sind.

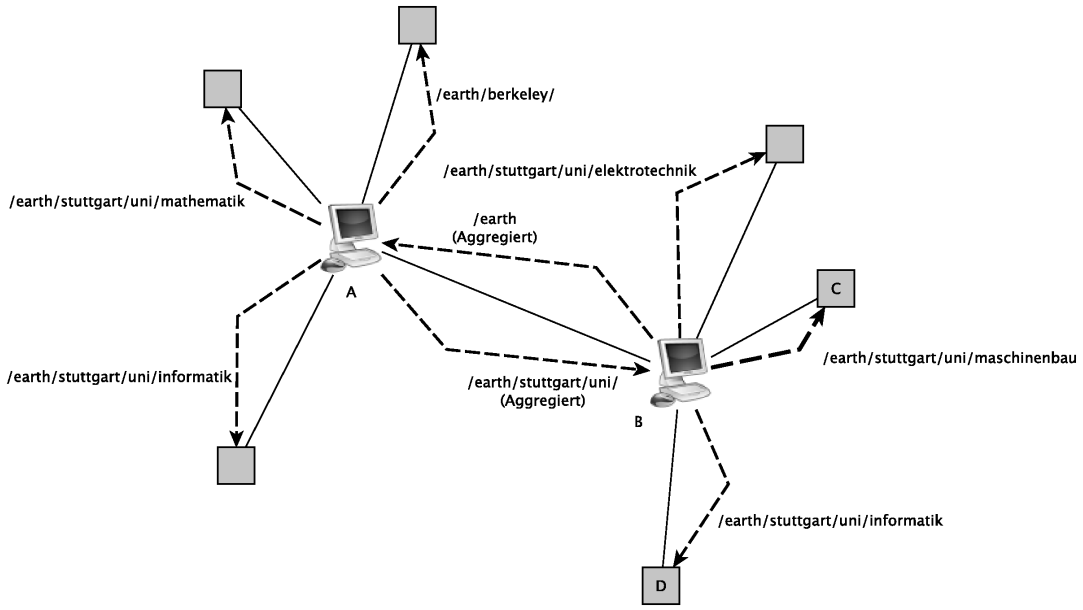


Abbildung 4.5: Catch-All-Problem bei aggregierten Routen. Durchgezogene Linien sind Kommunikationsverbindungen, gestrichelte Pfeile symbolisieren Routingtabelleneinträge.

Falls nun aus irgendeinem Grund sehr kurze Präfixe in einer Routingtabelle auftreten, fungieren diese als Catch-All-Routen. Diese kurzen Präfixe würden auf nahezu alle vorbeikommenden Nachrichten zutreffen, sodass ein enormer Datenstrom in eine – wahrscheinlich, aber nicht notwendigerweise – völlig falsche Richtung geleitet wird.

Adaptive Kompression Durch ein Schema zum adaptiven Aggregieren von Präfixen lassen sich solche unsinnigen Routen erkennen und aufspalten. Dabei wird abgewogen, ob es nun effizienter ist, genauere Routen zu propagieren und dafür weniger falsche Pakete weiterzuleiten, oder ob der Mehraufwand der genaueren Routen die falschen Pakete aufwiegt.

Dafür müssen zunächst Daten aus der Umgebung gesammelt werden. Daten über fehlgeleitete Pakete sind leicht zu erhalten. Kommen diese an einem Knoten an, der keine weitere Route für das Paket hat und auch selbst kein Adressat ist, wird ein `NACK` generiert und an den Vorgänger zurückgesendet. Das `NACK` enthält einen Hop-Count, der zu 0 initialisiert wird. Der Hop-Count wird immer beim Empfang einer Nachricht um 1 inkrementiert. Wenn ein Knoten von allen Kindknoten ein `NACK` erhalten hat, addiert er die inkrementierten Hop-Counts und sendet ein neues `NACK` mit dem summierten Hop-Count an seinen Vorgänger. Somit kann jeder Knoten leicht errechnen, wie viele falsche Pakete aus einem weitergeleiteten Paket generiert wurden: Genau $2 * \text{Hop-Count}$. Aus der im ursprünglichen Paket und auch im `NACK` mitgegebenen Zieladresse lässt sich erkennen, welcher Routingtabelleneintrag für diese falschen Pakete verantwortlich ist.

Welche Routen gegebenenfalls aufgespaltet werden müssen kann durch einen Vergleich zwischen dem verursachten Aufwand und dem erwarteten Aufwand durch das Propagieren genauerer Routen bestimmt werden. Der Aufwand zur Verteilung genauerer Routen kann durch Analyse der durch die ankommenden `TC`-Nachrichten bekannt gemachten Netzwerktopologie abgeschätzt werden: Jedes zusätzliche Paket wird durch das gesamte Netz geflutet.

Probleme der adaptiven Kompression Der Vergleich zwischen verursachtem und erwartetem Aufwand ist jedoch ein Vergleich zwischen Äpfeln und Birnen: Während die `NACK`-Nachrichten auf einem betrachteten Knoten den Aufwand im Bereich zwischen diesem Knoten und dem Zielgebiet widerspiegeln, ergibt die Analyse des Netzwerkes den Aufwand in dem aus Sicht des betrachteten Knotens vom Zielgebiet abgewandten Teil.

Es kann also die Gesamtlast im Netzwerk minimiert werden. Dabei werden jedoch Bereiche außerhalb des Zielgebietes potentiell stärker belastet. Da Knoten im Zielgebiet jedoch ohnehin an der Verteilung von Geocast-Nachrichten teilnehmen müssen, ist es effizienter, die Last soweit wie möglich im Zielgebiet zu konzentrieren.

Inhärente Routingschleifen durch fehlende Metrik Ein weiteres Problem beim Verteilen komprimierter Routen ist, dass zu einer Route als Ziel nur ein Nachbar angegeben wird, nicht etwa ein entfernter Knoten. Wenn sich ein Knoten also seinen Nachbarn als Next-Hop für einen gewissen Präfix bekannt macht und später von einem Nachbarn eine Bekanntmachung für den selben Präfix empfängt, kann der Knoten nicht entscheiden ob die neue Route zu anderen Knoten im Zielgebiet oder nur durch eine Routingschleife zu ihm selbst zurück führt. Da jeder Knoten alle empfangenen Routen weiterverbreitet, werden sich ständig Routingschleifen herausbilden. Somit würde letztendlich jede Geocast-Nachricht durch das Netz geflutet.

Dieses Szenario kann beispielsweise zustandekommen, wenn ein Knoten den ursprünglichen Broadcast der Routing-Nachricht aufgrund einer Störung nicht empfangen hat und die Route erst später von einem anderen Knoten lernt (Abbildung 4.6).

Für Multi-Path-Routing existiert keine brauchbare Metrik, die es erlaubt, die notwendige Unterscheidung zu treffen. In verdrahteten Netzen benutzt das Routing Information Protocol (RIP) den Hop-Count zum Ziel um zu identifizieren, ob eine Route reflektiert wurde oder ob es sich um eine neue Route handelt. Dies ist jedoch bei mehreren Zielen nicht möglich, da der Hop-Count zu jedem Zielknoten bzw. Zielgebiet unterschiedlich sein kann – Routen zu einer nahegelegenen Partition des Zielgebietes würden Routen zu einer entfernten Partition verdrängen. Nur wenn alle Zielknoten gleichweit entfernt sind, könnte auf diese Weise eine vollständige Abdeckung gewährleistet werden.

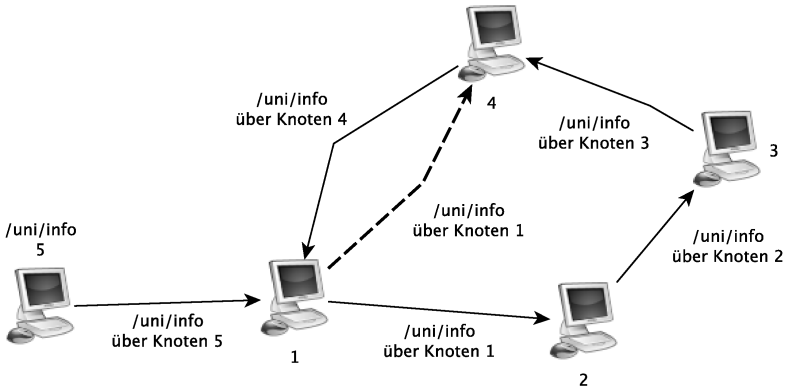


Abbildung 4.6: Propagieren von Routing-Nachrichten: Die Pfeile zeigen durch das Netz geleitete Routingnachrichten. Der gestrichelte Pfeil stellt eine verlorengegangene Nachricht dar. Knoten 1 muss für den fraglichen Präfix Routing-Einträge zu Knoten 4 und 5 einrichten und leitet somit Nachrichten zu Knoten 4 in eine falsche Richtung.

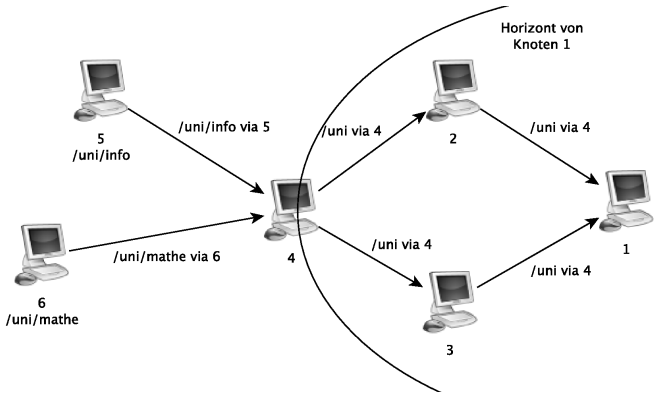
Auflösen von Routingschleifen durch erweiterten Horizont Eine Idee um das Metrik-Problem zu umgehen ist, nicht den Next-Hop als Ziel einer Route zu propagieren sondern denjenigen Knoten, der die Aggregation durchgeführt hat (siehe Abbildung 4.7). Damit können selbst nur reflektiert ankommende Routinginformationen genutzt werden, um eine Geocast-Nachricht zu routen. Betrachtet man nun jedoch mehr als einen Knoten, der Routen aussendet, offenbart sich auch hier ein kritisches Problem. Alle Knoten in der Umgebung des Zielgebietes werden Routen aggregieren und weitergeben. Ein Knoten erhält von allen diesen Knoten, die das Zielgebiet umgeben, Routen zu letzterem und kann wieder nicht entscheiden, welche der Routen letztlich zum selben Ziel führen. Somit müssen Nachrichten wieder unnötigerweise über mehrere parallele Pfade übertragen werden.

4.4.4 Fazit

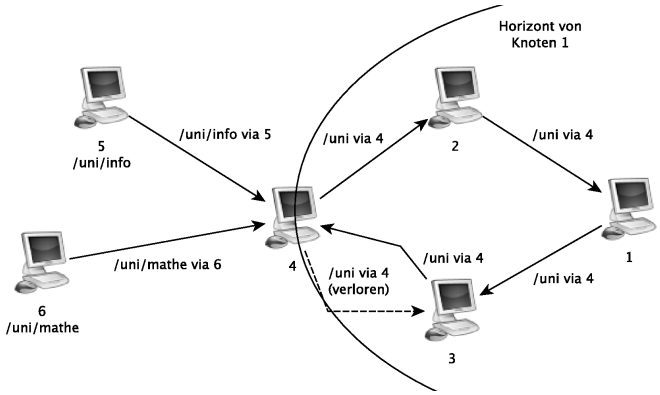
Multipath-Routing erscheint nach den ausgeführten Problemen kein geeigneter Ansatz für die effiziente Verteilung von Geocast-Nachrichten in Mesh-Netzwerken. Solche Ansätze werden daher, wie die multicastbasierten Ansätze, in dieser Arbeit nicht weiter verfolgt.

4.5 Single-Path-Routing

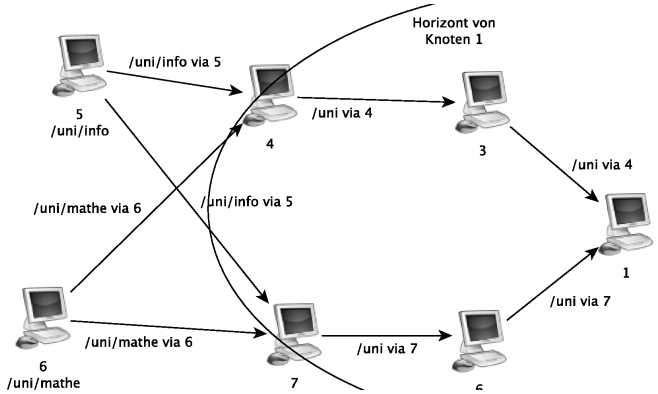
Im Unterschied zu allen bisher betrachteten Verfahren arbeiten Single-Path-Routingverfahren mehrstufig: In der ersten Phase wird nur eine einzige Route vom Sender ins Zielgebiet bestimmt. Erst in der zweiten Phase wird eine Nachricht im Zielgebiet vervielfältigt und verteilt. Diese Aufteilung ermöglicht es, die Protokolle der ersten Phase vollständig darauf



(a) Knoten 1 erkennt, dass beide Routen über Knoten 4 führen und muss nur über eine davon senden.



(b) Nachricht von Knoten 4 an Knoten 3 ging verloren. Knoten 4 erkennt, das die eingehende Nachricht eine Route über ihn selbst angibt und ignoriert die Nachricht.



(c) Knoten 1 kann wiederum nicht erkennen, dass beide Routen zu den selben Zielen führen und versendet Nachrichten daher über beide Routen.

Abbildung 4.7: Pfeile geben versendete Nachrichten an. Gestrichelte Pfeile sind verlorene Nachrichten.

auszulegen, effizient Routen zwischen entfernten Knoten im Netz – dem Geocast-Sender und einem Knoten im Zielgebiet – zu berechnen.

Protokolle für Single-Path-Routing basierend auf AODV und OLSR werden in den Abschnitten 4.5.2 und 4.5.3 beschrieben. Zunächst folgt jedoch in Abschnitt 4.5.1 die Beschreibung von geolimitiertem Broadcast, einem Algorithmus zum Fluten eines geographischen Zielgebietes, auf dem beide Protokolle für die erste Phase des Routings aufbauen.

4.5.1 Geolimitierter Broadcast

Geolimitierter Broadcast (GLBC) erlaubt es, Nachrichten innerhalb einer nicht-partitionierten Geocast-Domäne zu fluten, außerhalb dieser Domäne jedoch die Weiterleitung zu unterdrücken. Die Voraussetzung dafür ist, dass jeder Knoten, der eine GLBC-Nachricht weiterleitet, seine eigene Lokation kennt. Ein Knoten, der eine GLBC-Nachricht empfängt und sich im Zielgebiet befindet, leitet diese Nachricht an alle seine direkten Nachbarn weiter. Eine geeignete Erkennung und Eliminierung von Schleifen ist für die Verbreitung von GLBC-Nachrichten vorzusehen. Dies kann je nach den zur Verfügung stehenden Informationen z.B. ein Reverse-Path-Broadcast[1] oder eine Historie von Absender-Sequenznummern sein.

Eine strikte Beschränkung des Flutens auf das Zielgebiet kann aus mehreren Gründen zu einer schlechten Abdeckung des Zielgebietes führen. Die folgenden Absätze diskutieren die Problemstellung und beschreiben eine Erweiterung von GLBC, die diese Probleme löst.

Mikro-Partitionierung

Ein Problem ist in dünn besetzten Netzen relativ häufig zu erwarten: Geocast-Domänen, die zwar partitioniert sind, deren Partitionierung sich aber durch Einbeziehung einiger weniger Knoten außerhalb des Zielgebiets aufheben lässt. Abbildung 4.8 zeigt ein Beispiel mit einem einzelnen Knoten.

Diese Situation kann auch entstehen, wenn ein Knoten am Rand des Zielgebietes aufgrund von Ungenauigkeiten bereits die Lokation des benachbarten Gebietes angenommen hat, weil er sich z.B. in einem Durchgang befindet.

Behandlung von Knoten ohne Lokationsinformationen

Eine Voraussetzung von GLBC ist, dass jeder Knoten seine Lokation kennt. Wie in Abschnitt 4.2 beschrieben können Lokationsinformationen möglicherweise auch nachgerüstet werden. Falls keine solche Methode eingesetzt wird um Knoten ohne Lokationsinformationen damit zu konfigurieren, z.B. weil dies für das Routingprotokoll der ersten Phase nicht notwendig ist, muss dies auch bei der Verteilung per GLBC berücksichtigt werden.

[1] Eine Nachricht wird nur weitergeleitet, wenn sie über die Unicast-Route zum Absender empfangen wurde.

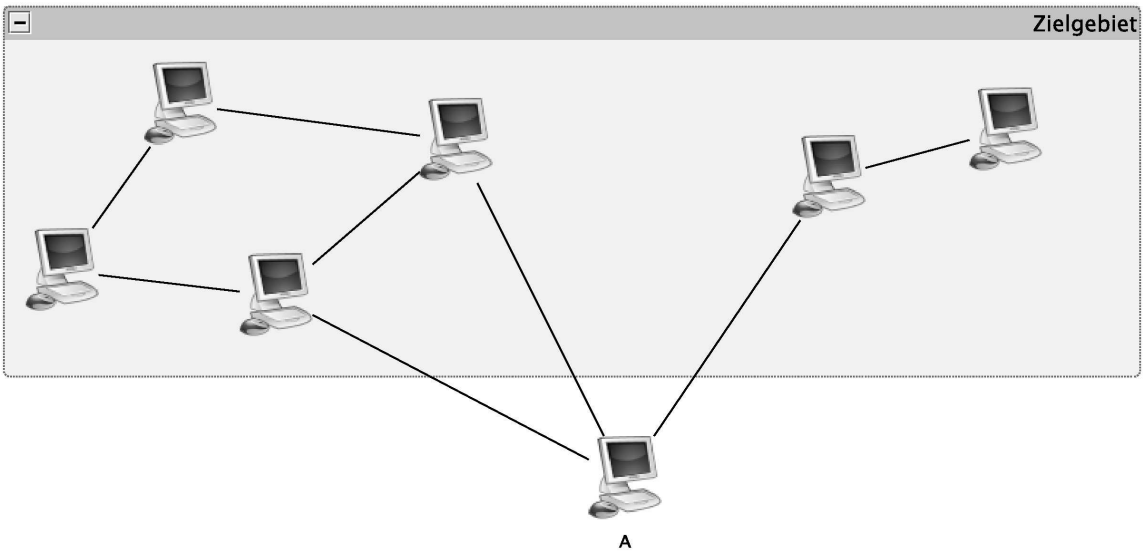


Abbildung 4.8: Notwendigkeit für limitiertes Fluten bei GLBC: Knoten A, der sich nicht im Zielgebiet befindet, ist der einzige der die Partitionierung der Geocast-Domäne aufhebt.

Knoten ohne Lokationsinfomrationen haben zwei Möglichkeiten mit GLBC-Nachrichten umzugehen. Zum einen können sie alle Nachrichten an alle Nachbarn weitergeben, d.h. für jede GLBC-Nachricht davon ausgehen, dass sie sich im Zielgebiet befinden. Dies vermeidet, dass eine Konzentration von Knoten ohne Lokationsinformationen eine Geocast-Domäne partitionieren kann. Wenn sich jedoch ein Band solcher Knoten durch das Netz zieht, kann eine Nachricht, die einen solchen Knoten erreicht, sehr weit aus dem Zielgebiet weggetragen werden und verursacht somit eine erhöhte Netzlast.

Die Alternative dazu ist, dass Knoten ohne Lokationsinformationen keine GLBC-Nachrichten weiterleiten. Dies vermeidet auch in unglücklichen Fällen den erhöhten Nachrichtenaufwand, kann jedoch zu partitionierten Geocast-Domänen führen. Die Zustellung einer Geocast-Nachricht an alle Partitionen einer Geocast-Domäne obliegt dann dem Routingprotokoll der ersten Phase.

Geolimitierter Broadcast mit Schutzzone

Um die beschriebenen Probleme abzumildern wird zu jeder Geocast-Domäne eine umgebende Schutzzone definiert. Knoten, die sich außerhalb des Zielgebietes aber noch innerhalb der Schutzzone befinden, leiten eine Nachricht weiter, als würden sie sich im Zielgebiet befinden.

Die Definition der Schutzzone findet implizit statt: Jede per GLBC verteilte Nachricht enthält eine Time-To-Live (TTL). Knoten innerhalb des Zielgebietes setzen die TTL beim Weiterleiten

einer Nachricht auf ihren Maximalwert, Knoten in der Schutzzone verringern die TTL um 1 und unterdrücken die Weiterleitung, wenn die TTL 0 erreicht.

Um den Nachrichtenaufwand optimal zu halten müsste die maximale TTL bei jeder Nachricht an die aktuellen lokalen Gegebenheiten angepasst werden. Ein Verfahren zu entwickeln, das die maximale TTL adaptiv wählen kann ist jedoch nicht Gegenstand dieser Arbeit. Daher ist hier der Maximalwert der TTL eine globale Konstante. Diese sollte typischerweise klein gewählt werden, um eine Nachricht nicht zu weit aus dem Zielgebiet heraus zu propagieren.

Es ist leicht ersichtlich, dass bei einer zu hohen Anzahl von Knoten ohne Lokationsinformationen auch GLBC mit Schutzzone als Verteilalgorithmus nicht mehr geeignet ist. Bei einer geringen Anzahl an lokationslosen Knoten erscheint es am sinnvollsten, Knoten ohne Lokation standardmäßig keine Nachrichten weiterleiten zu lassen und gegebenenfalls den Schutzradius anzupassen. Dies vermindert bzw. vermeidet den erhöhten Nachrichtenaufwand und erlaubt dennoch das überspringen kleinerer Partitionierungen.

4.5.2 Reaktiver Geocast

Das in diesem Abschnitt vorgestellte Protokoll realisiert ein reaktives Single-Path-Routing für Geocast-Nachrichten. Das Protokoll wurde in Anlehnung an AODV entwickelt, eignet sich jedoch für alle Mesh-Netze, in denen das Auffinden von Unicast-Routen einen hohen Nachrichtenaufwand bedeutet. Für das Routing von Nachrichten zum Zielgebiet ist es bei diesem Protokoll nicht erforderlich, dass Knoten, die lediglich Nachrichten weiterleiten, Lokationsinformationen besitzen.

Grundlegende Funktionsweise

Routingtabellen Jeder Knoten führt eine zusätzliche Geocast-Routingtabelle. In dieser notiert er sich Geocast-Präfixe, einen Timeout, einen Hop-Count und den Next-Hop für den jeweiligen Präfix. Der Hop-Count gibt die Entfernung zum ersten Knoten im Zielgebiet auf dieser Route an. Einträge werden aus der Tabelle entfernt, wenn der Timeout abgelaufen ist.

Routenfindung Möchte ein Sender Nachrichten per Geocast an einen Präfix versenden, zu dem er noch keine Route kennt, startet er per ERS einen Geocast-Route-Request (`GEO-RREQ`) mit dem gesuchten Präfix als Ziel. Alle Knoten die einen `GEO-RREQ` empfangen, behandeln diesen analog zu Route-Requests (`RREQ`) in Unicast-AODV, d.h. sie notieren eine Route zum Sender in ihrer Unicast-Routingtabelle und leiten den Request weiter.

Der erste Knoten, der den `GEO-RREQ` erhält und eine Route ins Zielgebiet kennt bzw. Teil des Zielgebietes ist, wird sich mit einem `GEO-RREP` beim Sender melden. Der `GEO-RREP` enthält einen Hop-Count, der die Entfernung zum ersten Knoten im Zielgebiet auf der Route angibt.

Falls der GEO-RREP von einem Zielknoten generiert wird, wird der Hop-Count mit 0 initialisiert. Falls ein Knoten, der lediglich eine Route zum Ziel kennt, den GEO-RREP generiert, wird der Hop-Count auf den Wert in der Routingtabelle des Knotens für die angefragte Lokation initialisiert. Jeder Knoten, der ein GEO-RREP weiterleitet, muss den Hop-Count um 1 inkrementieren. Knoten, die auf ein GEO-RREQ antworten, versenden zusätzlich ein GEO-RREQ-Paket per Unicast zum Zielgebiet, damit alle Knoten auf der Geocast-Route die Rück-Route zum Sender aufbauen können. Die Rück-Route wird später für Rückmeldungen über Routenabbrüche an den Sender verwendet (siehe Abschnitt „Routenabbau und Routenerhaltung“).

Knoten, die einen GEO-RREP zum Sender weiterleiten, notieren sich die Route ins Zielgebiet in der oben beschriebenen Geocast-Routingtabelle. Falls schon ein Eintrag mit dem gleichen Präfix existiert und der Next-Hop ebenfalls identisch ist, wird der Timeout erneuert. Falls der Next-Hop unterschiedlich ist, wird diejenige Route mit dem geringeren Hop-Count ausgewählt.

Nachrichtenverteilung Geocast-Nachrichten können nach dem Eingang des GEO-RREP beim Sender entlang der berechneten Route gesendet werden. Sobald ein Knoten des Zielgebietes erreicht wird, wird die Nachrichtenverteilung per geolimitiertem Broadcast fortgesetzt.

Routenabbau und Routenerhaltung Alle Routingeinträge sind, wie in Unicast-AODV vorgesehen, mit Timeouts versehen. Wenn dieser Timeout abläuft, wird ein Routingeintrag entfernt und die Route somit abgebaut. Zur Routenerhaltung muss der Sender periodisch GEO-RREQs per Unicast über die vorhandene Route senden.

Eine Geocast-Route-Delete-Nachricht (GEO-RDEL) kann verwendet werden, um eine Route proaktiv entfernen zu lassen. Wird ein GEO-RDEL empfangen oder weitergeleitet, wird die Route zum angegebenen Präfix gelöscht, falls das GEO-RDEL über den Next-Hop dieser Route empfangen wurde.

Routenoptimierung Wie in Abbildung 4.9 zu sehen ist, werden bei der Routenfindung möglicherweise „eckige“ Routen gefunden. Sender A hat eine Route zum Zielgebiet aufgebaut (dicke Pfeile). Sender B sucht per ERS (Kreis) eine Route zum Zielgebiet und findet dabei auf Knoten X eine bestehende Route. Über diesen können Nachrichten ins Zielgebiet gesendet werden (dünne Pfeile). Eine insgesamt deutlich kürzere Route wäre jedoch der direkte Weg über Knoten U (gestrichelte Pfeile), der jedoch nicht gefunden wurde.

Um solche Routen zu verbessern, werden GEO-RREQ-Nachrichten zur Routenerhaltung von allen Knoten, die sie passieren, per limitiertem Broadcast mit geringer TTL an die nähere Umgebung weitergegeben. Knoten, die die neuerlichen GEO-RREQ-Nachrichten empfangen, beantworten diese wie auch bei der regulären Routensuche mit GEO-RREP-Paketen. Wenn nun bei einem Knoten ein GEO-RREQ mit einem geringeren Hop-Count als der bestehende

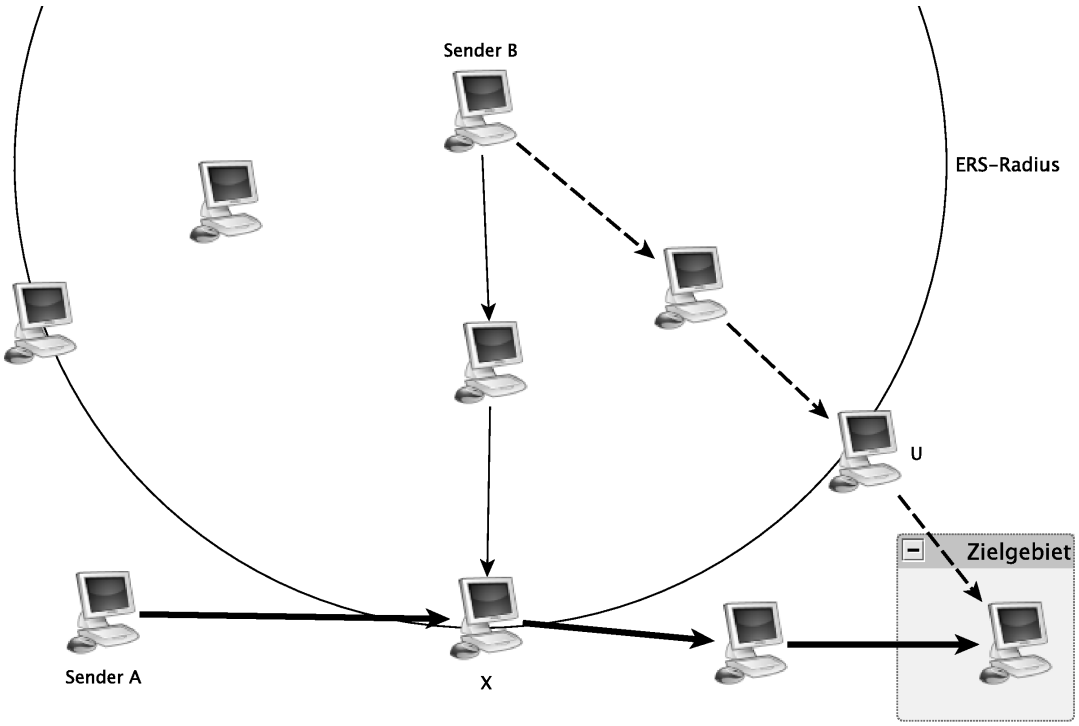
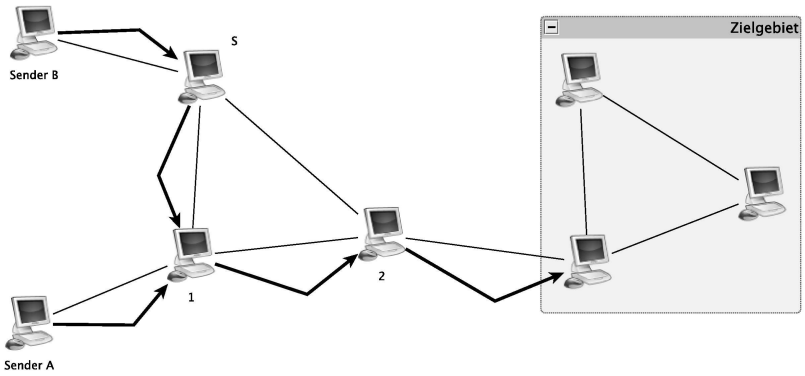


Abbildung 4.9: Problematik der suboptimalen Routen („Eckenproblem"). Dicke Pfeile: Existierende Route von Sender A. Dünne Pfeile: Gefundene Route von Sender B. Gestrichelte Pfeile: Nicht gefundene, optimale Route von Sender B.

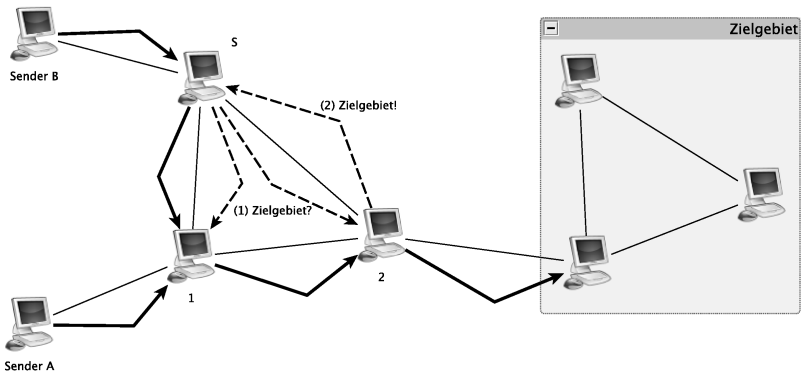
Routeneintrag eintrifft, wird die so gefundene verkürzte Route genutzt. Abbildung 4.10 illustriert dieses Verfahren.

Erkennung von Routenabbrüchen Ein Abbruch einer Route wird erkannt, indem entweder ein Knoten eine Nachricht nicht an den Next-Hop weiterleiten kann oder indem die Nachricht einen Knoten erreicht, der sich weder im Zielgebiet befindet noch eine Route dorthin kennt. Ein solcher Knoten wird zunächst davon ausgehen, dass lediglich eine kleine Lücke in der Route existiert und per limitiertem Fluten eines `GEO-RREQ` mit kleiner TTL versuchen, diese Lücke zu überbrücken und die Route wieder aufzunehmen. Falls die Route nicht wieder aufgenommen werden kann, z.B. weil kein Knoten in Richtung des Zielgebietes mehr erreichbar ist oder sich kein Knoten mehr im Zielgebiet befindet, wird ein `GEO-RDEL` zum Sender zurückgeleitet. Der Sender startet bei Empfang der `GEO-RDEL`-Nachricht gegebenenfalls eine erneute Routensuche.

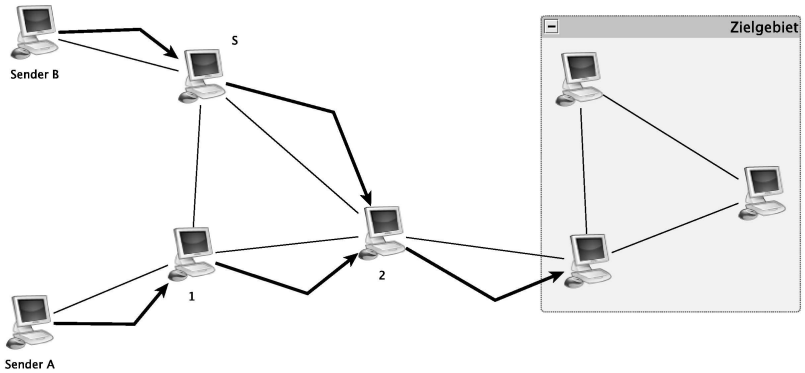
Die `GEO-RDEL`-Nachricht löscht nur Routingeinträge auf der Route eines einzelnen Senders. Falls mehrere Sender eine Route gemeinsam benutzen, wird der Sender, der noch kein `GEO-RDEL` erhalten hat, beim nächsten Sendeversuch einen solchen vom ersten gemeinsam genutzen Knoten erhalten, da auf diesem Knoten keine Route zum Ziel mehr existiert.



(a)



(b)



(c)

Abbildung 4.10: Optimierung von Geocast-Routen: (a) Sender A hat eine Route aufgebaut, Sender B hat später einen Weg über diese Route zum Ziel gefunden. (b) Knoten S sucht durch limitierten Broadcast nach einer besseren Route. Knoten 2 beantwortet die Anfrage. (c) Knoten S versendet Geocast-Pakete ins Zielgebiet nun über Knoten 2 statt Knoten 1.

Bekannte Einschränkungen

Dieses Protokoll hat keine Möglichkeit, partitionierte Geocast-Domänen zu erkennen oder zu beheben, da per ERS stets nur die nächstliegende Partition bzw. die nächstliegende bestehende Route gefunden wird. Um eine Route zwischen Sender und Zielgebiet zu bestimmen, müssen nur die Zielknoten Lokationsinformationen besitzen. Falls jedoch zu viele Knoten im Zielgebiet keine Lokationsinformationen besitzen, können diese, wie in den Problemen von GLBC in Abschnitt 4.5.1 beschrieben, eine Partitionierung der Geocast-Domäne verursachen.

Die gefundenen Routen sind nicht notwendigerweise optimal (Eckenproblem, siehe Abbildung 4.9). Bei kurzen Geocasts mit wenigen Paketen fällt dies nicht ins Gewicht - die Einsparung an Nachrichten beim Auffinden der Route amortisieren diesen Zusatzaufwand. Bei länger andauernden Geocasts kann die Route durch das in Abschnitt 4.5.2 beschriebene Verfahren zur Routenoptimierung verbessert werden.

Die Routenoptimierung arbeitet jedoch nur sehr eingeschränkt: Eine Route mit möglicherweise verbesserter Verbindungsqualität und damit auch höherer Zuverlässigkeit und besserem Durchsatz kann kaum gefunden werden, es wird lediglich der Hop-Count ein wenig verringert. Selbst diese Optimierung ist jedoch wenig effektiv. Auch bei fortlaufender Routenoptimierung kann die Route insgesamt lediglich um den Radius des limitierten Flutens verkürzt werden. Die Routenoptimierung bringt also nur dann wirkliche Verbesserungen, wenn der Radius des limitierten Flutens sehr groß gewählt wird, d.h. wenn große Netzbereiche geflutet werden. Stochastische Verfahren zur Wahl des Flutungsradius können dieses Problem möglicherweise lösen. Dazu sind jedoch gesonderte Untersuchungen notwendig, die im Rahmen dieser Arbeit nicht weiter betrachtet werden. Da das Verfahren zur Routenoptimierung keine Voraussetzung für andere Protokollkomponenten ist, kann es als optional angesehen werden.

Der Nachrichtenaufwand entspricht im Worst-Case noch immer dem einer Flutung. Bei der angenommenen Verkehrscharakteristik – wenige starke, im Netz verteilte, Sender – steht zu erwarten, dass ein `GEO-RREQ` den Großteil des Netzes berührt, bis eine Route gefunden wird.

Die Erkennung von Routenabbrüchen kann durch unglückliches Timing fehlschlagen, d.h. der Sender wird nicht über einen Routenabbruch benachrichtigt. Bei Geocasts, die aus mehreren Paketen bestehen, kann dies dazu führen, dass der Sender eine Zeit lang das Netz mit Paketen belastet, die keine Chance haben im Zielgebiet anzukommen.

Deutlich wird dies an folgendem Beispielszenario: Knoten S möchte eine Geocast-Nachricht versenden, Knoten X liegt auf der Route ins Zielgebiet, befindet sich jedoch außerhalb des Zielgebietes. Knoten X wird bei der Routenfindung zu einem Zeitpunkt t ein `GEO-RREQ`-Paket erhalten und eine Route zu S in seine Unicast-Tabelle eintragen. Wurde die Route gefunden, folgt zum Zeitpunkt $t_{rep} = t + \delta t_{rep}$ ein `GEO-RREP`-Paket von Knoten S an Knoten X. Zum Zeitpunkt $t_{data} = t + \delta t_{rep} + \delta t_{data}$ empfängt X das `GEO-DATA`-Paket. Wenn zu diesem Zeitpunkt die weitergehende Route bereits abgerissen ist, z.B. weil sich der nächste Knoten

aus dem Empfangsbereich von X herausbewegt hat, versucht X nun eine lokale Routen-Wiederherstellung mit Dauer $\delta t_{recover}$. Falls $\delta t_{rep} + \delta t_{data} + \delta t_{recover} > \textit{Unicast-Routentimeout}$ gilt, wurde die Unicast-Route zu S von X bereits gelöscht und das `GEO-RDEL` kann nicht zugestellt werden.

Erst wenn Nachrichten zur Routenerhaltung versendet werden, wird die Rückroute wieder aufgebaut, so dass das nächste `GEO-DATA`-Paket eine `GEO-RDEL`-Antwort auslösen kann.

Um dies zu beheben, kann die letzte Ungleichung optimiert werden. Wenn auch `GEO-DATA`-Pakete Unicast-Routen auffrischen, vereinfacht sich die Ungleichung zu $\delta t_{recover} > \textit{Unicast-Routentimeout}$. Somit verbleibt mehr Zeit für den Recovery-Algorithmus. Wird der Timeout für den Recovery-Mechanismus kleiner als der Unicast-Routingtimeout gewählt, kann dieses Problem nicht mehr auftreten.

4.5.3 Hierarchischer Geocast

In Abschnitt 2.5.1 wurde ein Konzept für Geocast-Routing mittels eines Overlay-Netzwerkes vorgestellt. Dieses lässt sich in Mesh-Netze einbetten. Insbesondere das Vorhandensein relativ stabiler und ortsfester Mesh-Router ermöglicht eine zuverlässige Verteilstruktur.

Voraussetzung für eine effiziente Arbeitsweise dieses Protokolls ist, dass Unicast-Routen ohne besondere Belastung des Netzwerkes, wie z.B. netzweites Fluten, aufgebaut werden können. Weiterhin müssen Knoten in der Lage sein, Geocast-Nachrichten, die eigentlich zur Weiterleitung mittels Unicast bestimmt sind, abzufangen und selbst zu verarbeiten.

Wie bei reaktivem Geocast (Abschnitt 4.5.2) ist es auch bei hierarchischem Geocast nicht unbedingt notwendig, dass weiterleitende Knoten ihre Lokation kennen. Der Einfluss von Knoten ohne Lokationsinformationen auf Effektivität und Effizienz des Protokolles wird im Abschnitt „Bekannte Einschränkungen“ untersucht.

In Anlehnung an die in [DR06, DR08] vorgestellte Geocast-Routinginfrastruktur wird ein Routerbaum entworfen. Der Aufbau geschieht durch Advertisement- und Solicitation-Nachrichten. Im Grunde sendet jeder Knoten, der keinen übergeordneten Knoten kennt, so lange Solicitation-Nachrichten aus, bis er eine Advertisement-Nachricht empfängt.

Grundlegende Funktion

Bei hierarchischem Geocast wird ein baumförmiges Overlay-Netz aus Designated Routern (DR) aufgebaut. Die Hierarchie der DRs entspricht dabei der Hierarchie der Präfixe der Lokationen. DRs auf einer Hierarchiestufe werden stets aus den DRs der darunterliegenden Stufe gewählt. Die DRs der untersten Stufe werden für jede Lokation aus der Menge der regulären Mesh-Knoten, die sich an der jeweiligen Lokation befinden, gewählt.

Um Geocast-Nachrichten zu versenden, leitet ein Knoten diese an den DR für seinen Präfix weiter. Dieser DR sendet die Nachricht durch den DR-Baum, bis ein Knoten im Zielgebiet erreicht wird. Von dort wird die Nachricht per GLBC im Zielgebiet weiterverteilt.

Routingtabellen Knoten, die sich als Designated Router (DR) etablieren, führen eine Routingtabelle über andere bekannte DRs. Einträge der Routingtabelle bestehen aus dem Präfix des Ziel-DRs, seiner Unicast-Adresse und einem Timeout. Läuft der Timeout eines Eintrags ab, wird er aus der Tabelle entfernt.

Die Routingtabelle enthält stets mindestens die Einträge für die einem Knoten über- und untergeordneten DRs. Zusätzlich können noch Links zu beliebigen anderen DRs enthalten sein.

Wahl der Blatt-DR Der Routerbaum wird bottom-up aufgebaut. Knoten warten zunächst auf ein von einem DR periodisch per GLBC versendetes Designated Router Advertismenet (`DRA`). Falls eine gewisse Zeit lang kein für ihre Lokation passendes `DRA` eintrifft, beginnen die Knoten damit periodisch Designated-Router-Solicitation-Nachrichten (`DRS`) per geolimitertem Broadcast zu senden. Falls bereits ein DR für diese Lokation existiert, wird er damit zum Aussenden eines `DRA` aufgefordert.

Falls auf ein `DRS` kein `DRA` innerhalb angemessener Zeit folgt, wählen die Knoten im Zielgebiet einen neuen DR. Dazu sendet jeder Knoten `DRS`-Nachrichten aus, um sich allen anderen Knoten im Zielgebiet bekannt zu machen. Aus dieser Datenbasis kann jeder Knoten lokal den neuen DR des Zielgebietes berechnen, wie im nächsten Abschnitt („Auswahlalgorithmus“) angegeben. Hat sich ein Knoten selbst als DR identifiziert, beginnt er umgehend mit dem Aussenden von `DRA`-Nachrichten per geolimitiertem Broadcast.

Falls unter den zur Auswahl stehenden Knoten Mesh-Router sind, werden diese bevorzugt ausgewählt. Mesh-Clients können aufgrund ihrer Mobilität die Lokation verlassen und stehen dann nicht mehr als DR zur Verfügung. Mesh-Router bieten entsprechend einen stabileren Dienst. Da Neuwahlen nur ausgeführt werden, wenn kein DR verfügbar ist, werden Mesh-Clients, die sich über längere Zeit an der Lokation befanden, mit hoher Wahrscheinlichkeit irgendwann als DR gewählt und diese Rolle dann auch behalten. Somit werden auch, falls keine Mesh-Router verfügbar sind, statische Knoten bevorzugt zu DRs. Unnötige Neuwahlen des DRs werden so vermieden.

In einer Implementierung sollten die Sendeperiode für `DRS`-Nachrichten sowie der Timeout für den Wahlalgorithmus so gewählt werden, dass die Wahrscheinlichkeit für eine fehlerhafte Wahl relativ gering bleibt.

Auswahlalgorithmus Der Algorithmus zur Auswahl eines Knoten als DR aus einer Menge bekannter Knoten muss eine konsistente Wahl ermöglichen. Sei d der Knoten, der nach dem Algorithmus aus der Menge K der zur Wahl stehenden Knoten ausgewählt wird, so muss d auch aus jeder Teilmenge G der Knoten ausgewählt werden, sofern er in dieser Teilmenge enthalten ist. Ein Auswahlalgorithmus, der diese Bedingung erfüllt, besteht darin, den Knoten mit der größten Adresse zu wählen.

Nach dieser Konsistenzeigenschaft kann ein Knoten nur dann einen anderen Knoten als d als DR auswählen, wenn er keine Wahlnachrichten von d empfangen hat, d.h. d nicht kennt. Es wird also nicht garantiert, dass alle Knoten den gleichen Knoten als DR auswählen,

jedoch minimiert die Konsistenzeigenschaft das Potential, einen falschen Knoten als DR zu wählen.

Im Falle von partitionierten Geocast-Domänen muss die Konsistenzeigenschaft nur in jeder Partition erfüllt werden. Der im Abschnitt „Aufbau des Routerbaumes“ beschriebene Proxying-Ansatz sorgt dafür, dass das Overlay-Netz dennoch korrekt arbeitet und alle Partitionen erreicht werden. Steht kein Proxying-Mechanismus zur Verfügung, ist nur eine Partition der Geocast-Domäne erreichbar, was laut den in Kapitel 3 aufgestellten Anforderungen ausreichend ist.

Eine Wahl kann zu einem inkonsistenten Ergebnis führen. Dabei haben einzelne Knoten den Knoten e als DR ausgewählt, obwohl nach der Konsistenzbedingung in ihrer Partition ein anderer Knoten d hätte ausgewählt werden müssen. Dies kann zustande kommen, falls `DRS`-Nachrichten von d die inkonsistent wählenden Knoten nicht erreicht haben. Dieses Problem löst sich jedoch nach einer gewissen Zeit von selbst: d versendet wie bereits beschrieben periodisch `DRA`-Nachrichten per GLBC. Diese werden irgendwann auch die Knoten erreichen, die inkonsistent gewählt haben. Diese Knoten können ihr Wissen über den DR ihrer Partition dann anhand der in der `DRA`-Nachricht enthaltenen Daten korrigieren.

Aufbau des Routerbaumes Übergeordnete DRs können per `DRS` gefunden werden. Diese werden per GLBC an den um eine Komponente verkürzten Präfix des suchenden DR durch das Netz gesendet. Erreicht die Nachricht den übergeordneten DR, meldet sich dieser mit einem `DRA` per Unicast bei dem neuen DR.

Dabei kann es vorkommen, dass ein übergeordneter DR r_{up} bereits einen anderen DR r' für den Präfix des anfragendenden DR r kennt. Dieses Problem tritt auf, wenn eine Geocast-Domäne partitioniert ist und in jeder Partition getrennt ein DR gewählt wurde. Zur Auflösung dieser Situation wählt r_{up} aus allen bekannten untergeordneten DRs einen aus, der in Zukunft als DR für diesen Präfix fungieren soll. Weiterhin leitet er die `DRA`-Nachricht von r weiter an r' und die von r' an r. Dadurch werden die Knoten gegenseitig über ihre Existenz benachrichtigt. Der nicht als DR gewählte Knoten wird dazu veranlasst, sich als Designated Proxy (DP) beim DR anzumelden. DPs bilden keine Hierarchie, sondern sind alle direkt dem DR für ihren Präfix unterstellt.

Da ein DR eine Nachricht an alle untergeordneten Proxies vervielfältigen muss, kann mit Proxies eine Vervielfältigung von Nachrichten auch außerhalb des Zielgebietes auftreten. Nach der eingangs aufgebauten Taxonomie wäre hierarchischer Geocast damit ein Multi-Path-Ansatz. Da der Proxy-Mechanismus jedoch lediglich eine Protokollerweiterung ist, d.h. da das Protokoll in Abwesenheit von partitionierten Geocast-Domänen auch ohne Proxies arbeiten kann, wird es dennoch als Single-Path-Routingprotokoll klassifiziert.

Antwortet kein DR auf ein `DRS`, wird davon ausgegangen, dass kein DR für den fraglichen Präfix existiert. Alle Geschwister-DRs würden durch weiter unten vorgestellte Mechanismen ebenfalls feststellen, dass kein übergeordneter DR mehr vorhanden ist. Nach dieser Feststellung beginnen nun alle DRs auf dem aktuellen Level periodisch `DRS`-Nachrichten per GLBC auszusenden. Dadurch werden jedem DR alle seine Geschwister-DRs bekannt gemacht.

Nach demselben Algorithmus wie bei der Wahl der Blatt-DR wird aus den Geschwistern ein DR als übergeordneter DR identifiziert, der daraufhin bei allen Geschwistern in die Routingtabellen eingetragen wird.

Verwaltung des Routerbaumes Jeder DR sendet periodisch per Unicast `DRA`s an seinen übergeordneten und alle untergeordneten DRs und DPs. Bleibt eine erwartete Nachricht eine Zeit lang aus, wird davon ausgegangen, dass der betroffene DR nicht mehr erreichbar ist. Falls es sich dabei um den übergeordneten DR handelt, werden `DRS`-Nachrichten mit dessen Präfix per GLBC versendet. Da bei einem verschwundenen übergeordneten DR auch alle Geschwisterknoten `DRS`-Nachrichten per GLBC starten, wird somit festgestellt welche DRs in Frage kommen, um die Stelle des übergeordneten DR zu übernehmen. Die Wahl findet nach denselben Kriterien wie bei der Wahl der Blatt-DR und, im Fall partitionierter Geocast-Domänen, dem Aufbau des Routerbaumes statt.

Verschwindet ein untergeordneter DR oder DP, entfernt der übergeordnete Knoten lediglich den Routingtabelleneintrag. Ein Ersatz für den bisherigen DR wird bei Bedarf wieder bottom-up bestimmt. Der neue DR wird dann selbstständig nach einem übergeordneten DR suchen und auf den bereits zuvor übergeordneten Knoten stoßen.

Routing von Geocast-Nachrichten

Sender im Zielgebiet Falls der Sender einer Geocast-Nachricht bereits im Zielgebiet sitzt, kann er eine Nachricht einfach per geolimitiertem Broadcast an alle anderen Knoten im Zielgebiet verteilen.

Sender außerhalb des Zielgebietes Ein Sender außerhalb des Zielgebietes leitet eine Geocast-Nachricht zu dem für seine aktuelle Lokation zuständigen DR. Dieser sendet die Nachricht weiter durch den DR-Baum. Als Ziel für eine Weiterleitung im Baum wird immer der DR mit dem längsten zutreffenden Präfix ausgewählt. Ist kein solcher verfügbar, wird die Nachricht an den übergeordneten DR weitergereicht. Erreicht die Nachricht einen beliebigen Knoten des Zielgebietes – dabei muss es sich nicht um den Ziel-DR handeln –, wird die Nachricht via GLBC weiterverteilt.

Handhabung auf DR-Proxies Ist ein DR ein Proxy, muss für vom übergeordneten DR erhaltene Nachrichten keine gesonderte Behandlung der Geocast-Nachrichten erfolgen. Kommt eine Nachricht vom übergeordneten DR an, wird diese regulär an untergeordnete Knoten weitergeroutet. Nachrichten, die von untergeordneten Knoten erhalten wurden, müssen jedoch in jedem Fall zusätzlich zur normalen Weiterleitung an den übergeordneten DR weitergegeben werden, da sie möglicherweise auch über andere Proxy-DRs in weitere Partitionen der Geocast-Domäne gesendet werden müssen.

Der übergeordnete DR muss alle eingehenden Nachrichten an alle untergeordneten DPs kopieren. Einzige Ausnahme ist hierbei der Knoten, von dem die Nachricht empfangen wurde, falls es sich bei diesem um einen untergeordneten Proxy-DR handelt.

Abkürzungen im Baum

Die Route die ein Geocast-Paket durch den Routerbaum nimmt, entspricht im Allgemeinen nicht der optimalen Route vom Sender zum Zielgebiet. Um Routen zu häufig adressierten Lokationen zu optimieren, können jederzeit wie in [DR08] vorgesehen Abkürzungen in den Baum eingefügt werden.

Statische Abkürzungen wie in [DR06] vorgeschlagen wären zwar prinzipiell möglich, werden für diese Arbeit jedoch nicht als sinnvoll erachtet. Dabei baut jeder Knoten Verbindungen zu allen Geschwisterknoten seines Vaterknotens und zu seinem Großvaterknoten auf. Um dies zu ermöglichen, muss ein Knoten jedoch proaktiv Wissen über entfernte DRs erlangen, was zwangsweise mit zusätzlichem Nachrichtenaufwand einhergeht.

Für alle DRs besteht die Möglichkeit, adaptiv Abkürzungen aufzubauen. Präfixe, die Ziel einer Abkürzung sein können, müssen stets zuerst in einer weitergeleiteten Geocast-Nachricht erwähnt worden sein. Daher ist kein zusätzlicher Nachrichtenaufwand notwendig um gewünschte Abkürzungen zu bestimmen. Lediglich zum Aufbau der Abkürzung ist ein Designated Router Shortcut Solicitation-Paket (`DRSS`) notwendig, das über den DR-Baum per Unicast versendet wird. Das Antwortpaket ist bereits ein `DRA` und kann somit zur ohnehin notwendigen Aufrechterhaltung der Abkürzung gezählt werden.

Zur Bestimmung der Ziele von Abkürzungen ordnet ein DR andere DRs anhand einer Bewertungsfunktion in drei Klassen ein. In diese gehen die durch eine Abkürzung zu diesem DR erwartete Einsparung an Hops im Overlay-Netz sowie die Häufigkeit der Kontaktierung und die Zeit seit der letzten Kontaktierung ein. Zu den DRs in der höchsten Klasse werden Abkürzungen aufgebaut. Die Anzahl der Abkürzungen wird jedoch durch eine Konstante eingeschränkt, um den von DRs zu haltenden Zustand zu begrenzen.

Im ursprünglich vorgestellten Protokoll wurden unidirektionale Abkürzungen verwendet. In der Variante für Mesh-Netze werden jedoch bidirektionale Verbindungen genutzt: Da ohnehin eine Meldung in beide Richtingen zwischen den beteiligten DRs erforderlich ist, wird die Abkürzung in die Gegenrichtung ohne zusätzliche Kosten aufgebaut.

Damit bereitet der Abbau einer Abkürzung ein Problem: Da ein Router empfangene `DRA`s stets in seine Routingtabelle aufnimmt und die Ziele der vorhandenen Einträge fortlaufend mittels `DRA`s über seine Lebendigkeit informiert, müssten beide Enden einer Abkürzung simultan beschließen, eine solche Abkürzung abzubauen. Andernfalls würde ein eintreffendes `DRA` des jeweils anderen Knotens eine sofortige Wiederaufnahme der Abkürzung bewirken.

Zur Lösung wird festgelegt, dass derjenige Knoten, der den Shortcut aufgebaut hat, diesen auch verwaltet. Ein zusätzliches Flag in den Einträgen der Routingtabelle gibt an, ob es sich bei dem im Eintrag angegebenen Knoten um den Ausgangspunkt einer Abkürzung handelt. Falls ja, werden `DRA`s für diesen Eintrag nur dann gesendet, wenn ein `DRA` des fraglichen

Knotens eingeht. DRA-Nachrichten werden ebenfalls um ein Flag erweitert, das angibt, ob es sich um ein DRA eines Ausgangspunkts oder eines Ziels einer Abkürzung oder um ein reguläres DRA innerhalb des Routerbaumes handelt.

Dadurch, dass DRAs in eine Richtung nur reaktiv versendet werden, können Knoten, die eine Abkürzung angefordert haben, diese nun unilateral durch Unterdrückung weiterer DRA-Nachrichten abbauen.

Weitere Optimierungen

OLSR sieht zur Optimierung von Flutungen das Konzept der Multi-Point-Relays vor. Wenn dieses oder ähnliche Optimierungen zur Verfügung stehen, können diese für sämtliche GLBC-basierten Verteilungen wie die Wahl von DRs, Verteilen von DRAs an Basislokationen oder Verteilen von Geocast-Nachrichten im Zielgebiet genutzt werden.

Falls ein Sender einen voraussichtlich lange andauernden Geocast versenden möchte, kann er, statt allen Datenverkehr über den DR-Baum zu routen, auch auf Hinweis der sendenden Anwendung eine adaptive Abkürzung aufbauen. Anschließend können Geocast-Pakete per Unicast zum DR des Zielgebietes transportiert und dort per GLBC weiterverteilt werden.

Hört ein DR per GLBC verteilte DRAs eines anderen DR, so kann er opportunistische Abkürzungen zu den Präfixen des anderen DRs aufbauen.

Bekannte Einschränkungen

Für die Wahl der DR müssen insbesondere in oberen Baumebenen sehr große Bereiche des Netzwerkes geflutet werden. Ein DR muss allerdings relativ selten neu gewählt werden, da bevorzugt die nahezu ständig verfügbaren Mesh-Router als DR gewählt werden. Falls keine neuen Lokationen im Netz erscheinen oder DRs ausfallen, findet die Wartung der DR-Hierarchie ausschließlich über Unicast statt. Daher ist der Flutungsaufwand insgesamt vertretbar.

Nach dem Wahlalgorithmus werden übergeordnete DRs immer aus der Menge der untergeordneten DRs gewählt. Der Knoten, der später die Wurzel des Baumes darstellt, stellt also gleichzeitig auf jeder Ebene einen DR dar. Dies kann in zu großen Netzen die verfügbaren Ressourcen eines Routers übersteigen. Für die angenommene Netzgröße von mehreren tausend Knoten ist dies jedoch kein Problem.

Designated Proxies können das Aufheben einer Partitionierung einer Geocast-Domäne nur dann erkennen, wenn sie per GLBC eine Nachricht von ihren Geschwister-DPs oder ihrem übergeordneten DR erhalten. DRAs per GLBC werden jedoch nur auf der untersten Ebene der DR-Hierarchie versendet. Ansonsten werden nur DRS-Pakete für DR-Wahlen per GLBC verteilt. Damit ergibt sich, dass sich eine Proxy-Hierarchie nach Aufhebung einer Partitionierung in den meisten Fällen erst dann auflösen wird, wenn der übergeordnete DR oder mindestens einer der DPs ausfällt.

Um das Problem der Ballung von DR-Rollen auf einem Mesh-Router zu vermeiden, muss die Menge der Knoten, die als Kandidaten für einen DR dienen, erweitert werden. Prinzipiell kann dies dadurch geschehen, dass nicht nur DRs, sondern alle Knoten, die einen GLBC für einen übergeordneten Präfix erhalten und selbst noch kein DR sind, ebenfalls an der Wahl teilnehmen. Allerdings ist dafür ein anderer Wahlalgorithmus notwendig, der die Konsistenzeigenschaft erfüllt.

Wie eingangs erwähnt ist es nicht zwingend notwendig, dass jeder Netzknoten seine Lokation kennt. Soll kein Verfahren zum automatischen Lernen von Lokationen (siehe Abschnitt 4.2) eingesetzt werden, können jedoch Knoten ohne Lokationsinformationen die Funktionalität des Netzes auf mehrere Weisen beeinträchtigen. Da laut Systemmodell Mesh-Clients stets ihre Lokation kennen, werden es ausschließlich Mesh-Router sein, die ohne Lokation im Netz auftreten. Ohne Lokationsinformationen kann ein Knoten nicht an der Wahl für DRs teilnehmen und somit auch selbst nicht als DR auftreten. Damit fällt die Rolle des DR einem Client zu. Zum einen wird dadurch die Verfügbarkeit des DR für die aktuelle und möglicherweise auch übergeordnete Lokationen unnötig beeinträchtigt, zum anderen wird es häufiger notwendig den DR neu zu wählen, was wiederum die Netzlast erhöht.

Der Einflussbereich eines DRs im Mesh-Netz wird bei hierarchischem Geocast implizit durch GLBC bestimmt. Sind im Netz Knoten ohne Lokationsinformationen vorhanden, können durch unpassend gewählte Schutzzonen Geocast-Domänen als partitioniert angesehen werden. Mittels Proxy-DRs können solche Partitionierungen zwar überbrückt werden, jedoch geht dies durch einen erhöhten Nachrichtenaufwand und nicht-optimale Routen zulasten der Effizienz.

4.5.4 Fazit

Für Protokolle, die auf netzweite Flutungen setzen, ist eine schlechte Skalierbarkeit zu erwarten. Reaktiver Geocast auf Basis von AODV versucht die Flutungen zu begrenzen, jedoch muss im Worst-Case dennoch für jeden Geocast das gesamte Netz geflutet werden. Da aber Unicast-AODV für fast jede Unicast-Route das Netz vollständig flutet, steht zu erwarten, dass reaktiver Geocast ohnehin nicht in wirklich großen Netzen eingesetzt wird.

Bei hierarchischem Geocast ist eine deutlich bessere Skalierbarkeit zu erwarten. Wenn ausreichend stabile Knoten vorhanden sind, muss lediglich beim Aufbau der Router-Hierarchie netzweit geflutet werden. Änderungen der Netztopologie, z.B. verursacht durch mobile Knoten, erfordern dann nur noch ein lokal beschränktes Fluten. Theoretisch kann hierarchischer Geocast auch mit einer größeren Menge an Knoten ohne Lokationsinformationen arbeiten. Da die Behebung der dabei auftretenden Partitionierungen sowie die häufigere Neuwahl von DRs von Geocast-Domänen die Effizienz senkt, ist für reale Anwendungen davon auszugehen, dass nur vereinzelte Knoten ohne Lokationsinformationen toleriert werden können.

5 Implementierung

Zur Evaluierung wurden beide entwickelten Geocast-Routingprotokolle in Simulationen implementiert. Abschnitt 5.1 beschreibt Basisalgorithmen und Funktionalitäten, die beiden Implementierungen gemein sind. In den Abschnitten 5.2 und 5.3 werden die spezifischen Teile der Implementierungen von reaktivem und hierarchischem Geocast erläutert.

Die Simulationen wurden mit dem Netzwerksimulator ns2, Version 2.34, durchgeführt. ns2 besteht aus einer Mischung von C++- und TCL-Code. Während die eigentliche Simulation in C++ implementiert ist, dienen TCL-Skripte dazu, die Simulation zu konfigurieren und den Simulationsablauf zu steuern.

Eine AODV-Implementierung ist in ns2 bereits vorhanden. Als OLSR-Implementierung kam das RFC-konforme UM-OLSR 0.8.8 [Ros09] zum Einsatz.

Da ns2 gravierende Schwächen bei der Simulation von Broadcast-Nachrichten für andere Zwecke als zum Routing hat, mussten die Geocast-Implementierungen direkt in den AODV- bzw. OLSR-Routingcode eingebettet werden. Soweit möglich, wurden die in den Unicast-Protokollen verwendeten Konstanten, z.B. für Timeouts und Anzahl von Versuchen, wiederverwendet. Es wurde weitestgehend auf redundante Datenspeicherung verzichtet. Daten, die bereits aus den Paket-Headern der unteren Schichten abgelesen werden können, z.B. der Vorgänger beim Empfang eines `GEO-RREP`, werden im Geocast-Teil des Pakets nicht erneut aufgeführt.

5.1 Gemeinsame Funktionalitäten

5.1.1 Geolimitierter Broadcast

Wird eine Nachricht empfangen, die per geolimitiertem Broadcast (GLBC) weiterverteilt werden soll, wird die Funktion `forwardGLBC()` ausgeführt (siehe Algorithmus 5.1). Zunächst wird überprüft, ob sich der Knoten im Zielgebiet der Nachricht befindet. Falls ja, wird die TTL auf den Maximalwert gesetzt und das Paket per Broadcast versendet. Der Maximalwert der TTL ist jeweils in den Optionen der Geocast-Protokolle definiert (siehe Abschnitte 5.2.2 und 5.3.3). Falls nein, wird die TTL erniedrigt und das Paket nur dann weitergeleitet, wenn die TTL noch nicht auf 0 gesunken ist.

Algorithmus 5.1 Geolimitierter Broadcast

```
procedure FORWARDGLBC(packet)
    if ISMYLOCATION(packet.geoArea) then
        packet.TTL ← MaximumTTL
        BROADCAST(packet)
    else
        packet.TTL ← packet.TTL - 1
        if packet.TTL > 0 then
            BROADCAST(packet)
        end if
    end if
end procedure
```

5.1.2 Handhabung von Lokationsinformationen

Ortsbestimmung

Um die Lokation eines Knotens festzustellen, gibt es mehrere Möglichkeiten. Zu Testzwecken kann die Lokation von Hand gesetzt werden. Für Simulationen steht mit der Klasse `geocast::DBLocationHandler` ein Interface zum automatischen Feststellen der aktuellen Lokation im Simulationsgebiet zur Verfügung.

In der Funktion `geocast::floor2lut()` ist eine Lookup-Table fest codiert, die es ermöglicht, zu einer Knotenposition die aktuelle Lokation zu erfragen. Diese Funktion muss für die jeweils gewünschte Simulationsumgebung angepasst und ns2 entsprechend neu kompiliert werden. Die Methode `geocast::DBLocationHandler::updateLocation()` aktualisiert mit Hilfe dieser Funktion die aktuelle Lokation.

Eine Erweiterung dieser Funktionalität bietet die Klasse `geocast::InstantUpdateDBLocationHandler`. Dort ist die Methode `getLocation()` so überschrieben, dass bei jedem Zugriff zunächst die gespeicherte Lokation mittels `updateLocation()` aktualisiert wird.

Für den Fall, dass Aufräumarbeiten notwendig sind, wenn sich die Lokation ändert, bietet die Überklasse `geocast::LocationHandler` einen Callback-Mechanismus an. Die Callback-Funktion wird dabei aufgerufen, sobald eine Änderung der Lokation festgestellt wurde, d.h. gegebenenfalls auch während eines Vergleiches von Lokationen.

Um eine atomare Verarbeitung von Lokationsänderungen zu garantieren, muss daher bei Verwendung von Callbacks das Update der Lokation zwingend in einem eigenen Event erfolgen. Dies kann erreicht werden, indem die Klasse `geocast::DBLocationHandler` verwendet und mittels eines Timers periodisch `updateLocation()` aufgerufen wird. Bei Verwendung von `geocast::InstantUpdateDBLocationHandler` obliegt es dem Nutzer zu prüfen, ob die Atomizitätseigenschaft durch die Funktion seiner Callback-Funktion verletzt wird.

Prüfung auf eigene Lokation

Zur Überprüfung, ob sich ein Knoten an einer in einem Geocast-Paket adressierten Lokation befindet, dient die Funktion `isMyLocation()`, abgebildet in Algorithmus 5.2. Ein Knoten befindet sich an der angegebenen Lokation, falls die Paketadresse komponentenweise ein Anfangsstück des eigenen Präfix ist.

Bei der Prüfung wird zwischen drei Fällen unterschieden:

- Sind beide Präfixe identisch, befindet sich der Knoten im Zielgebiet.
- Ist der Präfix des Knotens kürzer als die im Paket angegebene Adresse, kann er sich nicht im Zielgebiet aufhalten. Dies gilt, da sich Knoten stets an Basis-Adressen aufhalten, d.h. zur Lokation des Knotens gibt es keine untergeordneten Lokationen. Daher kann auch die Knotenadresse kein Anfangsstück einer anderen symbolischen Adresse sein.
- Ist die im Paket angegebene Adresse kürzer als der Präfix des Knotens, befindet sich der Knoten nur dann im Zielgebiet, wenn die vollständige Paketadresse ein Anfangsstück des Knotenpräfix ist.

Im letzten Fall muss darauf geachtet werden, dass die letzte Komponente der Paketadresse einer vollständigen Adresskomponente im Knotenpräfix entspricht. Andernfalls können Nachrichten, deren letzte Komponente nur ein Anfangsstück der korrespondierenden Adresskomponente des Knotenpräfix ist, fälschlicherweise für an den Knoten addressiert gehalten werden. Ein Beispiel mit der Paketadresse `/earth/uni` und dem Knotenpräfix `/earth/universität/raum1` verdeutlicht dies. Die Paketadresse ist im Sinne von Geocast kein Anfangsstück des Knotenpräfix, da sie sich bereits in der zweiten Komponente davon unterscheidet. Wird nicht auf Gleichheit einzelner Adresskomponenten geachtet, würde sich der Knoten als Adressat des Pakets betrachten.

Algorithmus 5.2 Präfixprüfung der Lokation

```
function isMyLocation(packetLocation)
    if packetLocation == myLocation then
        Return TRUE
    else if length(packetLocation) > length(myLocation) then
        Return FALSE
    else                                // length(packetLocation) < length(myLocation)
        prefixLength ← length(packetLocation)
        prefix ← substring(myLocation, 0 .. (prefixLength - 1))
        Return packetLocation == prefix and charAt(myLocation, prefixLength) == „/“
    end if
end function
```

Aufgrund dessen, dass Lokationen in der Simulation niemals mit einem „/“ enden, kann diese zusätzliche Prüfung sehr einfach erfolgen: Das erste Zeichen nach dem gemeinsamen Anfangsstück muss ein „/“ sein.

Die Implementierung des Prüf-Algorithmus ist gemeinsam mit der Liste der Lokationen in der Klasse `geocast::LocationHandler` zu finden.

5.1.3 TCL-Schnittstelle

Die TCL-Schnittstelle beider Implementierungen wurde um mehrere Befehle erweitert. Befehle, die in beiden Varianten enthalten sind und identisch funktionieren, werden hier beschrieben. Varianten-spezifische Befehle werden in den Abschnitten zur jeweiligen Variante erläutert.

`set-location` Parameter: Geocast-Lokation. Setzt die Lokation für einen Knoten. Nur für Testzwecke zu verwenden.

`send-geocast` Parameter: Geocast-Lokation. Startet den Sendevorgang und gegebenenfalls die Routensuche für eine Geocast-Nachricht an das angegebene Zielgebiet.

5.1.4 Logging

Für eine vereinfachte Auswertung werden relevante Ereignisse als gesonderte Logzeilen in den Trace-File von ns2 geschrieben. Im Folgenden werden die geocast-spezifischen Logzeilen erläutert. Das Format der Zeilen ist für alle Implementierungen identisch, jedoch treten nicht in jeder Variante alle möglichen Kombinationen der Parameter auf.

`Loc <Timestamp> _<Knoten-ID>_ <neue Lokation>` Gibt an, dass sich die Lokation des Knotens zum angegebenen Zeitpunkt zur neuen Lokation geändert hat.

`Geo <Timestamp> _<Knoten-ID>_ <Ziellokation> <GSSN>` Zum angegebenen Zeitpunkt wurde auf dem Knoten eine Sendeanforderung gestartet. Eine Geocast-Nachricht mit der gegebenen Geocast-ID (GSSN) an das Zielgebiet wurde generiert.

`NoRoute <Timestamp> <neue Lokation> <Sender-ID> <GSSN> <Router-Status> <Routentyp>` Ein Sender musste eine von ihm generierte Nachricht verwerfen, da keine Route ins Zielgebiet existierte. Der Router-Status ist bei reaktivem Geocast stets `RTR`. Bei hierarchischem Geocast haben Designated Router den Zustand `RTR`; andere Knoten haben den Zustand `NoRTR`. Der Routentyp gibt an, ob es sich um einen Unicast-Routingfehler (`UNI`) oder einen Geocast-Routingfehler (`GEO`) handelt. Bei reaktivem Geocast tritt nur der Zustand `GEO` auf.

`RouteDel <Timestamp> <neue Lokation> <Sender-ID> <GSSN> <Router-Status> <Routentyp>` Ein Knoten auf der Route ins Zielgebiet konnte die durch Sender-ID und GSSN identifizierte Nachricht nicht weiterleiten, da ihm eine Route fehlt. Die Felder Router-Status und Routentyp erhalten die Belegung, wie in der Beschreibung von `NoRoute` angegeben.

`Deliver <Timestamp> _<Knoten-ID>_ <Ziellokation> <GSSN> <Sender-ID> (hops: <Hopcount>)` Gibt an, dass eine Nachricht, identifiziert durch Sender-ID und GSSN, einen Knoten im Zielgebiet nach einer gewissen Zahl von Hops erreicht hat und ausgeliefert wurde.

5.2 Reaktiver Geocast auf Basis von AODV

Die Implementierung für AODV-basierten Geocast setzt auf der in ns2 integrierten AODV-Implementierung auf. Die Methoden zur Verarbeitung von Nachrichten des Geocast-Routingprotokolls wurden direkt in den AODV-Routingagenten integriert. Abbildung 5.2 bietet einen Überblick über die weiteren Komponenten, die für das Geocast-Routing hinzugefügt wurden.

5.2.1 Duplikaterkennung

Zur Duplikaterkennung kann bei `GEO-RREQ`-Nachrichten auf die Mechanismen von Unicast-AODV zurückgegriffen werden. Bei anderen Geocast-Nachrichten werden neu eingeführte Geocast-Sendersequenznummern (GSSN) verwendet. Jeder Knoten führt eine Liste von

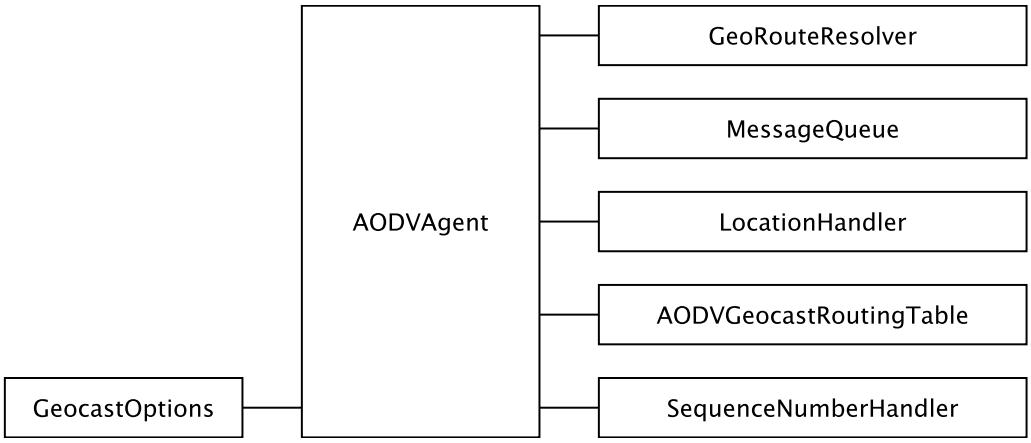


Abbildung 5.1: Überblick über die Komponenten eines Knotens bei reaktivem Geocast

Algorithmus 5.3 Erkennung von Duplikaten beim Empfang

```
function isDuplicate(packet)
    gssnTableEntry ← GSSN-Table.getTimestamp(Sender, GSSN)
    now ← currentTime
    if gssnTableEntry ≠ NULL then
        if now > gssnTableEntry.expire then
            gssnTableEntry.expire ← now + MAX_PACKET_LIFETIME
            return false
        else
            return true
        end if
    else
        GSSN-Table.insert(Sender, GSSN, now + MAX_PACKET_LIFETIME)
        return false
    end if
end function
```

Sendersequenznummern, die GSSN-Table. Jeder Eintrag dieser Tabelle besteht aus einer Knotenadresse, einer Sequenznummer und einem Zeitstempel. Der Zeitstempel gibt an, wann der Eintrag erstellt wurde. Einträge werden nach einer gewissen Zeit, der (angenommenen) maximalen Paketlebensdauer `MAX_PACKET_LIFETIME`, verworfen. Beim Empfang eines Paketes wird zur Duplikatprüfung die Funktion `isDuplicate()`, abgebildet in Algorithmus 5.3, aufgerufen. `isDuplicate()` prüft, ob die GSSN des empfangenen Paketes bereits bekannt ist.

Der Algorithmus zur Duplikaterkennung ist gemeinsam mit der Verwaltung der Geocast-Sendersequenznummer in der Klasse `geocast::SequenceNumberHandler` zu finden.

5.2.2 Parameter

Dieser Abschnitt beschreibt die Parameter, die beim Aufbau der Simulation variiert werden können. Die zugehörigen Variablen finden sich in der Klasse `geocast::aodv::GeocastOptions`. Im TCL-Code können die Parameter mit dem Befehl `Agent/AODV set <Parameter> <Wert>` gesetzt werden.

glbc_ttl_max_ Integer-Wert. Maximale TTL für geolimitierten Broadcast. Standardwert: 3

rreq_max_retries_ Integer-Wert. Gibt an, wie oft ein `GEO-RREQ` ausgesendet werden soll, bis gemeldet wird, dass keine Route gefunden werden kann. Standardwert: 3

recovery_ttl_ Integer-Wert. Radius für das begrenzte Fluten eines `GEO-RREQ` um eine abgerissene Route wiederherzustellen. Standardwert: 3

max_send_jitter_ Double-Wert. Obergrenze für die Verzögerung beim Weiterleiten von Nachrichten. Standardwert: 0,1 Sekunden

rreq_delay_ Double-Wert. Zeit, zwischen dem Versenden von zwei `GEO-RREQ`-Paketen für eine Lokation. Standardwert: 3 Sekunden

5.2.3 Konstanten

Im Folgenden werden die Konstanten aufgeführt, die für AODV-Geocast verwendet werden. Es ist jeweils angegeben, ob es sich um eine Konstante von Unicast-AODV oder eine für AODV-Geocast neu eingeführte Konstante handelt.

`AODV_GEO_ADDR_MAX` Neu eingeführt. Gibt die maximale Länge einer Geocast-Adresse, d.h. den in einem Paket reservierten Platz, an. Wert: 512 Zeichen

`AODVTYPE_GEO_RREQ` Neu eingeführt. Gibt die Pakettypnummer des Geocast-Route-Request-Pakets an. Wert: 0x20

`AODVTYPE_GEO_RREP` Neu eingeführt. Gibt die Pakettypnummer des Geocast-Route-Reply-Pakets an. Wert: 0x21

`AODVTYPE_GEO_RDEL` Neu eingeführt. Gibt die Pakettypnummer des Geocast-Route-Delete-Pakets an. Wert: 0x22

`AODVTYPE_GEO_DATA` Neu eingeführt. Gibt die Pakettypnummer des Geocast-Datenpakets an. Wert: 0x24

`MAX_PACKET_LIFETIME` Neu eingeführt. Gibt die (angenommene) maximale Lebensdauer eines Pakets an. Wert: 120 Sekunden

`ACTIVE_ROUTE_TIMEOUT` Aus Unicast-AODV. Gibt an, wie lange eine Route gültig ist. Wert: 10 Sekunden

`NETWORK_DIAMETER` Aus Unicast-AODV. Gibt den maximalen Durchmesser des Netzwerkes an. Wert: 30

5.2.4 Paketformat

Alle Paketdefinitionen sind in der Datei `aodv/aodv_packet.h` zu finden.

Die AODV-Paketdefinitionen wurden um vier Pakettypen für Geocast erweitert: `GEO-RREQ`, `GEO-RREP`, `GEO-RDEL` und `GEO-DATA`. Um Probleme beim Parsen von Geocast-Adressen auf einfache Weise zu vermeiden, wird in jedem Paket ein Puffer fester Länge von ca. 512 Byte Länge für die Repräsentation einer Geocast-Adresse als nullterminierter String vorgehalten.

Die AODV-Implementierung in ns2 legt fest, dass alle Pakete mit einer Typnummer und einem sich daran anschließenden reservierten Bereich beginnen. Erst darauf folgen die paketspezifischen Felder.

Die in den Paketdefinitionen verwendeten Konstanten sind in Abschnitt 5.2.3 beschrieben.

GEO-RREQ

Ein Geocast-Route-Request signalisiert, dass eine Route zu einem im Paket angegebenen Ziel gesucht wird. Da ein GEO-RREQ auch dazu dient eine Unicast-Reverse-Route zum Absender aufzubauen, sind die Felder des Pakets größtenteils denen des Unicast-RREQ entnommen. Lediglich Destination Address und Destination Sequence Number wurden durch die Geocast-Adresse ohne Sequenznummer ersetzt. GEO-RREQ verwendet den Pakettyp AODVTYPE_GEO_RREQ.

Listing 5.1 Typdefinition eines GEO-RREQ-Paketes

```
struct hdr_aodv_geo_rreq {
    u_int8_t   re_type;                          // Packet Type
    u_int8_t   reserved[2];                      // Reserved
    u_int32_t  rq_bcast_id;                      // Broadcast ID - used for dups
    u_int8_t   rq_hop_count;                     // Hop Count
    char       destinationArea[AODV_GEO_ADDR_MAX]; // Destination
    nsaddr_t   rq_src;                           // Source IP Address
    u_int32_t  rq_src_seqno;                     // Source Sequence Number
    u_int16_t  ttl;                              // TTL for ERS
};
```

GEO-RREP

Ein Geocast-Route-Reply signalisiert, dass eine Route zu einem angefragten Zielgebiet gefunden werden konnte. Ein GEO-RREP enthält lediglich Informationen über das Zielgebiet und einen Hopcount, der die Entfernung ins Zielgebiet angibt. GEO-RREP verwendet den Pakettyp AODVTYPE_GEO_RREP.

Listing 5.2 Typdefinition eines GEO-RREP-Paketes

```
struct hdr_aodv_geo_rrep {
    u_int8_t   re_type;                          // Type
    u_int8_t   reserved[2];                      // Reserved
    char       destinationArea[AODV_GEO_ADDR_MAX]; // Destination area
    u_int16_t  hopcount;                         // Hops to destination area
};
```

GEO-DATA

Ein Geocast-Datenpaket simuliert die eigentliche Datenübertragung. Da keine realen Daten übertragen werden, besteht ein GEO-DATA-Paket nur aus der Zieladresse, der GeocastID zur Duplikaterkennung und einem TTL-Wert, der für den geolimitierten Broadcast verwendet wird. GEO-DATA verwendet den Pakettyp AODVTYPE_GEO_DATA.

Listing 5.3 Typdefinition eines GEO-DATA-Paketes

```
struct hdr_aodv_geo_data {
    u_int8_t   re_type;                          // Type
    u_int8_t   reserved[2];                      // Reserved
    char       destinationArea[AODV_GEO_ADDR_MAX]; // Destination area
    u_int16_t  ttl;                              // TTL for GLBC
    u_int32_t  geocastId;                        // Sender-Unique ID to suppress duplicates
};
```

GEO-RDEL

Ein Geocast-Route-Delete signalisiert, dass ein Paket nicht ins Zielgebiet geleitet werden konnte, da die Route abgerissen ist. Ein GEO-RDEL enthält die Adresse des Zielgebietes sowie eine GeocastID – die ID des Datenpaketes, durch das der Routen-Abriss entdeckt wurde. GEO-RDEL verwendet den Pakettyp AODVTYPE_GEO_RDEL.

Listing 5.4 Typdefinition eines GEO-RDEL-Paketes

```
struct hdr_aodv_geo_rdel {
    u_int8_t   re_type;                          // Type
    u_int8_t   reserved[2];                      // Reserved
    char       destinationArea[AODV_GEO_ADDR_MAX]; // Destination area
    u_int32_t  geocastId;                  // ID of the data packet that caused the GEO_RDEL
};
```

5.2.5 TCL-Schnittstelle

Zusätzlich zu den in Abschnitt 5.1.3 aufgeführten Kommandos wird folgendes Kommando über die TCL-Schnittstelle bereitgestellt:

del-location Parameter: Geocast-Lokation. Löscht einen Routingtabelleneintrag zu einem Zielgebiet auf dem Knoten. Nur für Testzwecke zu verwenden.

5.2.6 Utility-Klassen

GeoRouteResolver

Der GeoRouteResolver realisiert das Expanding Ring Search für die Routensuche.

Ein Timer ruft periodisch eine Routine auf, die zunächst den Suchradius anpasst und anschließend den AODV-Agent dazu veranlasst, einen GEO-RREQ auszusenden. Wurde eine festgelegte Maximalzahl von GEO-RREQs ohne Antwort versendet, wird stattdessen signalisiert, dass keine Route gefunden werden konnte. Für jede gesuchte Route muss ein gesonderter GeoRouteResolver erstellt werden.

`GeoRouteResolver`-Objekte enthalten Informationen darüber, wie oft ein `GEO-RREQ` wiederholt werden soll, mit welcher TTL begonnen werden soll, und in welchen Schritten die TTL erhöht werden muss. Der Initialwert für die Anzahl der Wiederholungen wird der Variable **rreq_max_retries_** entnommen. Die Anfangs-TTL sowie die Schrittweite werden so errechnet, dass mit dem letzten `GEO-RREQ` das gesamte Netzwerk geflutet wird. Die Netzwerkgröße wird der Konstante `NETWORK_DIAMETER` entnommen.

Weiterhin ist eine Liste von Knoten enthalten, von denen Pakete für das anzufragende Zielgebiet vorliegen. Kann keine Route gefunden werden, wird ein `GEO-RDEL` an alle diese Knoten gesendet.

Wenn durch ein eintreffendes `GEO-RREP` dem AODV-Agent signalisiert wird, dass eine Route gefunden wurde, wird der `GeoRouteResolver` für dieses Ziel von der Behandlungsroutine für das `GEO-RREP`-Paket deaktiviert.

Algorithmus 5.1 Handler des GeoRouteResolver

```
    function EXPIRE
 2:     if retries > RREQ_RETRIES then
            for all requestors do
 4:             AODV-AGENT.SENDGEODEL(geoArea, requestor)
            end for
 6:         Remove all messages for destionation from queue
        else
 8:         entry ← GEOROUTINGTABLE.LOOKUP(geoArea)
            if entry == NULL then
10:             TTL ← TTL + TTL-Increment
                AODV-AGENT.SENDGEORREQ(geoArea, TTL)
12:             retries ← retries + 1
                RESCHEDULE
14:         end if
        end if
16: end function
```

MessageQueue

Die `MessageQueue` speichert zu jeder Lokation die Nachrichten, die zum Versand bereitstehen, für die aber noch keine Route gefunden wurde. Da in der Simulation keine Inhalte übertragen werden, werden ledglich die identifizierenden Paare aus Absenderadresse und Sequenznummer abgelegt.

5.2.7 Geocast-Routingtabelle

Die Geocast-Routingtabelle wurde als Hashtabelle implementiert. Als Schlüssel dient der String der Ziellokation, als Wert werden der Next-Hop, die Entfernung zum Zielgebiet in Hops und ein Verfallszeitpunkt gespeichert. Routen verfallen, wenn länger als die in `ACTIVE_ROUTE_TIMEOUT` angegebene Zeitspanne kein `GEO-RREP` für diese Route eingeht.

Als Typ der Routingtabelle wurde `std::tr1::unordered_map` aus der Boost-Library verwendet.

5.2.8 Abfrage der Lokation

Es wird der `geocast::InstantUpdateDBLocationHandler` verwendet, da kein Callback bei Änderungen der Lokation notwendig ist.

5.2.9 Logging

Die Funktion `format_aodv()` aus `trace/cmu-trace.cc` wurde modifiziert, um den Paketinhalt von Geocast-Paketen im Trace-File auszugeben. Der Paketinhalt wird dabei wie folgt formatiert:

`GEO_RREQ bcid:<Broadcast-ID> hopcnt:<Hopcount> area:<Ziel-Präfix> src:<Quell-adresse> srcSeq:<Sendersequenznummer der Quelle> ttl:<TTL>` Geocast-Route-Request. Broadcast-ID und Sendersequenznummer sind aus Unicast-AODV entnommen und werden für die Reverse-Route verwendet. Die Quelladresse ist die Adresse des Knotens, der die Route aufbauen möchte.

`GEO_RREP area:<Ziel-Präfix> hopcnt:<Hopcount>` Geocast-Route-Reply. Der Ziel-Präfix ist der Präfix, zu dem die Route aufgebaut wird.

`GEO_DATA area:<Ziel-Präfix> geosrc:<Quelladresse> ttl:<TTL> geocastID:<GSSN>` Geocast-Datenpaket. Die Quelladresse ist die Adresse des Knotens, der die Geocast-Nachricht generiert hat.

`GEO_RDEL area:<Ziel-Präfix> geosrc:<Quelladresse>` Geocast-Route-Delete. Der Ziel-Präfix ist der Präfix des Gebietes, zu dem die Route abgerissen ist. Die Quelladresse gibt die Quelle des Geocasts an, also den Knoten, der über den Routenabbruch benachrichtigt werden soll.

Weiterhin wurden für eine vereinfachte Analyse Methoden hinzugefügt mit der Änderungen der Lokation, Geocast-Sendeanfragen, die Auslieferung von Geocast-Nachrichten sowie fehlende Routen in den Trace-File geschrieben werden. Das Format dieser Logzeilen wurde bereits in Abschnitt 5.1.4 beschrieben.

5.2.10 Befehls- und Paketverarbeitung

`send-geocast`

Wird der Befehl `send-geocast` erteilt, wird zunächst eine `GEO-DATA`-Nachricht generiert und mit einer GSSN versehen. Anschließend überprüft der sendende Knoten, ob er sich bereits im Zielgebiet befindet. Falls ja, wird die Nachricht über den GLBC-Mechanismus weitergegeben. Falls nein, wird die Nachricht in der `MessageQueue` abgelegt und ein `GeoRouteResolver` für das Ziel erstellt, der seinerseits wie in Abschnitt 5.2.6 beschrieben `GEO-RREQ`s versendet.

`GEO-RREQ`

Nach dem Empfang eines `GEO-RREQ` durchläuft das Paket zunächst Routinen von Unicast-AODV, um die Reverse-Route aufzubauen. Dabei wird auch festgestellt, ob es sich um ein Duplikat handelt und das Paket bereits verarbeitet wurde. Anschließend wird geprüft, ob die Nachricht per Unicast oder per Broadcast empfangen wurde.

Wurde das Paket per Unicast empfangen, heißt dies, dass der Sender bereits über eine Route informiert ist und der `GEO-RREQ` nur zum Aufbau der Reverse-Route weitergeleitet wurde. In diesem Fall wird weiterhin überprüft, ob sich der Knoten bereits im Zielgebiet befindet. Falls ja, wird das Paket verworfen. Falls nein, wird das Paket entlang der bestehenden Route weiter ins Zielgebiet geleitet.

Ein Empfang per Broadcast bedeutet, dass noch nach einer Route gesucht wird. In diesem Fall wird untersucht, ob eine Route ins Zielgebiet bekannt ist oder ob der Knoten sich bereits im Zielgebiet befindet. Falls eine dieser Bedingungen erfüllt ist, wird ein `GEO-RREP` generiert und zum suchenden Knoten zurückgeschickt. Der `GEO-RREQ` wird, wie im obigen Fall, ins Zielgebiet weitergeleitet. Falls der Knoten keine Informationen über das Zielgebiet hat, wird der `GEO-RREQ` unter Berücksichtigung der TTL per Broadcast weitergesendet.

GEO-RREP

Beim Empfang eines GEO-RREP wird zunächst in der Routingtabelle gesucht, ob bereits eine bessere Route, d.h. eine mit weniger Hops, ins Zielgebiet existiert. Falls ja, wird das Paket verworfen. Falls nein, wird der Routingtabelleneintrag erneuert. Wenn der Knoten nicht der Adressat des GEO-RREP ist, leitet er das Paket zum eigentlichen Adressaten weiter. Stehen in der Nachrichten-Queue Nachrichten für das im GEO-RREP angegebene Zielgebiet bereit, werden diese im Anschluss als GEO-DATA-Pakete per Geocast versendet. Ein möglicherweise laufender GeoRouteResolver für das angegebene Zielgebiet wird gestoppt.

GEO-DATA

Wird ein GEO-DATA Paket per Unicast empfangen, befindet es sich noch auf dem Weg ins Zielgebiet.

Befindet sich der empfangende Knoten im Zielgebiet, setzt er die TTL auf den Maximalwert und versendet das Paket per Broadcast an alle Nachbarn, d.h. schwenkt um auf Verteilung per GLBC.

Befindet sich der Knoten nicht im Zielgebiet, sucht er in seiner Geocast-Routingtabelle nach dem Next-Hop zum fraglichen Zielgebiet. Falls kein solcher Eintrag existiert, wird das Paket zunächst in die lokale Nachrichten-Queue gestellt und es wird analog zur Verarbeitung von send-geocast ein GeoRouteResolver gestartet. Hierbei wird jedoch kein ERS durchgeführt, sondern es wird nur einmalig mittels einem stark limitierten Fluten nach einer Route ins Zielgebiet gesucht. Der Radius der Flutung wird durch die Variable **recovery_ttl_** bestimmt. Kann keine Route gefunden werden, wird ein GEO-RDEL zum Absender des GEO-DATA gesendet und alle Nachrichten für das fragliche Zielgebiet werden aus der Queue gelöscht.

Wird eine Route in der Routingtabelle gefunden, wird das Paket zum Next-Hop dieser Route per Unicast weitergeleitet.

Der Empfang von GEO-DATA-Paketen per Broadcast bedeutet, dass das Paket bereits per GLBC im Zielgebiet verteilt wird. Ein so empfangenes Paket wird zunächst mittels des SequenceNumberHandlers auf Duplizität geprüft. Die Weiterleitung erfolgt mittels der Funktion forwardGLBC() aus Abschnitt 5.1.1.

GEO-RDEL

Wird ein GEO-RDEL empfangen, wird die Route zum im GEO-RDEL angegebenen Zielgebiet gelöscht. Falls der Knoten nicht der finale Empfänger des GEO-RDELs ist, wird das Paket per Unicast zum angegebenen Empfänger, der Quelle der abzubauenden Route, weitergeleitet. Nachrichten, die möglicherweise noch für das Zielgebiet gepuffert sind, werden nicht entfernt. Dies übernimmt der Timeout-Mechanisus des GeoRouteResolvers.

Negatives Ergebnis eines `GeoRouteResolvers`

Kann ein `GeoRouteResolver` keine Route aufbauen, d.h. wurde er nicht vor Erreichen der maximalen Anzahl von Versuchen durch ein empfangenes `GEO-RREP`-Paket gestoppt, werden, wie bereits in Abschnitt 5.2.6 beschrieben, alle Nachrichten verworfen, die für die fragliche Lokation bereitlagen. Weiterhin wird allen anderen Knoten, die eine Route zum Zielgebiet angefragt haben, mit einem `GEO-RDEL` per Unicast signalisiert, dass keine Route gefunden werden kann.

5.3 Hierarchischer Geocast auf Basis von OLSR

Die Implementierung für OLSR-basierten Geocast setzt auf UM-OLSR [Ros09] als OLSR-Implementierung für ns2 auf.

5.3.1 Architektur des Routers

Abbildung 5.3.1 gibt einen Überblick über die Komponenten eines Knotens, der für hierarchischen Geocast auf Basis von OLSR konfiguriert ist. Die Routing- und Befehlsverarbeitungsfunktionalität ist in zwei Klassen aufgeteilt: `geocast::olsr::GeocastRouter` und `geocast::olsr::DesignatedRouter`. Der `GeocastRouter` stellt einen Basis-Prozess dar, der auf jedem Knoten läuft, unabhängig davon, ob der Knoten als DR agiert oder nicht. Der `DesignatedRouter` übernimmt die Funktionalitäten eines DR und wird nur bei Bedarf aktiviert.

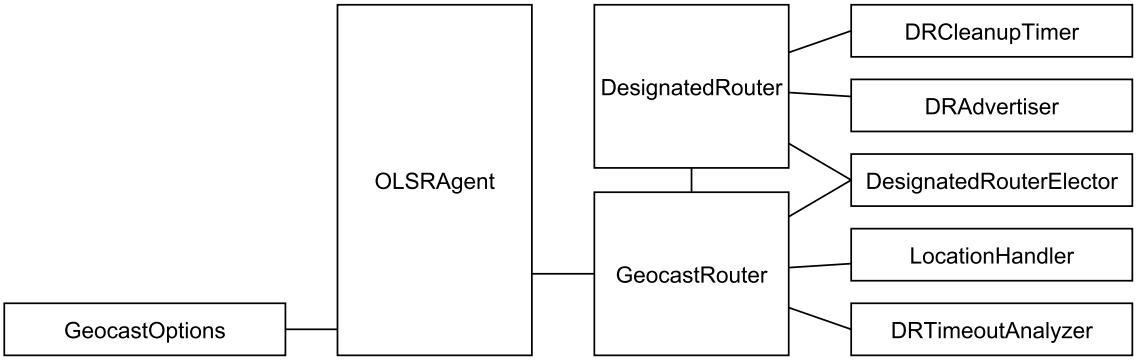


Abbildung 5.2: Überblick über die Komponenten eines Knotens bei hierarchischem Geocast

GeocastRouter

Der `GeocastRouter` wird vom OLSR-Code aufgerufen um Geocast-Nachrichten zu verarbeiten und gegebenenfalls eine Forwarding-Entscheidung zu treffen. Es obliegt dem `GeocastRouter`, eine Nachricht von Unicast-Verbreitung auf GLBC-Verbreitung umzustellen. Weiterhin ist er dafür zuständig die Lokationsinformationen zu halten und einen DR für die aktuelle Lokation zu bestimmen. Die für die `WRAPPER`-Nachricht notwendige Funktionalität zum Ein- und Auspacken von Paketen ist ebenfalls in dieser Klasse implementiert.

DesignatedRouter

Der `DesignatedRouter` verwaltet die Routinginformationen des Geocast-Baumes, d.h. die Adressen und Präfixe anderer DRs. Ebenso fällt die Verwaltung über- oder untergeordneter Proxies in seinen Aufgabenbereich.

5.3.2 Duplikaterkennung

UM-OLSR bietet bereits einen Mechanismus zur Erkennung von duplizierten Nachrichten. Daher muss für hierarchischen Geocast keine gesonderte Duplikaterkennung implementiert werden.

5.3.3 Parameter

Dieser Abschnitt beschreibt die Parameter, die beim Aufbau der Simulation variiert werden können. Die zugehörigen Variablen finden sich in der Klasse `geocast::olsr::GeocastOptions`. Im TCL-Code können die Parameter mit dem Befehl `Agent/OLSR set <Parameter> <Wert>` gesetzt werden.

glbc_ttl_max_ Integer-Wert. Maximale TTL für geolimitierten Broadcast. Standardwert: 3

draInterval_ Double-Wert. Zeit, zwischen dem Versenden von `DRA`s. Standardwert: 2 Sekunden

maxDraMissing_ Integer-Wert. Anzahl der `DRA`s, die verloren gehen dürfen, bevor eine Route als veraltet angesehen wird. Standardwert: 3

drElectionTime_ Double-Wert. Dauer eines Wahlvorganges. Standardwert: 10 Sekunden

drsSendCount_ Integer-Wert. Anzahl von `DRS`-Nachrichten, die während eines Wahlvorganges versendet werden. Standardwert: 4

lambda_ Double-Wert. Parameter zur Einstellung der Gewichtung der Historie für den Shortcut-Algorithmus (siehe [DR08]). Standardwert: 0,5

minRefCount_ Integer-Wert. Minimale Anzahl der Referenzen, die nötig sind, damit eine Lokation als Ziel eines Shortcuts in Frage kommen kann (siehe [DR08]). Standardwert: 10 Sekunden

locationUpdateInterval_ Double-Wert. Zeit zwischen zwei Überprüfungen der aktuellen Lokation. Standardwert: 1 Sekunde

5.3.4 Konstanten

Im folgenden werden die Konstanten aufgeführt, die für OLSR-Geocast verwendet werden.

`OLSR_WARMSET_SIZE` Größe des Warmset für die Ermittlung von Shortcuts (siehe [DR08]). Wert: 1000

`OLSR_HOTSET_SIZE` Größe des Hotset für das Halten von aktiven Shortcuts (siehe [DR08]). Wert: 50

5.3.5 Paketformat

Alle Paketdefinitionen sind in der Datei `olsr/OLSR_pkt.h` zu finden. OLSR-Paketheader bestehen aus einem gemeinsamen Teil, der für alle Pakettypen identisch ist, und einem typspezifischen Teil. Über ein Union-Konstrukt wird der typspezifische Header in einem OLSR-Paket abgelegt. Im Folgenden werden zunächst die relevanten gemeinsamen Headerfelder beschrieben. Darauf folgt eine Beschreibung der neu hinzugefügten Pakettypen und ihrer typspezifischen Header.

Alle OLSR-Pakete enthalten zunächst die Typnummer, die angibt, um welche Art von OLSR-Paket es sich handelt. Weiterhin sind ein TTL-Feld, ein Sequenznummern-Feld und ein Absender-Feld enthalten. Das TTL-Feld kann für GLBC genutzt werden. Über das Sequenznummern-Feld stellt UM-OLSR eine Duplikaterkennung bereit, so dass keine weitere Behandlung duplizierter Nachrichten im geocast-spezifischen Code notwendig ist.

Das Absender-Feld gibt an, welcher Knoten eine Nachricht ursprünglich erstellt hatte, da die Absenderadresse im IP-Header bei jedem Weiterleiten geändert wird. Dies resultiert daraus, dass UM-OLSR bis zu vier Broadcast-Nachrichten in einem Paket zusammenfasst, auch wenn diese von unterschiedlichen Absendern stammen.

Die OLSR-Paketdefinitionen wurden um fünf Pakettypen für Geocast erweitert: Designated Router Advertisement (`DRA`), Designated Router Solicitation (`DRS`), Geocast Data (`GEO-DATA`), verschachtelte Nachrichten (`WRAPPER`) und Designated Router Shortcut Solicitation (`DRSS`). Um Probleme beim Parsen von Geocast-Adressen zu vermeiden, wird in jedem Paket ein Puffer fester Länge von 512 Byte für die Repräsentation einer Geocast-Adresse als nullterminierter String vorgehalten.

Designated Router Advertisement (DRA)

Ein DRA signalisiert, dass der Absender als DR für die im Paket angegebene Lokation arbeitet. Mit dem infrastructureFlag identifiziert sich ein Absender als Mesh-Router bzw. Mesh-Client. Das dra_type-Feld gibt an, ob es sich um ein normales (REGULAR) DRA im DR-Baum handelt, oder ob das DRA zu einer Shortcut-Verbindung gehört. In letzterem Fall wird angegeben, ob es sich beim Absender um denjenigen Knoten handelt, der den Shortcut aufgebaut hat (SHORTCUTSOURCE) oder denjenigen, der Ziel des Shortcuts ist (SHORTCUTTARGET).

Listing 5.5 Typdefinition einer DRA-Nachricht

```
typedef struct OLSR_dra {
        char destinationArea[GEO_ADDR_MAX];
        bool infrastructureFlag;
        enum dra_type_t {REGULAR, SHORTCUTSOURCE, SHORTCUTTARGET};
        dra_type_t dra_type;
} OLSR_dra;
```

Designated Router Solicitation (DRS)

Mit einem DRS wird nach einem Router für eine Lokation gesucht. Gleichzeitig dienen DRS-Nachrichten der Wahl eines neuen DR, falls für die fragliche Lokation kein DR existieren sollte. Das infrastructureFlag gibt an, ob es sich bei dem Absender um einen Mesh-Router oder einen Mesh-Client handelt. Der requestorPrefix enthält die tatsächliche Lokation des Absenders, destinationArea enthält die Lokation, für die ein DR gesucht wird.

DRS-Nachrichten sind ausschließlich zum Versand per GLBC vorgesehen.

Listing 5.6 Typdefinition einer DRS-Nachricht

```
typedef struct OLSR_drs {
        char destinationArea[GEO_ADDR_MAX];
        bool infrastructureFlag;
        char requestorPrefix[GEO_ADDR_MAX];
} OLSR_drs;
```

Geocast-Daten (GEO-DATA)

Ein GEO-DATA-Paket symbolisiert eine Geocast-Nachricht an ein Zielgebiet. Da in der Simulationsumgebung keine Daten versendet werden, enthält das GEO-DATA-Paket keinen Platz für eine Nutzlast.

Listing 5.7 Typdefinition einer GEO-DATA-Nachricht

```
typedef struct OLSR_geo {
       char destinationArea[GEO_ADDR_MAX];
} OLSR_geo;
```

Geocast-Wrapper (WRAPPER)

Die WRAPPER-Nachricht ist ein der Implementierung geschuldeter Sonderfall. Um Designated Proxies zu unterstützen, muss eine Nachricht vom übergeordneten DR per Unicast an alle Proxies versendet werden. Dabei passiert die Nachricht möglicherweise auch Knoten, die dieselbe Nachricht über geolimitierten Broadcast erhalten haben. Da beide Nachrichten dieselbe Sequenznummer und denselben Absender haben, wird eine der beiden Nachrichten als Duplikat erkannt und unterdrückt. Die WRAPPER-Nachricht enthält die Daten einer OLSR-Nachricht (OLSR_msg), verwendet aber eine neue Sequenznummer um die Duplikaterkennung zu umgehen. Eine WRAPPER-Nachricht kann selbst wieder als Teil einer OLSR-Nachricht auftreten. Es ist jedoch nicht möglich, WRAPPER-Nachrichten nochmals in andere WRAPPER-Nachrichten zu verpacken.

WRAPPER-Nachrichten sind ausschließlich zum Versand per Unicast vorgesehen.

Listing 5.8 Typdefinition einer WRAPPER-Nachricht

```
typedef struct OLSR_wrapper {
       u_int8_t      msg_type_;    ///< Message type.
       u_int8_t      vtime_;       ///< Validity time.
       u_int16_t     msg_size_;    ///< Message size (in bytes).
       nsaddr_t      orig_addr_;   ///< Main address of the node which generated this
           message.
       u_int8_t      ttl_;         ///< Time to live (in hops).
       u_int8_t      hop_count_;   ///< Number of hops which the message has taken.
       u_int16_t     msg_seq_num_; ///< Message sequence number.

       union {
              OLSR_dra      dra_;
              OLSR_drs      drs_;
              OLSR_geo      geo_;
       } msg_body_;                ///< Message body.
} OLSR_wrapper;
```

Designated Router Shortcut Solicitation (DRSS)

Die DRSS-Nachricht wird per Geocast in das angegebene Zielgebiet geroutet und fordert den dortigen DR auf, sich beim absendenden DR per Unicast zu melden, damit ein Shortcut aufgebaut werden kann.

Listing 5.9 Typdefinition einer DRSS-Nachricht

```
typedef struct OLSR_ss {
     char destinationArea[GEO_ADDR_MAX];
} OLSR_ss;
```

5.3.6 TCL-Schnittstelle

Zusätzlich zu den in Abschnitt 5.1.3 aufgeführten Kommandos werden folgende Kommandos über die TCL-Schnittstelle bereitgestellt:

set-infrastructure Parameter: Keine. Konfiguriert einen Knoten als Mesh-Router.

set-client Parameter: Keine. Konfiguriert einen Knoten als Mesh-Client.

dump-geoinfo Parameter: Keine. Schreibt den internen Zustand des Geocast-Routers auf dem Knoten in den Trace-File. Die erstellte Ausgabe ist nur für Debugging-Zwecke geeignet.

5.3.7 Utility-Klassen

DRTimeoutAnalyzer

Der DRTimeoutAnalyzer ist ein Timer im GeocastRouter. Seine Aufgabe ist zu überprüfen, ob eine gewisse Zeit lang kein Designated Router Advertisement vom übergeordneten DR mehr empfangen wurde, d.h. ob der übergeordnete DR ausgefallen ist.

DesignatedRouterElector

Der DesignatedRouterElector implementiert den Wahlalgorithmus, um einen DR zu wählen, oder um zu bestimmen, welcher Knoten der übergeordnete DR einer Gruppe von Proxies wird. Diese Klasse enthält auch einen Timer, der das Versenden von DRS-Nachrichten auslöst.

Der implementierte Wahlalgorithmus ist sehr einfach gehalten, erfüllt jedoch die Konsistenzeigenschaft: Stehen Mesh-Router zur Wahl, wird aus diesen ein Knoten ausgewählt. Nur wenn keine Mesh-Router zur Wahl stehen, wird auf Mesh-Clients zurückgegriffen. Aus der jeweiligen Knotenmenge wird derjenige Knoten mit der größten Adresse ausgewählt.

DRCleanupTimer

Der DRCleanupTimer entfernt periodisch abgelaufene Einträge aus den Routingtabellen des DesignatedRouters.

DRAdvertiser

Der `DRAdvertiser` versendet periodisch Designated Router Advertisements, um einen Knoten als DR bekannt zu machen und Verbindungen zu ihm bekannten anderen DRs aktuell zu halten.

5.3.8 Abfrage der Lokation

Es wird der `DBLocationHandler` verwendet. Ein Timer sorgt dafür, dass die aktuelle Lokation einmal pro Sekunde erneuert wird. Die Verarbeitung von Lokationsänderungen in einem eigenen Event ist notwendig, um atomare Ausführung zu garantieren. Bei einer Änderung der Lokation muss unter anderem ein eventuell laufender `DesignatedRouter` abgeschaltet werden. Wenn dies während der Verarbeitung einer Nachricht passieren würde, wäre ein fehlerhaftes Routing oder der vollständige Verlust der Nachricht die Folge.

5.3.9 Logging

Die Funktion `CMUTrace::format_olsr_geocast()` gibt den Inhalt von Geocast-Paketen in den Trace-File aus. `WRAPPER`-Nachrichten werden dabei entpackt, damit auch ihr Inhalt in den Trace geschrieben werden kann. Die Funktion wird aus `CMUTrace::format_olsr()` heraus aufgerufen. Der Nachrichteninhalt wird dabei wie Folgt formatiert:

`[DRA <Quelladresse> <Hopcount> <OLSR-Sequenznummer> <Zielgebiet> <TTL>]` Designated Router Advertisement mit Router-Adresse, Hopcount, der von UM-OLSR vergebenen Sequenznummer, dem Präfix des DRs und der TTL.

`[DRS <Quelladresse> <Hopcount> <OLSR-Sequenznummer> <Präfix des anfragenden Knotens> <Zielgebiet> <TTL>]` Designated Router Solicitation mit Knotenadresse, Hopcount, der von UM-OLSR vergebenen Sequenznummer, dem Präfix des Knotens, dem Präfix, für den ein DR gesucht wird, und der TTL.

`[GEO <Quelladresse> <Hopcount> <OLSR-Sequenznummer> <Zielgebiet> <TTL>]` Geocast-Nachricht mit Senderadresse, Hopcount, der von UM-OLSR vergebenen Sequenznummer, dem Präfix des Zielgebietes und der TTL.

`[WRAPPER <Quelladresse> <Hopcount> <OLSR-Sequenznummer> <TTL> <Eingepackte Nachricht>]` Wrapper-Nachricht mit Adresse des einpackenden Knotens, Hopcount, der von UM-OLSR vergebenen Sequenznummer, der TTL und der eingepackten Nachricht. Die entpackte Nachricht wird mit `format_olsr_geocast()` formatiert.

Weiterhin wurden für eine vereinfachte Analyse Methoden hinzugefügt, mit der Änderungen der Lokation, Geocast-Sendeanfragen, die Auslieferung von Geocast-Nachrichten sowie fehlende Routen in den Trace-File geschrieben werden. Das Format dieser Logzeilen wurde bereits in Abschnitt 5.1.4 beschrieben.

5.3.10 Befehls- und Paketverarbeitung

Befehle der TCL-Schnittstelle

Alle Befehle, die über die TCL-Schnittstelle eingehen, werden vom `GeocastRouter` verarbeitet.

set-location Dieser Befehl wird in `GeocastRouter::setLocation(string)` verarbeitet. Wird mittels `set-location` einem Knoten mitgeteilt, dass er sich an einer neuen Lokation befindet, so wird zunächst diese Lokation gespeichert und, falls sich die Lokation tatsächlich geändert hat, der `DesignatedRouter` deaktiviert. Anschließend wird der `DesignatedRouterElector` für eine Wahl eines DR für die neue Lokation konfiguriert.

send-geocast Dieser Befehl wird in `GeocastRouter::sendGeocast(string)` verarbeitet. Falls der sendende Knoten sich bereits im Zielgebiet der Nachricht befindet, wird die Nachricht direkt als GLBC verteilt. Andernfalls wird die Nachricht per Unicast an den DR der eigenen Lokation gesendet, der die Nachricht zum Ziel weiterleitet. Ist der Knoten selbst der DR, wird die Nachricht direkt dem nächsten bekannten DR in Richtung des Zielgebietes überstellt.

Nachrichtenempfang

Alle Nachrichten werden zunächst vom `GeocastRouter` untersucht und gegebenenfalls an den `DesignatedRouter` weitergegeben.

`DRA` Ein Designated Router Advertisement (`DRA`) wird zunächst in `GeocastRouter::process_dra(OLSR_msg, nsaddr_t)` untersucht.

Falls das `DRA`-Paket per GLBC verteilt wird, muss zwischen der Behandlung auf einem DR und auf einem regulären Knoten unterschieden werden. Auf einem regulären Knoten wird geprüft, ob das Zielgebiet des Pakets mit dem eigenen Präfix übereinstimmt. Falls ja, wird

der Absender des Pakets als DR für den eigenen Präfix notiert. Unabhängig davon wird das Paket per GLBC weiterverteilt.

Auf einem DR wird das Paket an `DesignatedRouter::process_dra_glbc(OLSR_dra, nsaddr_t)` weitergegeben und, wie in Abschnitt „`DRA` per GLBC auf einem DR“ beschrieben, verarbeitet.

Handelt es sich um ein Unicast-`DRA`, das nicht an den Knoten adressiert ist, wird es per Unicast zum Adressaten weitergeleitet. Da Unicast-`DRA`s nur an DRs gesendet werden, muss andernfalls der Knoten einen aktiven `DesignatedRouter` besitzen. Das Paket wird an `DesignatedRouter::process_dra_uc(OLSR_dra, nsaddr_t)` zur weiteren in Abschnitt „`DRA` per Unicast auf einem DR“ beschriebenen Verarbeitung übergeben.

`DRA` per GLBC auf einem DR In `DesignatedRouter::process_dra_glbc(OLSR_dra, nsaddr_t)` verarbeitet ein DR per GLBC empfangene `DRA`s. Falls ein `DRA` für einen eigenen Präfix empfangen wird, wurde eine Partitionierung aufgehoben. Es muss mit einer Instanz des `DesignatedRouterElectors` bestimmt werden, welcher der beiden Knoten in Zukunft als DR aktiv bleiben soll. Der verlierende Knoten löscht seinen DR-Status für den fraglichen Präfix und alle übergeordneten Präfixe. Der gewinnende Knoten sendet sofort seinerseits ein `DRA` per GLBC aus, um allen Knoten den aktuell korrekten DR mitzuteilen.

`DRA` per Unicast auf einem DR In `DesignatedRouter::process_dra_uc(OLSR_dra, nsaddr_t)` wird bei der Verarbeitung von per Unicast empfangenen `DRA`s zwischen mehreren Fällen unterschieden.

Ist der Knoten DR für den in der Nachricht angegebenen Präfix so existieren mehrere DRs für einen Präfix und es muss eine Proxy-Hierarchie aufgebaut werden. Über einen `DesignatedRouterElector` wird entschieden, welcher der Knoten weiter DR für den Präfix bleiben soll und welcher Knoten zum DP wird. Der DP trägt den DR als neuen übergeordneten Knoten ein. Der DR notiert sich den DP in seiner Proxy-Routingtabelle.

Wenn der Knoten kein DR für den in der Nachricht angegebenen Präfix ist, handelt es sich um eine Routenaktualisierung eines anderen Knotens. Kennt der Knoten die Route noch nicht, so wird diese in die Routingtabelle eingetragen. Kennt der Knoten die Route bereits mit dem Absender des `DRA` als Ziel, wird der Timeout der Route zurückgesetzt. Wenn im empfangenen `DRA` das `dra_type`-Feld auf den Typ `SHORTCUTSOURCE` gesetzt wurde, wird neben der Aktualisierung des Routingtabelleneintrages mit einem `DRA` mit dem Typ `SHORTCUTTARGET` geantwortet.

Kennt der Knoten die Route bereits mit einem anderen Ziel, existieren wiederum mehrere DRs für einen Präfix. In diesem Fall muss aus den beiden als Routenziel bekannten DRs eine Proxy-Hierarchie gebildet werden. Mittels eines `DesignatedRouterElectors` wird der zukünftige DR bestimmt und als Ziel der Route eingetragen. Um die untergeordneten Knoten zum Aufbau einer Proxy-Hierarchie zu veranlassen, sendet der Knoten `DRA`-Nachrichten an den DR und den Proxy, die den jeweils anderen Knoten als Absender enthalten.

Wann immer ein Routingtabelleneintrag geändert wurde, muss der Eintrag für den übergeordneten Knoten mit dem neuen oder erneuerten Eintrag verglichen werden, da ein neuer übergeordneter DR einen alten DR verdrängen kann.

DRS Designated Router Solicitation Nachrichten (`DRS`) werden ausschließlich per GLBC versendet. Wird eine solche Nachricht in `GeocastRouter::process_drs(OLSR_msg)` empfangen, wird geprüft ob die Nachricht an die Basislokation des Knotens gerichtet ist und für diese Lokation eine Wahl durchgeführt wird. Falls ja, wird der Absender als Kandidat der Wahl hinzugefügt.

Weiterhin wird die Nachricht an `DesignatedRouter::process_dra(OLSR_dra, nsaddr_t)` übergeben. Dort wird geprüft, ob der Knoten bereits DR für die fragliche Lokation ist. Falls ja, wird die Weiterleitung der Nachricht unterdrückt und ein `DRA` per Unicast als Antwort an den suchenden Knoten gesendet.

Falls nein, wird der Absender als Kandidat einer möglichen Wahl im `DesignatedRouter` hinzugefügt. Zuletzt wird die Nachricht per GLBC weitergeleitet, wenn die Weiterleitung nicht wie beschrieben unterdrückt wurde.

GEO-DATA Geocast-Nutznachrichten werden in `GeocastRouter::process_geo(OLSR_msg, nsaddr_t)` verarbeitet.

Wurde die Nachricht per GLBC empfangen, wird sie auch, wie in Abschnitt 5.1.1 beschrieben, per GLBC weitergeleitet. Falls der Knoten ein DR ist, leitet er die Nachricht in einer `WRAPPER`-Nachricht an alle ihm direkt untergeordneten Proxies weiter.

Unicast-Nachrichten an das Gebiet des Knotens werden in jedem Fall direkt in GLBC-Nachrichten konvertiert. Andernfalls wird die Nachricht per Unicast zum in der Nachricht angegebenen Empfänger weitergegeben. Ist der Knoten selbst der Empfänger, aber kein DR, wird die Nachricht zum nächstgelegenen DR weitergeleitet.

Ist der Knoten Empfänger und auch selbst DR, durchsucht er seine Routingtabelle nach dem Eintrag mit dem längsten gemeinsamen Präfix der adressierten Lokation und leitet die Nachricht dorthin weiter. Falls kein solcher Knoten gefunden werden kann, leitet der Knoten die Nachricht an seinen übergeordneten Knoten weiter.

Falls der Knoten ein Proxy-DR ist, wird die Nachricht per Unicast zum übergeordneten DR weitergegeben. Die einzige Ausnahme sind Nachrichten, die vom übergeordneten DR empfangen wurden. Ist der Knoten selbst ein übergeordneter DR, gibt er die Nachricht an alle ihm untergeordneten Proxies weiter.

WRAPPER Falls die Wrapper-Nachricht an den verarbeitenden Knoten adressiert ist, wird die enthaltene Nachricht entpackt, decodiert und wie eine auf normalem Wege empfangene Nachricht weiter verarbeitet. Ist der verarbeitende Knoten nicht der Empfänger, wird die Nachricht per Unicast weitergeleitet.

DRSS Designated Router Shortcut Solicitation Nachrichten (DRSS) werden in Designated-Router::process_ss(OLSR_msg) verarbeitet. Ist der empfangende Knoten der DR für die im DRSS-Paket angegebene Lokation, sendet er ein DRA mit dra_type SHORTCUTTARGET an den Absender des DRSS. Andernfalls leitet er das DRSS-Paket weiter durch den DR-Baum zum DR der angegebenen Lokation.

Timer

DRAdvertiser Der DRAdvertiser veranlasst einen aktiven DesignatedRouter dazu, DRA-Nachrichten zu versenden. DRA-Nachrichten werden in vier Kategorien versendet: Proxies, Vater und Kinder, Shortcuts und eigene Lokation. DRAs an die eigene Lokation enthalten die Basislokation des Knotens und werden per GLBC verteilt.

DRAs an untergeordnete Proxies enthalten jeweils die Lokation, für die das Proxying stattfindet und werden per Unicast versendet. Diese DRAs enthalten den dra_type REGULAR.

DRAs für Vater und Kinder werden ebenfalls immer per Unicast versendet, enthalten aber, abhängig von der Lokation des Ziels, unterschiedliche Lokationen. Wenn es sich um eine untergeordnete Lokation, d.h. einen Kindknoten im DR-Baum, handelt, wird die darüberliegende Lokation verwendet. Andernfalls, d.h. für übergeordnete Knoten und für alle Knoten, die als Shortcut bekannt sind, wird die höchste Lokation verwendet, für die der DR zuständig ist. DRAs an Vater und Kinder enthalten den dra_type REGULAR.

Über Shortcuts werden DRAs nur dann versandt, wenn der sendende Knoten auch den Aufbau des Shortcuts gewünscht hat. Diese DRAs enthalten den dra_type SHORTCUTSOURCE.

DRTimeoutAnalyzer Der DRTimeoutAnalyzer teilt einem GeocastRouter auf einem nicht-DR-Knoten mit, dass der Timeout für die periodischen Meldungen des DR abgelaufen ist, d.h. dass ein neuer DR gewählt werden muss. Es wird ein DesignatedRouterElector gestartet, um einen neuen DR für den Präfix des Knotens zu bestimmen.

DesignatedRouterElector Ein Ablauf des Timers des DesignatedRouterElector löst das Senden von DRS-Nachrichten für die gesuchte Lokation aus. Nach mehrmaligem Ablauf des Timers wird statt einem erneuten Senden das Wahlergebnis bestimmt. Über eine Callback-Funktion wird das Objekt, das die Wahl gestartet hat, über das Ergebnis informiert.

Handelte es sich um einen DesignatedRouter und wurde dieser selbst zum DR bestimmt, trägt er den fraglichen Präfix in eine Liste ein und startet eine neue Wahl für den darüberliegenden Präfix. Wurde ein anderer Knoten als DR ausgewählt, wird dieser als übergeordneter Knoten in der Routingtabelle vermerkt.

Handelte es sich um einen GeocastRouter und wurde dieser selbst zum DR bestimmt, wird der DesignatedRouter aktiviert und als DR für die Basislokation konfiguriert. Der DesignatedRouter verhält sich anschließend wie im letzten Absatz beschrieben, d.h. er

startet eine Wahl für den nächsten übergeordneten DR. Wurde der Knoten nicht als DR gewählt, wird die Adresse des gewählten DR vermerkt und die Wahl beendet.

DRCleanupTimer Ein Ablauf des `DRCleanupTimer` veranlasst einen `DesignatedRouter` dazu, die Routing- und Proxy-Tabelle sowie den Link zum übergeordneten DR auf abgelaufene Routingtabelleneinträge zu untersuchen und solche Einträge zu entfernen. Dies dient nur dazu, den Speicherverbrauch zu senken, da bei jedem Zugriff auf die Routing- und Proxy-Tabelle ohnehin geprüft wird, ob ein gefundener Eintrag noch aktuell ist.

6 Evaluierung

Die in Kapitel 5 beschriebenen Implementierungen der entwickelten Routing-Protokolle wurden zur Evaluation in einer simulierten Umgebung eingesetzt. Da eine Simulation von der Größe eines Metropolitan Area Networks auch modernste Rechencluster vor eine enorme Aufgabe stellt, wurde für die Simulation ein kleineres Netz mit einem Indoor-Szenario gewählt. Die folgenden Abschnitte beschreiben die simulierte Umgebung sowie den technischen Aufbau der Simulation. Abschnitt 6.3 erörtert, in welchen Punkten deutliche Abweichungen von einem realen System zu erwarten sind, bevor schließlich in Abschnitt 6.4 die Simulationsergebnisse diskutiert werden.

6.1 Evaluierungsumgebung

Als Umgebung zur Evaluierung wurde ein Modell des 2. Stocks des Informatik-Gebäudes der Universität Stuttgart verwendet. Das fragliche Stockwerk bietet breite Hauptgänge mit Aufenthaltsbereichen sowie unterschiedlich große Büroräume. Ein 2D-Modell des Simulationsgebiets mit raumgenauen Lokationen wurde bereits im Rahmen des Nexus-Projekts [Pro08] erstellt und musste lediglich in ein passendes Eingabeformat konvertiert werden.

6.1.1 Netz-Infrastruktur

Mesh-Router werden entsprechend dem realen Aufbau des Test-Netzwerkes in diesem Bereich in der Simulation platziert. Um den Effekt einer verbesserten Infrastruktur zu untersuchen, wurden ebenfalls Simulationsläufe durchgeführt, bei denen sich an jeder Basislokation ein Mesh-Router befindet. Die Platzierung der Mesh-Router erfolgte gemeinsam mit der Berechnung der Mobilität wie in Abschnitt 6.1.2 beschrieben.

6.1.2 Mobilität und Knotenplatzierung

Für die Erstellung eines Mobilitätsszenarios für ns2 wird das „GraphWalk“-Modul von CanuMobiSim [Ste05] verwendet. Dafür muss das 2D-Modell in eine reine Graph-Darstellung überführt werden. Jeder Raum des 2D-Modells entspricht einem Knoten, wobei der Knoten die Koordinaten des Raummittelpunktes erhält. Die Kanten des Graphen sind die im Modell angegebenen Durchgänge. Weiterhin wird die Position aller statischen Knoten, d.h. der

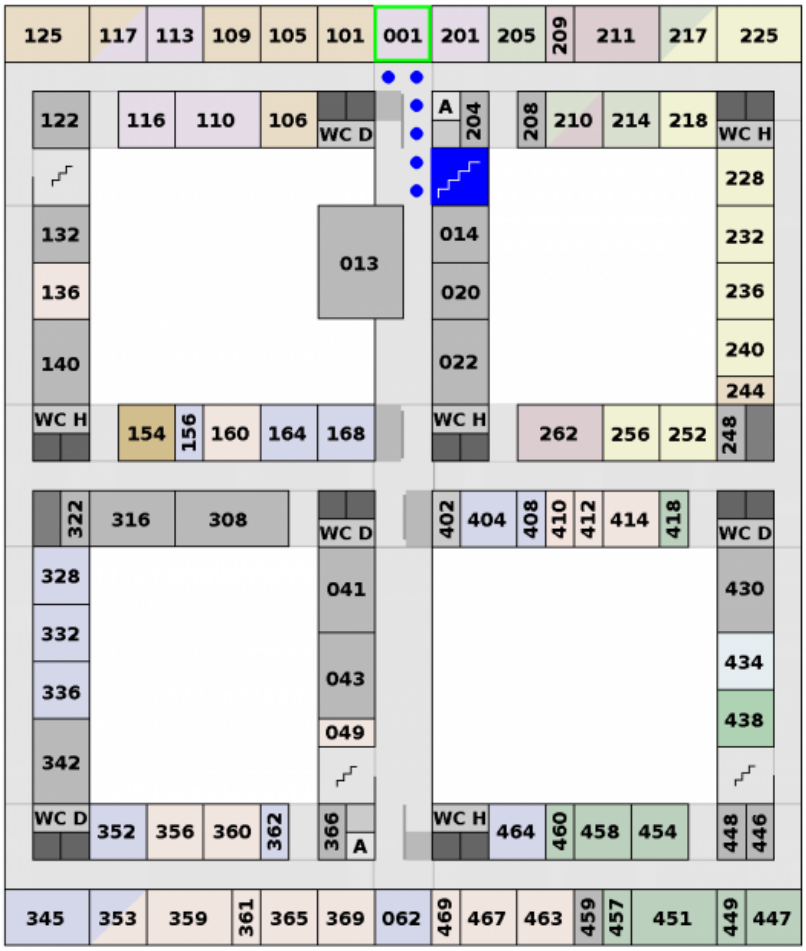


Abbildung 6.1: Raumplan des Informatik-Gebäude der Universität Stuttgart, Vaihingen, 2. Stock. Quelle: Info-Terminals[1]

Mesh-Router, eingetragen. Zu den Mesh-Routern ist jeweils bekannt, in welchem Raum sie sich befinden. Über das 2D-Modell kann wiederum der Mittelpunkt des jeweiligen Raumes bestimmt werden, der dann in die CanuMobiSim-Konfiguration eingetragen wird.

In verschiedenen Simulationsläufen werden unterschiedliche Fortbewegungsgeschwindigkeiten getestet: Fußgänger bewegen sich zwischen $0,5\frac{\mathrm{m}}{\mathrm{s}}$ und $3\frac{\mathrm{m}}{\mathrm{s}}$, Fahrradfahrer zwischen $2\frac{\mathrm{m}}{\mathrm{s}}$ und $10\frac{\mathrm{m}}{\mathrm{s}}$. Diese Werte werden zusammen mit Minimal- und Maximaldauer von Bewegungspausen ebenfalls in der Konfiguration des „GraphWalk"-Moduls abgelegt. Der Wert für eine initiale Bewegungspause ist auf 0 festgelegt. Die Dauer späterer Bewegungspausen liegt zwischen einer und etwa acht Minuten.

[1] `http://infoterminals.informatik.uni-stuttgart.de:8090/FIFI-1.2/terminal/index.jsp?page=events&terminal=1` (nur zu erreichen aus dem internen Netz der Informatik)

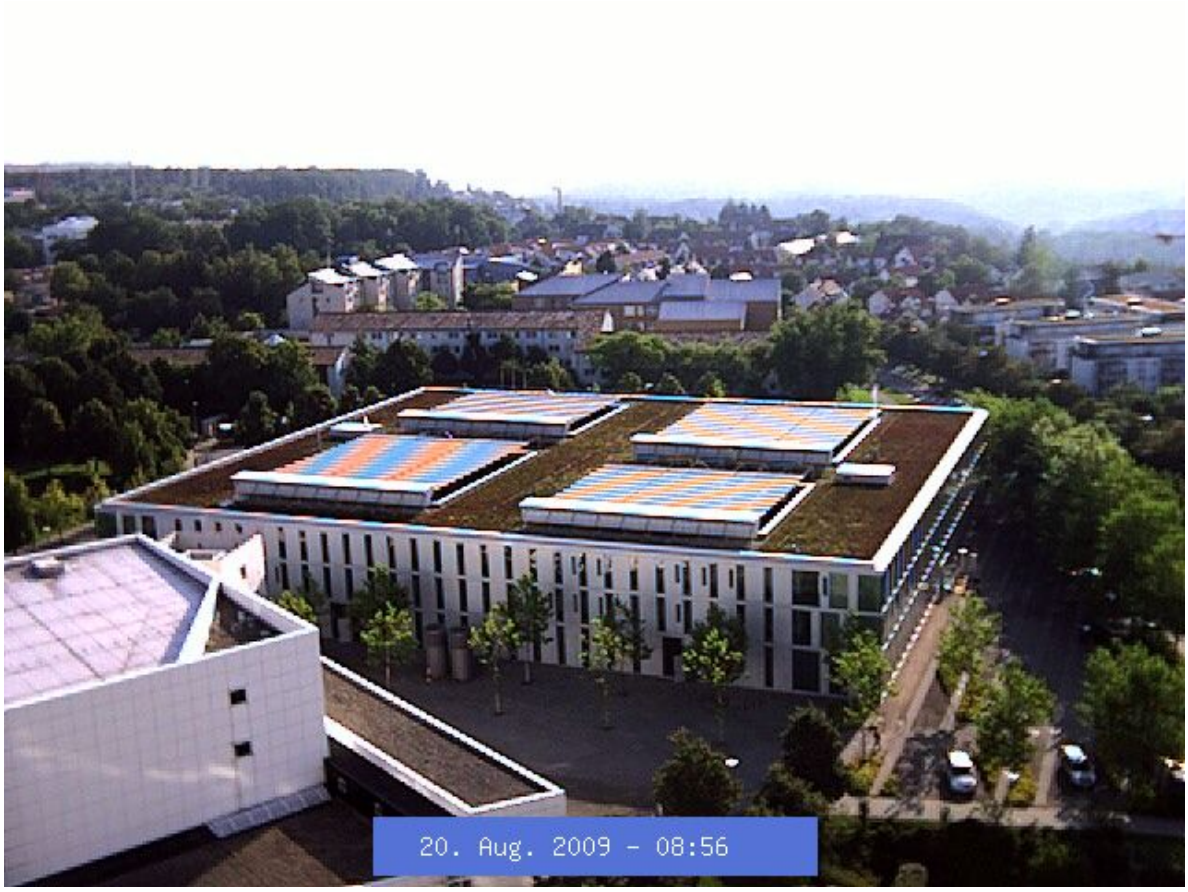


Abbildung 6.2: Informatik-Gebäude der Universität Stuttgart, Vaihingen. Deutlich zu erkennen sind die im Bild mit Sonnensegeln abgedeckten Innenhöfe. Quelle: Webcam[1]

Aus den beschriebenen Eingaben erzeugt CanuMobiSim eine Szenarioberschreibung für ns2. Diese kann in ein Simulationsskript eingebunden werden und enthält die Mobilitätsinformation für den gesamten Simulationsverlauf.

Betrachtet man ein mit CanuMobiSim derart generiertes Mobilitätsmodell, stellt man fest, dass es Phasen gibt, in denen sich alle Knoten bewegen, während im Rest der Zeit alle Knoten stillstehen. Erst nach längerer (Simulations-)Zeit driften die Bewegungs- und Wartephasen einzelner Knoten auseinander. Um diesen Effekt zu kompensieren, werden Mobilitätsdaten schon ab 7200 Sekunden vor dem eigenlichen Simulationsbeginn generiert. Später werden diese „zu früh“ generierten Daten abgeschnitten, so dass nur die Mobilitätsdaten für die tatsächliche Simulationszeit verbleiben.

[1] `http://neubau.informatik.uni-stuttgart.de/gross.html`

6.1.3 Lokationsinformationen

Die Zuordnung von Koordinaten zu Lokationen wird aus dem Modell extrahiert und in eine Eingabedatei für ns2 konvertiert (siehe Abschnitt 5.1.2). Simulierte Knoten können so jederzeit zu ihrer geographischen Position ihren Präfix raumgenau bestimmen.

6.1.4 Verkehrsmodell Geocast

Wie bereits in Abschnitt 3.1.3 beschrieben wird davon ausgegangen, dass wenige Knoten als Sender vieler Nachrichten auftreten.

Sendezeitpunkte für Geocast-Nachrichten werden nach einem Poisson-Prozess einmal für das gesamte Netz errechnet. Dies ermöglicht es auf einfache Art und Weise einzustellen, ob sich Geocast-Routen gegenseitig beeinflussen können. Um die in Abschnitt 3.1.3 beschriebene Verteilung von Sendern und angefragten Lokationen zu modellieren, wird über eine Power-Law-Verteilung jedem Knoten und jeder Lokation im Netz eine Wahrscheinlichkeit zugewiesen, mit der sie als Sender beziehungsweise Ziel einer Geocast-Nachricht auftreten. Zu jedem Sendezeitpunkt werden dann der absendende Knoten und das Zielgebiet entsprechend dieser Wahrscheinlichkeiten ausgewählt.

Geocast-Nachrichten werden in drei Häufigkeitsstufen simuliert: Eine Nachricht alle 20 Sekunden, eine Nachricht pro Sekunde und drei Nachrichten pro Sekunde. Im ersten Fall soll sichergestellt werden, dass bei reaktivem Geocast keine Geocast-Routen mehr existieren, d.h. dass bei jeder Nachricht eine neue Route aufgebaut werden muss. In den beiden letzteren Fällen soll das Netz mit Geocast-Nachrichten unter Druck gesetzt werden.

Bei reaktivem Geocast startete der Nachrichtenversand für Unicast wie auch Geocast mit Simulationsbeginn. Bei hierarchischem Geocast wird dagegen dem Netz ausreichend Zeit eingeräumt, Unicast-Routen und den Geocast-Routerbaum aufzubauen. Daher starteten in diesem Fall Unicast-Verbindungen 30 Sekunden, Geocast-Sendeanfragen 90 Sekunden nach Simulationsbeginn.

6.1.5 Verkehrsmodell Unicast

Da Geocast-Anwendungen in einem Mesh-Netz sehr wahrscheinlich parallel zu Unicast-Anwendungen wie WWW oder E-Mail eingesetzt werden, soll dieses Zusammenspiel auch in der Simulation erforscht werden.

Die Startzeitpunkte von Unicast-Verbindungen werden je Knoten durch einen Poisson-Prozess ermittelt. Für jede Verbindung wird gleichverteilt ein Zielknoten und eine Dauer zwischen 3 und 20 Sekunden gewählt. Es werden TCP-Verbindungen mit einer Fenstergröße von 32 und einer Paketgröße von 512 Byte aufgebaut.

Der Abstand zweier Verbindungen je Knoten variiert in verschiedenen Simulationsläufen zwischen 30 Sekunden, einer Minute und 2 Minuten. Ebenso werden jedoch auch Simulationen völlig ohne Unicast-Traffic durchgeführt, um die ungestörte Funktionsweise der Geocast-Protokolle untersuchen zu können.

6.2 Untersuchte Größen

Um die Eignung der entwicklten Protokolle für den angedachten Einsatzzweck zu untersuchen, wurden Effektivität und Effizienz der Geocast-Nachrichtenzustellung untersucht.

Effektivität

Als wichtigster Maßstab der Effektivität dient der Anteil an Nachrichten, die ihr Zielgebiet erreicht haben. An zweiter Stelle steht die Abdeckung des Zielgebietes bei der Verteilung einer Nachricht.

Eine weitere Größe, die die Effektivität jedoch in geringerem Maße beeinflusst, ist die Verzögerung von Nachrichten. Als Verzögerung wird der zeitliche Abstand zwischen einer Sendeanforderung und dem Erreichen des ersten Knotens im Zielgebiet angesehen. Laut Systemmodell dürfen beliebige Verzögerungen bei der Nachrichtenauslieferung auftreten, jedoch wird eine zu stark verzögerte Nachricht von einem Benutzer als verloren gewertet werden.

Effizienz

Die Effizienz der Protokolle wird durch zwei Größen charaktersiert. Als wichtigste Größe gilt dabei die Fairness, d.h. die Verteilung der Last auf die Knoten. Falls sich im Netz wenige Hot-Spots ausbilden sollten, die die Hauptlast der Routingarbeit tragen müssen, ist davon auszugehen, dass das eingesetzte Routingprotokoll nicht ausreichend skaliert. Um die Last zu quantifizieren, wurde die Anzahl der Sendeereignisse auf Schicht 2 (Sicherungsschicht) verwendet.

Als weitere Größe folgt die Pfadlänge. Nachrichten mit einer übermäßig hohen Pfadlänge sind ein Indiz dafür, dass das Routingprotokoll keine optimalen Routen findet und deuten somit auf eine Netzüberlastung hin.

6.3 Ungenauigkeiten der Simulation

Simulationen bilden stets nur eine vereinfachte Version der realen Welt ab. Um eine bessere Interpretation der Simulationsergebnisse zu ermöglichen, werden im Folgenden die bekannten Ungenauigkeiten und Vereinfachungen der Simulation ausgeführt.

Die vermutlich gravierendste Ungenauigkeit ist das Modell des drahtlosen Übertragungskanals in ns2. Das eingesetzte Two Ray Ground Reflection Propagation Model modelliert weder Abschattungen durch Objekte noch Kanalfehler. Um jeden Knoten gibt es ein kreisförmiges Gebiet, innerhalb dessen eine Nachricht stets empfangen wird. Außerhalb hat eine Nachricht keine Auswirkungen.

Mit dem Shadowing Model bietet ns2 zwar auch noch ein Modell eines drahtlosen Übertragungskanals mit Abschattungen. Allerdings verwendet dieses keine Umgebungsinformationen. Das Modell kann lediglich in soweit parameteriert werden als es sich um ein Indoor- oder Outdoor-Szenario handelt, und wie dicht das Simulationsgebiet im Mittel mit Hindernissen besetzt ist. Derart vage Konfigurationsmöglichkeiten erscheinen wenig sinnvoll, um aussagekräftige Ergebnisse zu gewinnen. Daher wurde für den drahtlosen Kanal das Two Ray Ground Reflection Propagation Model verwendet. Die ausgebildete Netztopologie entspricht in etwa der, die im real installierten Mesh-Netzwerk des Instituts für parallele und verteilte Systeme der Universität Stuttgart zu beobachten ist.

Insbesondere treten auf dem Übertragungskanal keine Bitfehler auf. Nachrichten können nur aus zwei Gründen verloren gehen: Durch Kollisionen mit einer gleichzeitigen Übertragung eines anderen Knotens und durch Überlaufen einer Sendewarteschlange auf der Sicherungsschicht.

Ein Nebeneffekt des Two Ray Ground Reflection Propagation Model ist, dass der Hop-Count von Nachrichten als sehr gut funktionierende Metrik erscheint.

Abschnitt 2.3.2 beschreibt die Funktionsweise von OLSR nach RFC3626. Inzwischen wurden einige der Probleme, die beim Einsatz von OLSR in realen Systemen aufgetreten sind, gelöst. Für Simulationen sind jedoch nur RFC-konforme Implementierungen von OLSR verfügbar, so dass der Einfluss der Verbesserungen auf das Geocast-Routing nicht untersucht werden konnte.

6.4 Ergebnisse

Im Folgenden werden die Ergebnisse der Simulation präsentiert und ausgewertet. Alle Ergebnisse wurden über mehrere Simulationsläufe gemittelt.

6.4.1 Routingfehler und Nachrichtenverlust

Abbildungen 6.3 bis 6.8 zeigen die Nachrichtenverlustraten aufgeschlüsselt nach allen Simulationsparametern. Jeder Messwert wurde als Mittelwert von fünf Simulationsläufen mit den angegebenen Parametern errechnet. Die Fehlerfälle sind grob in vier Klassen einzuteilen: Geocast-Routingprobleme, Unicast-Routingprobleme, Kollisionen und verzögerte Nachrichten.

Verzögerte Nachrichten sind strenggenommen keine verlorenen Nachrichten. Wird jedoch die Latenz bei der Auslieferung zu hoch, wird ein Benutzer eine Nachricht dennoch als verloren ansehen. Die Akzeptanzgrenze wurde in diesem Graphen auf 60 Sekunden gesetzt.

Bei Untersuchungen zum Unicast-Traffic ist zu beachten, dass Verbindungen knotenweise generiert wurden. Daraus resultiert, dass z.B. in der Unicast-Stufe „low" bei 100 mobilen Knoten deutlich mehr Unicast-Verbindungen angefordert werden als bei nur 30 Knoten.

Reaktiver Geocast

Die Werte bei reaktivem Geocast sind wie folgt zu interpretieren:

Keine Geocast-Route beim Absender Der Knoten, der eine Geocast-Sendeanforderung erhalten hat, konnte keine Route ins Zielgebiet finden.

Keine Geocast-Route auf Zwischenknoten Ein weiterleitender Knoten konnte eine Nachricht nicht per Geocast-Routing weiterleiten. Der Next-Hop auf der Route ins Zielgebiet war nicht mehr verfügbar und die Bestimmung einer alternativen Route ist fehlgeschlagen.

Keine Unicast-Route Ein Knoten konnte ein Paket aufgrund einer fehlenden Unicast-Route nicht weiterleiten. Tritt bei reaktivem Geocast nicht auf, da das Routing ausschließlich auf Geocast-Routen beruht.

Kollision Ein `GEO-DATA`-Paket ist durch eine Kollision beim Versand verloren gegangen.

Paket verzögert Die Nachricht wurde erst mehr als 60 Sekunden nach der Sendeanforderung im Zielgebiet ausgeliefert.

Es fällt zunächst auf, dass kaum Routingprobleme auf Zwischenknoten auftreten, d.h. wenn eine Route einmal aufgebaut ist, ist diese sehr stabil. Der Fall der auf Zwischenknoten unterbrochenen Route berücksichtigt nur Geocast-Routingfehler, die nicht behoben werden konnten. Ebenfalls auffällig ist, dass reaktiver Geocast mit nur wenigen Infrastrukturknoten vergleichsweise viele Nachrichten verspätet ausliefert. Die wahrscheinlichste Erklärung ist, dass durch die Mobilität der Knoten Routen doch häufig abreißen und somit bei einer Auslieferung die Route unter Umständen mehrfach repariert werden muss.

Weiterhin ist zu beobachten, dass Paketverluste und Probleme bei der Routensuche oft in ähnlichem Maße wachsen. Gründe für das Nichtauffinden einer Route können Netzpartitionierungen, Mobilität von Knoten während der Routensuche und Verlust der Routing-Pakete sein. Während Netzpartitionierungen gerade bei den Simulationen mit 30 und 50 mobilen Knoten relativ häufig auftreten, ist die Mobilität ein eher vernachlässigbarer Faktor. Zum einen bewegen sich stets nur einzelne Knoten, zum anderen sollte das mehrfache Aussenden der `GEO-RREQ`-Nachrichten beim Expanding Ring Search ausreichend Redundanz bieten, um Knoten, die sich während der Routensuche wegbewegen, zu umgehen und eine Route ins Zielgebiet zu finden.

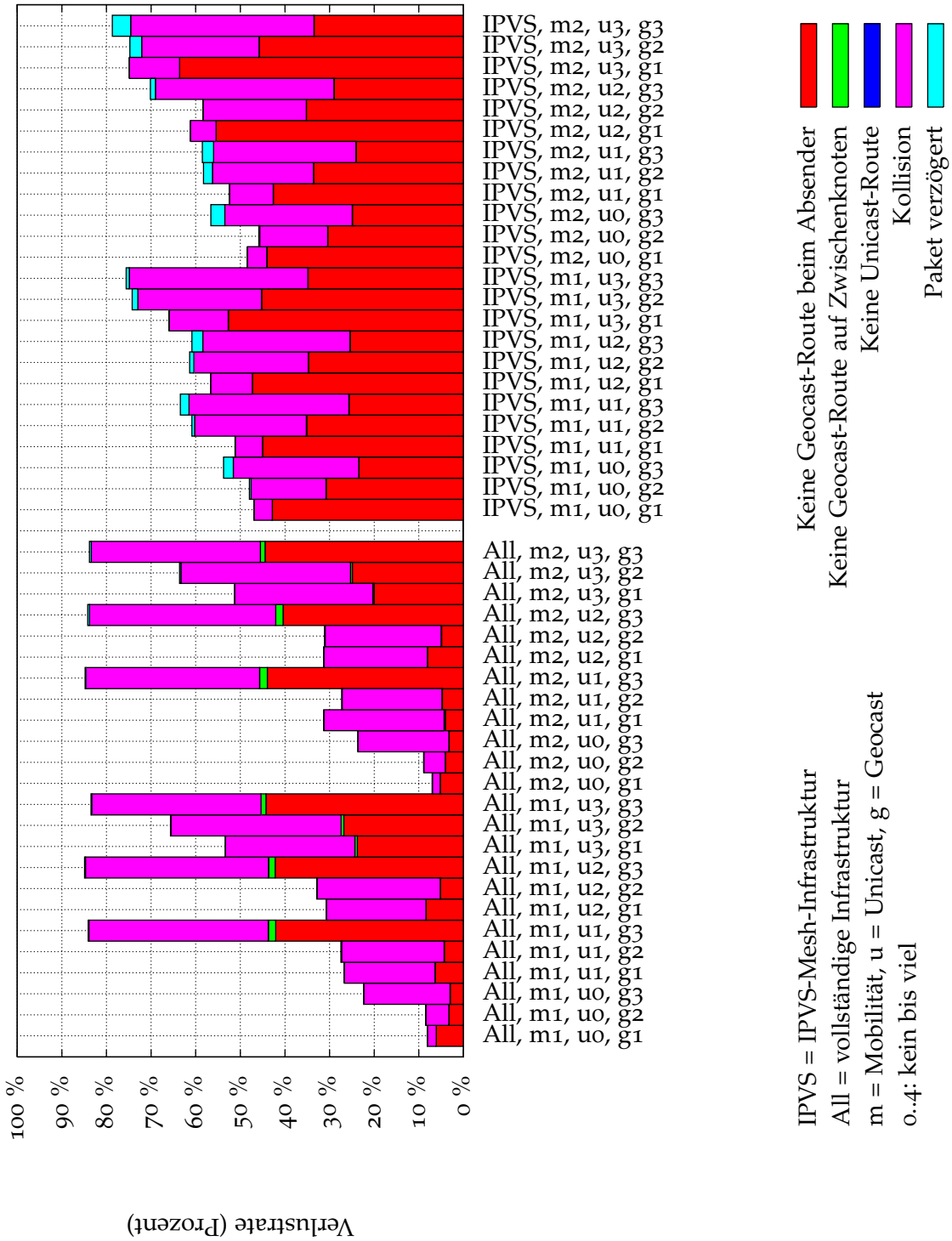


Abbildung 6.3: Nachrichtenverlustraten bei AODV, 30 Knoten

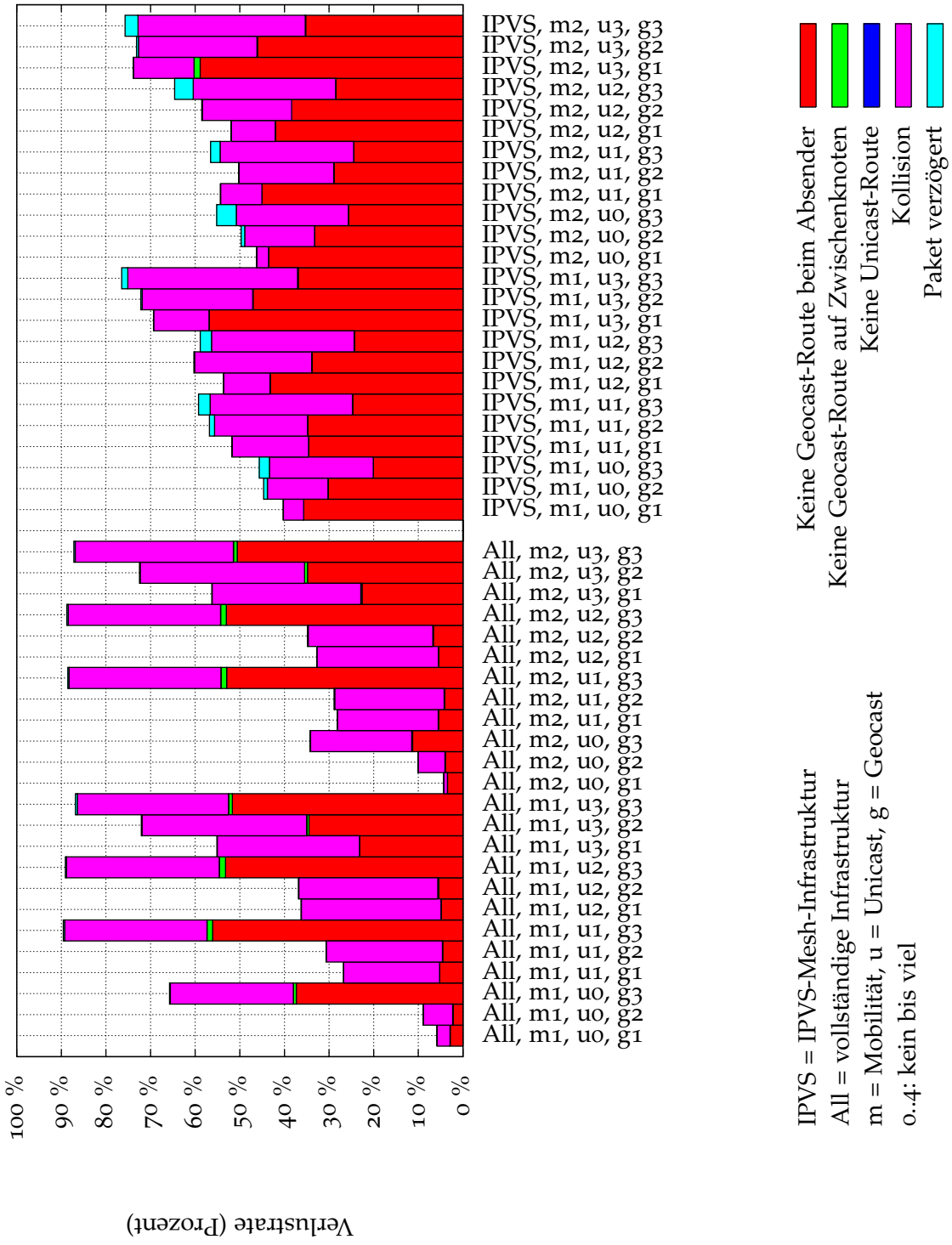


Abbildung 6.4: Nachrichtenverlustraten bei AODV, 50 Knoten

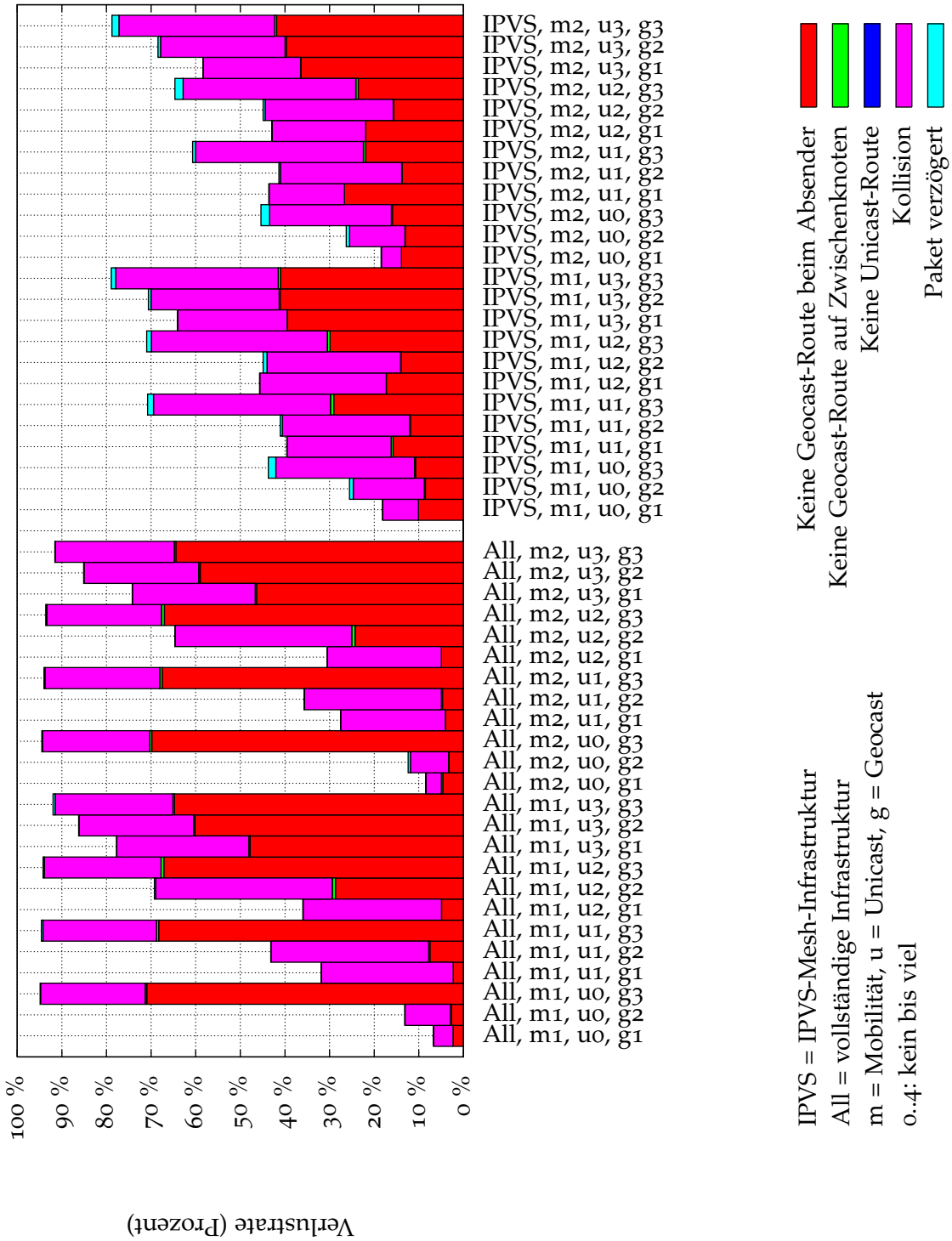


Abbildung 6.5: Nachrichtenverlustraten bei AODV, 100 Knoten

Als einziger Grund verbleibt also das Problem der Paketverluste. Da für den Aufbau einer Route zunächst ein Round-Trip – `GEO-RREQ` zum Zielgebiet und `GEO-RREP` zurück zum Sender – erforderlich ist, bevor ein `GEO-DATA`-Paket auf die Reise gehen kann, ist leicht ersichtlich dass in nicht-partitionierten Netzen Routingfehler beim Absender und der Verlust von `GEO-DATA`-Paketen durch Kollisionen in ähnlichen Größenordnungen liegen müssen.

AODV-basierter Geocast ist auf eine vollständige Infrastruktur angewiesen. Bei 30 bzw. 50 mobilen Knoten ist das Netz ohne Infrastrukturknoten zu dünn, um Routen finden zu können. Bei 100 mobilen Knoten verkehrt sich diese Eigenschaft allerdings ins Gegenteil: Wird das Netz zu dicht, werden deutlich weniger Routen gefunden. Ohne die Mesh-Router an jeder Lokation ist die Routenfindung mit 100 mobilen Knoten deutlich zuverlässiger. Der Grund hierfür dürfte sein, dass bei einem Broadcast in einem dichteren Netz die Kollisionswahrscheinlichkeit höher ist, d.h. dass mehr `GEO-RREQ`-Pakete verloren gehen.

Weiterhin kann man erkennen, dass AODV-basierter Geocast sehr empfindlich gegenüber der Gesamtlast des Netzes ist. Wenn mehr Unicast-Verbindungen angefordert werden oder mehr Geocast-Routen gesucht werden müssen, steigt der Anteil von Routingfehlern und Kollisionen extrem an. Die Bewegungsgeschwindigkeit von Knoten spielt dagegen kaum eine Rolle.

Insgesamt scheint AODV-basierter Geocast am besten geeignet für Netze mit einer mittleren Knotendichte und geringem Verkehrsaufkommen.

Hierarchischer Geocast

Die Werte bei hierarchischem Geocast sind wie folgt zu interpretieren:

Keine Geocast-Route beim Absender Falls der Absender ein DR ist, kennt er keine Route zu einem anderen DR, der sich näher am Zielgebiet befindet. Ist der Absender kein DR, kennt er keinen DR für seinen Präfix.

Keine Geocast-Route auf Zwischenknote Ein DR, der eine Nachricht von einem anderen Knoten weiter durch den Routing-Baum leiten sollte, kennt keinen anderen DR, der sich näher am Zielgebiet befindet.

Keine Unicast-Route Ein Knoten, der eine Nachricht weiterleiten sollte, kennt keine Unicast-Route zum Ziel der Nachricht.

Kollision Die Nachricht ist aufgrund einer Kollision beim Versand verloren gegangen.

Paket verzögert Die Nachricht wurde erst mehr als 60 Sekunden nach der Sendeanforderung im Zielgebiet ausgeliefert.

Routingfehler bei Absendern, die keine DRs sind, werden nicht gesondert aufgeführt. Zum einen tritt dieser Fehler in den Simulationen vernachlässigbar selten auf, zum anderen kann er sehr einfach behoben werden: Jeder Nicht-DR muss eine Sendeanforderung solange

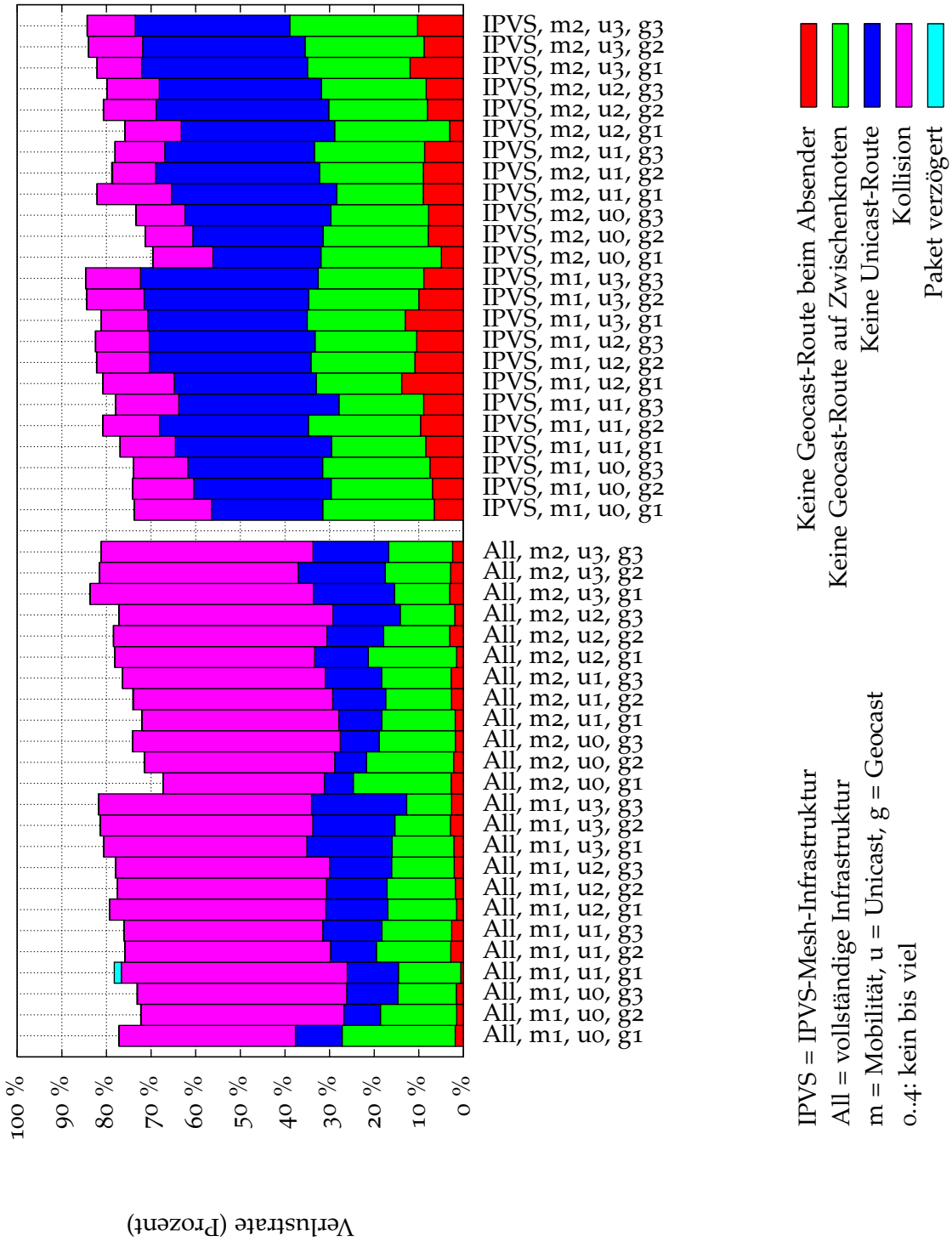


Abbildung 6.6: Nachrichtenverlustraten bei OLSR, 30 Knoten

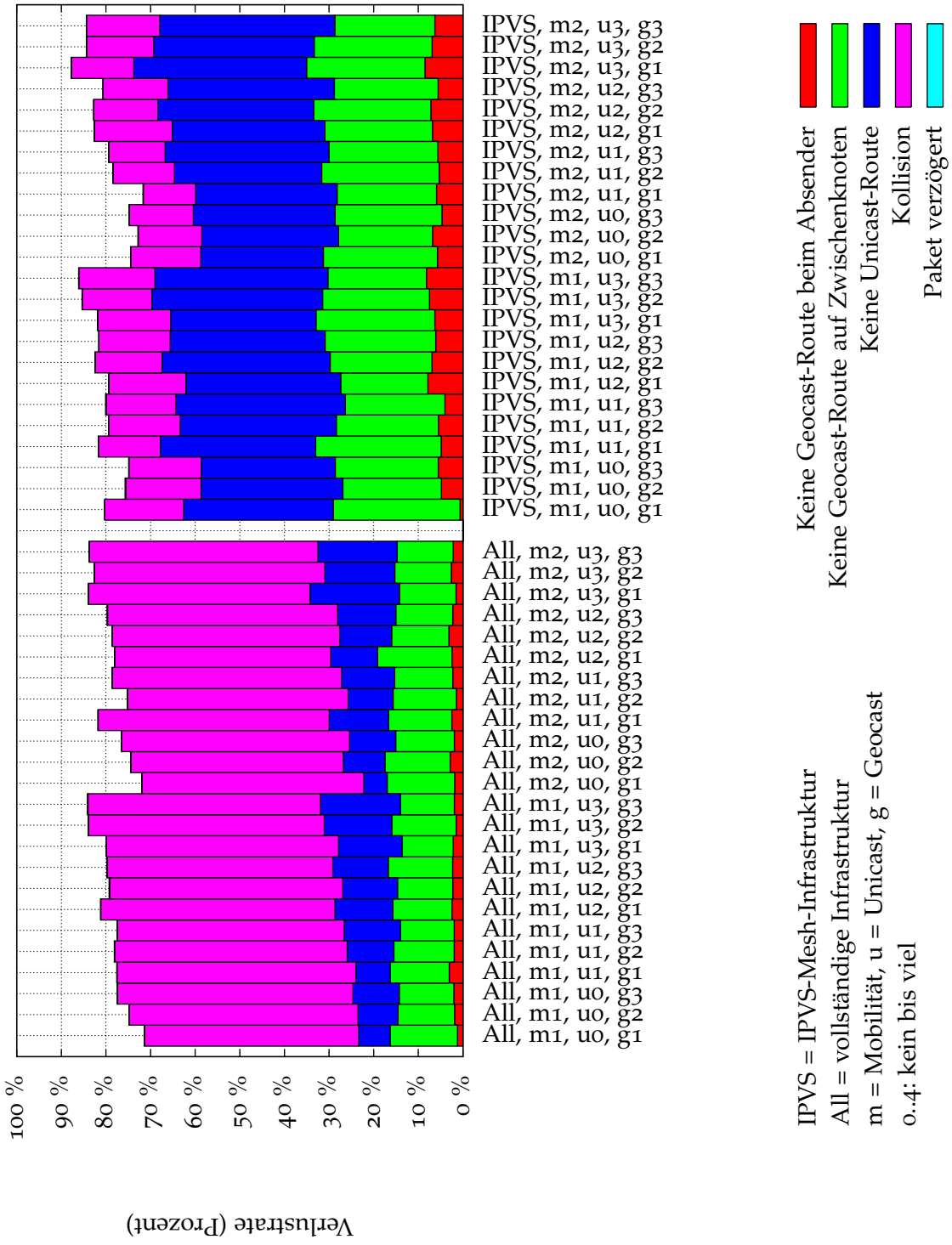


Abbildung 6.7: Nachrichtenverlustraten bei OLSR, 50 Knoten

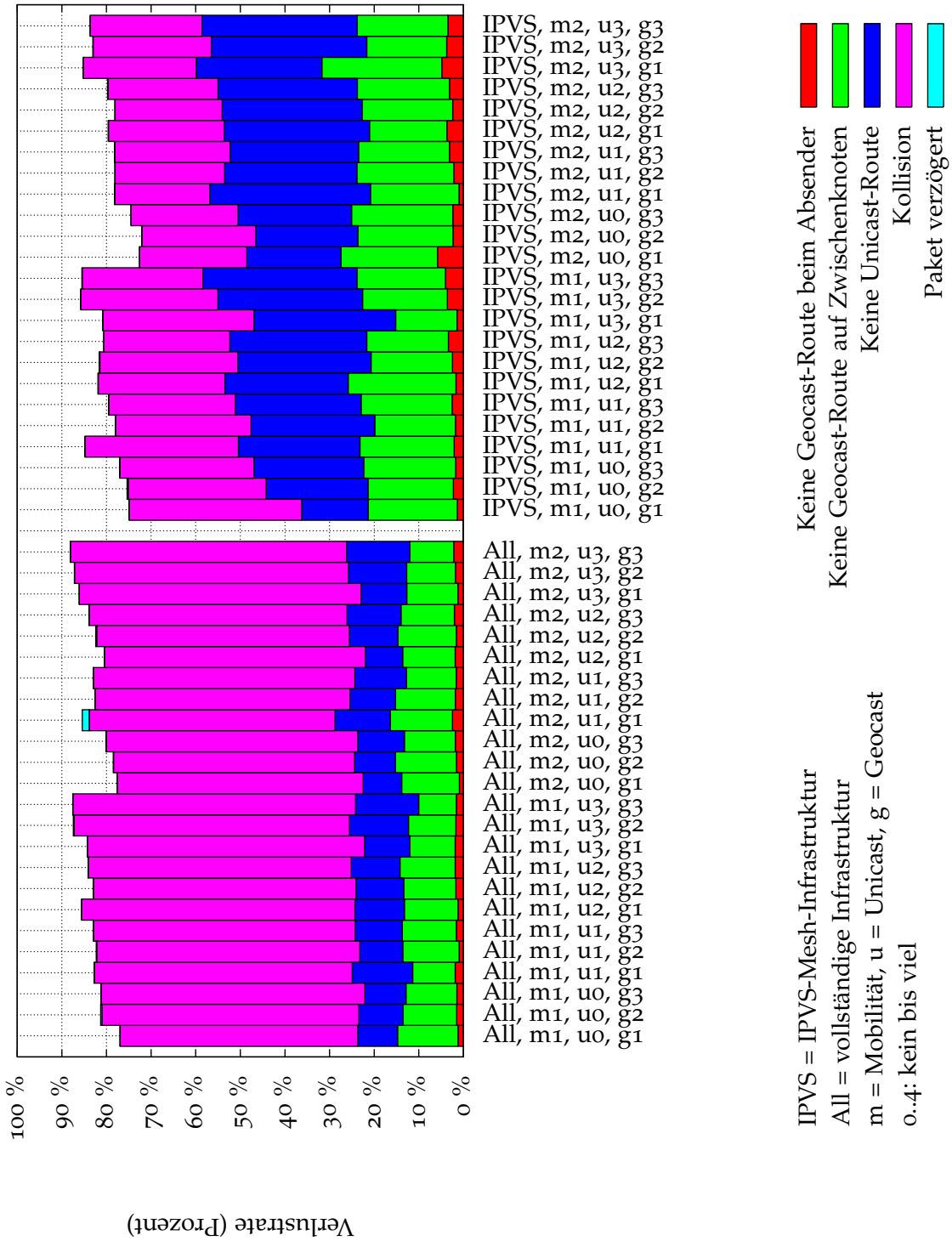


Abbildung 6.8: Nachrichtenverlustraten bei OLSR, 100 Knoten

aufbewahren, bis er wieder einen DR kennenlernt. Üblicherweise passiert dies, nachdem ein Knoten für einige Sekunden an einer Lokation verweilt.

Insgesamt sehen die OLSR-Ergebnisse deutlich schlechter aus als die Ergebnisse von AODV: Die Nachrichtenverlustrate liegt nur sehr selten unter 70%. Leicht zu erkennen ist, dass nahezu alle Nachrichten, die nicht durch Routingfehler verworfen werden, durch Kollisionen verloren gehen. Dies ist der hohen Grundlast durch OLSR zuzuschreiben. Da Knoten ständig `hello`- und `TC`-Nachrichten generieren und versenden, ist der Übertragungskanal bereits stark belastet.

Wie bei reaktivem Geocast gibt es auch in diesem Fall mehrere Gründe für Routingfehler: Netzwerkpartitionierungen, Mobilität und der Verlust von Routing-Informationen. Da sich die Menge der Routingfehler mit der Knotenanzahl kaum ändert, sind Netzwerkpartitionierungen hier wohl kaum ein Problem. Dass Mobilität ein starker Faktor ist, wird daraus ersichtlich, dass sich bei Simulationen ohne vollständige Infrastruktur der Anteil der Routingfehler in etwa verdoppelt. Die verbleibenden Fehlerfälle verteilen sich in etwa gleichmäßig auf Unicast- und Geocast-Routingfehler, was darauf hindeutet, dass sehr viele `TC`- bzw. `DRA`-Pakete verloren gehen.

Im Vergleich zu AODV fällt auch auf, dass in OLSR-Simulationen der Anteil verspäteter Nachrichten sehr gering ist. Allerdings ist die statistische Signifikanz dieses Wertes in den OLSR-Simulationen durchaus anzuzweifeln, da nur verhältnismäßig wenige Nachrichten ihr Zielgebiet erreicht haben.

6.4.2 Verzögerung von Nachrichten

Die Verzögerung von ausgelieferten Nachrichten, kumuliert über je fünf Simulationsläufe aller Parametersätze, ist in Abbildung 6.9 zu sehen. Die Verzögerung bezieht sich auf den zeitlichen Abstand zwischen Ankunft der Sendeanforderung und der ersten Auslieferung im Zielgebiet.

Bei beiden Protokollen werden etwa 93% der ausgelieferten Nachrichten in weniger als 10 Sekunden in ihr Zielgebiet transportiert. Erst bei Verzögerungen von über 20 Sekunden werden Unterschiede deutlich. Hierarchischer Geocast transportiert nahezu 100% der Nachrichten in maximal 30 Sekunden ins Zielgebiet, während bei reaktivem Geocast nur etwa 98% die 60-Sekunden-Marke unterbieten.

Zu erklären sind solche großen Verzögerungen durch verschiedene Effekte. Bei reaktivem Geocast muss zunächst eine Route aufgebaut werden, was allein schon über 10 Sekunden in Anspruch nehmen kann. Dazu kommen Verzögerungen bei der Auslieferung einer Geocast-Nachricht durch möglicherweise notwendiges Reparieren von Routen.

Übermäßige Verzögerungen bei hierarchischem Geocast können zum einen dadurch zustande kommen, dass ein Paket nicht den direkten Weg durch das Netz nimmt, sondern durch den DR-Baum geleitet werden muss. Hierarchischer Geocast enthält, genau wie die Unicast-Variante, keine Methode um Routingschleifen zu erkennen. Ein Unicast-Paket kann daher beliebig im Netz zirkulieren und wird nur durch eine Kollision oder die Ankunft im

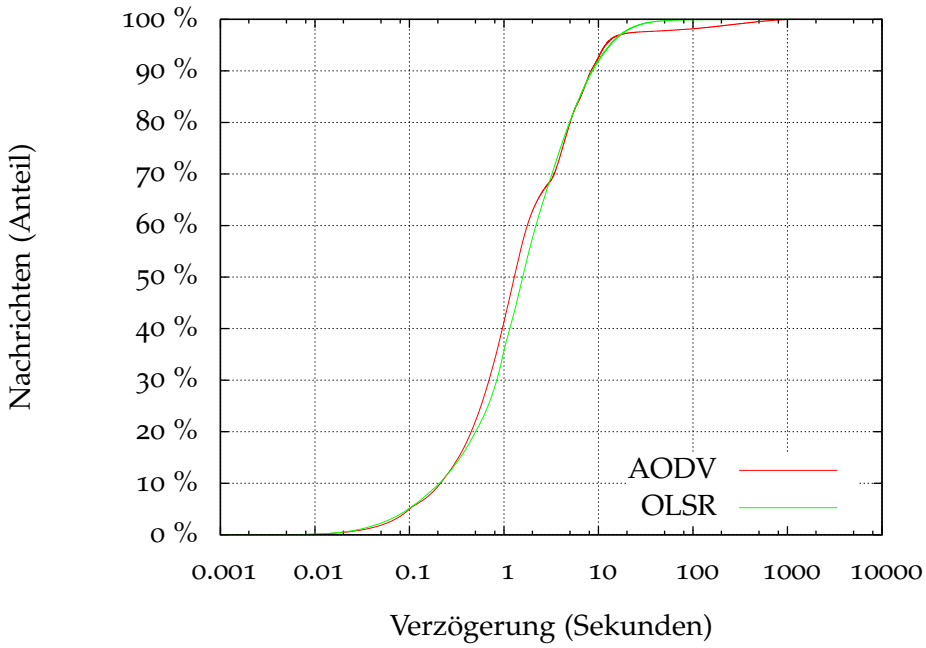


Abbildung 6.9: Kumulierte Verzögerung der Auslieferung von Nachrichten

Zielgebiet aus dem Netz entfernt. Die Duplikaterkennung von OLSR greift hier nicht immer, da ein Paket mitunter länger als die angenommene maximale Paketlebensdauer durch das Netz wandert und somit das für die Duplikaterkennung notwendige Wissen bereits entfernt wurde.

Um überhaupt nach längerer Zeit einen bereits besuchten Knoten erreichen zu können, muss es im Netz Geocast-Routingschleifen gegeben haben. Da die Latenz bei hierarchischem Geocast jedoch nur sehr selten über eine Minute steigt, ist davon auszugehen, dass Routingschleifen zwar auftreten, es sich aber nur um eine Randerscheinung handelt.

6.4.3 Abdeckung des Zielgebietes

Die Daten zur Abdeckung des Zielgebietes wurden für jeden Parametersatz über fünf Simulationsläufe gemittelt. Die Abdeckung des Zielgebietes durch eine Nachricht ist definiert als der Anteil der sich zum Zeitpunkt der ersten Auslieferung der Nachricht im Zielgebiet aufhaltenden Knoten, der die Nachricht auch tatsächlich erhalten hat.

Wie zu erwarten ist, stellt sich Fluten als sehr guter Verteilalgorithmus im Zielgebiet dar: Von allen ausgelieferten Nachrichten erreichten ca. 85% alle ihre Zielknoten. Etwa 95% der Nachrichten erreichen immerhin noch die Hälfte ihrer Zielknoten.

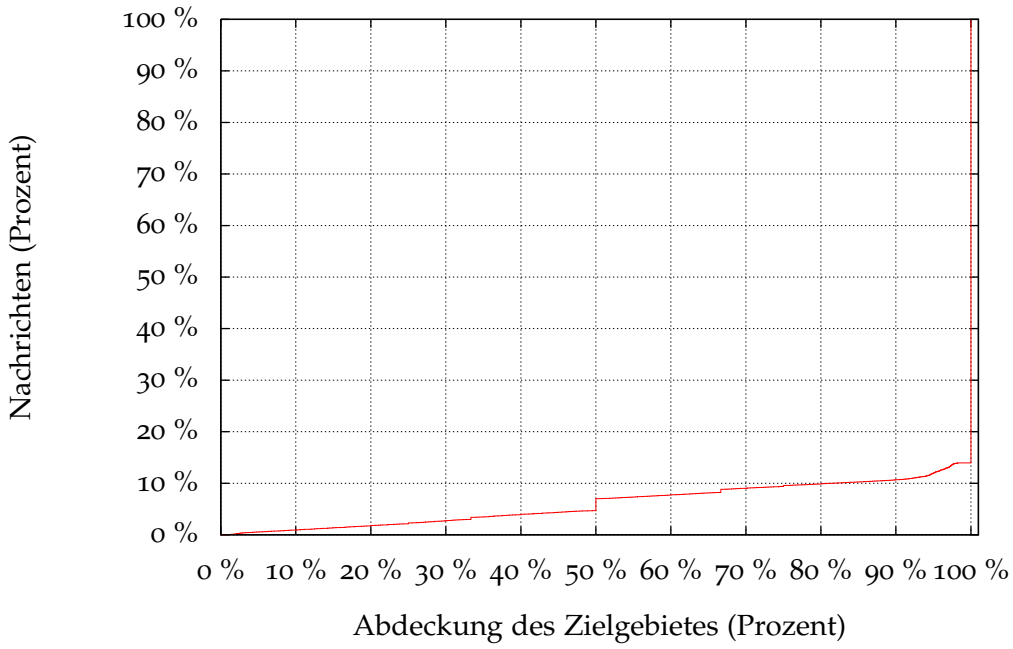


Abbildung 6.10: Kumulierte Abdeckung des Zielgebietes

Eine unvollständige Abdeckung des Zielgebietes kann aus der Mobilität von Knoten resultieren. Wenn ein Knoten sich während der ersten Auslieferung einer Nachricht im Zielgebiet befindet, selbiges jedoch verlässt, bevor ihn die Nachricht erreicht, wird dieser Knoten als nicht abgedeckt angesehen.

Ein weiterer Grund sind Kollisionen, bei denen einzelne Knoten oder Knotengruppen eine Nachricht nicht empfangen können und damit ebenfalls nicht abgedeckt werden.

6.4.4 Fairness

Um die Fairness beider Geocast-Routingprotokolle messen zu können, wurden getrennte Simulationen durchgeführt. Mobilität von Knoten und Unicast-Traffic wurden komplett abgeschaltet. Das angenommene Verkehrsmodell mit Pareto-Verteilungen wurde zu einer Gleichverteilung modifiziert. Die Ergebnisse sind in Abbildung 6.11 abgebildet.

Dies sichert zwei Eigenschaften: Zum einen werden keine Hot-Spots durch Häufungen von Sendeanfragen auf einzelnen Knoten oder übermäßig frequentierte Zielgebiete erzeugt. Zum anderen kann ein Knoten eine an einer Lokation hohe Last nun nicht während eines Simulationslaufes kompensieren, indem er sich an eine andere Lokation bewegt.

Reaktiver Geocast zeigt eine etwas gleichmäßigere Verteilung der Ereignisse. Die stärkere Ungleichverteilung bei hierarchischem Geocast ist allerdings durchaus zu erwarten. Je weiter oben im DR-Baum ein Knoten steht, desto mehr Nachrichten wird er für untergeordnete

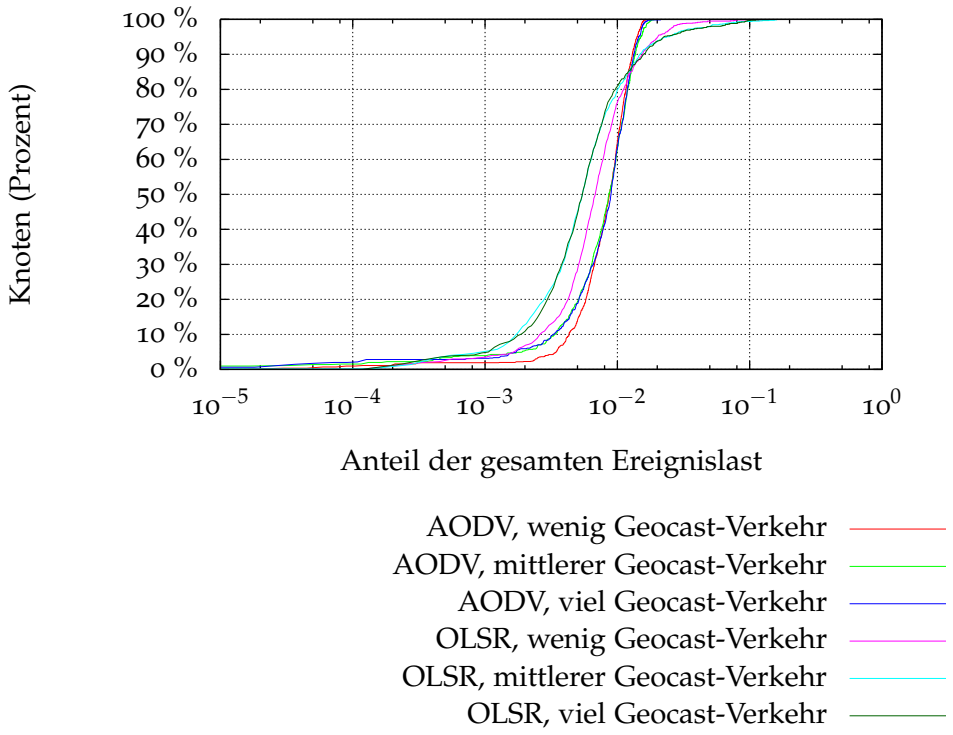


Abbildung 6.11: Kumulierte Verteilung von Sendeereignissen im Netz

Knoten weiterleiten müssen. Eine Verbesserung durch Abkürzungen ist kaum zu erwarten: Durch die Gleichverteilung der angefragten Lokationen ist es höchst unwahrscheinlich, dass irgendeine Lokation ausreichend Referenzen erhält, damit eine Abkürzung aufgebaut werden kann.

6.4.5 Pfadlänge

Die Werte für die Pfadlängen von Nachrichten (Abbildung 6.12) wurden über fünf Simulationsläufe aller Parametersätze kumuliert. Die maximal auftretende Pfadlänge bei reaktivem Geocast entspricht in etwa der erwarteten Netzgröße. Bei hierarchischem Geocast werden dagegen Pfadlängen sichtbar, die die Knotenanzahl selbst in den größten Simulationen übersteigen. Wie in Abschnitt 6.4.2 dargelegt, handelt es sich hierbei um Routingschleifen.

6.5 Fazit

Reaktiver Geocast erscheint geeignet für Netze mit einer geringer Mobilität und geringer Netzlast. Aufgrund des Umstands, dass Unicast-AODV nur für kleine Netze entwickelt

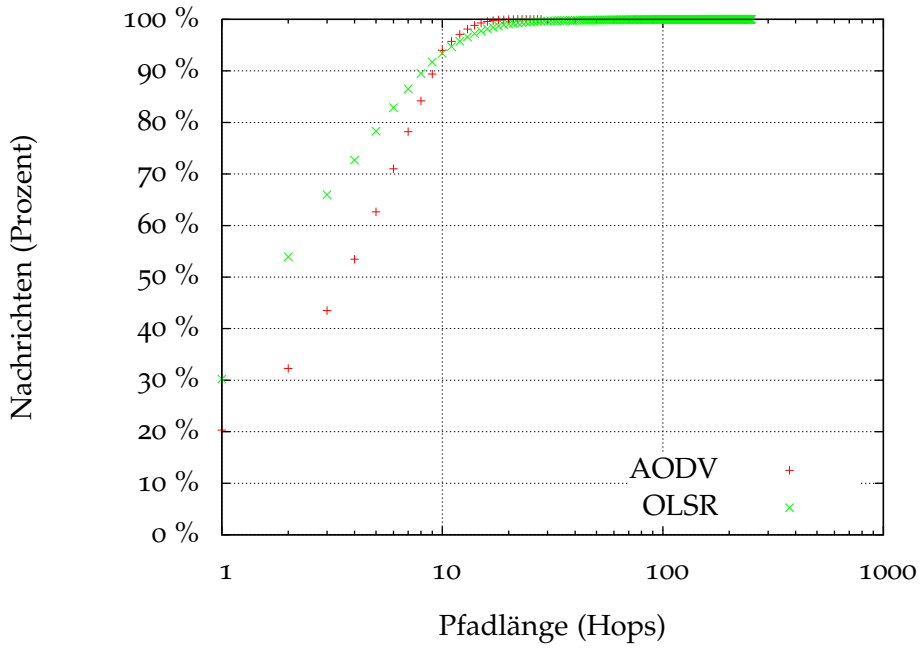


Abbildung 6.12: Kumulierte Pfadlänge von ausgelieferten Nachrichten

wurde und reaktiver Geocast wiederum nur eine optimierte Variante von Unicast-AODV ist, ist nicht damit zu rechnen, dass reaktiver Geocast mehr als einige hundert Knoten effizient versorgen kann.

Hierarchischer Geocast verspricht eine deutlich bessere Skalierbarkeit. Jedoch bleiben zwei kritische Probleme zu lösen, die den Einsatz von hierarchischem Geocast gänzlich unmöglich machen können. Zum einen müssen die auftretenden Routingschleifen beseitigt werden – notfalls dadurch, dass ein kreisendes Paket verworfen wird. Zum anderen wird die Funktion des Protokolls durch die übermäßigen Paketverluste stark beeinträchtigt. Es ist damit zu rechnen, dass in einem realen Netzwerk deutlich weniger Paketverluste auftreten, da dort zumeist eine verbesserte OLSR-Variante zum Einsatz kommt. Abschließend kann dies nur ein Test in einem solchen Netzwerk belegen, der allerdings den Rahmen der vorliegenden Arbeit sprengt.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die vorliegende Arbeit beschäftigte sich mit dem Entwurf von an die Eigenschaften von Mesh-Netzwerken angepassten effizienten Geocast-Routingprotokollen. Dazu wurden zunächst bestehende Unicast- und Multicast-Routingprotokolle für Mesh-Netze untersucht. Die Entwicklung eines Systemmodells und das Herausarbeiten der besonderen Eigenschaften von Geocast in Mesh-Netzwerken legte den Grundstein für die eigentiche Entwicklung von Geocast-Routingverfahren.

Ansätze zur Verwendung bestehender Multicast-Routingprotokolle zum Transport von Geocast-Nachrichten sind gescheitert, da die Multicast-Protokolle keine Rücksicht auf die Lokalitätseigenschaften von Geocast-Domänen nehmen. Eine Modifikation der Multicast-Routingprotokolle erbrachte ein voraussehbar schlechtes Kosten-Nutzen-Verhältnis: Aller Aufwand, der getrieben werden muss, um ein bestehendes Protokoll umzubauen, hätte gleichzeitig den Verlust wünschenswerter Eigenschaften dieser Protokolle beduetet. Daher wurde damit fortgefahren, basierend auf den Ideen der Multicast-Protokolle neue Routingprotokolle zu entwerfen.

Der Entwurf der von Multicast inspirierten Multi-Path-Routingprotokolle war ebensowenig erfolgversprechend wie das direkte Aufsetzen auf Multicast. In AODV-basierten Systemen steht zu erwarten, dass die Optimierungen viele Fehlerfälle produzieren, d.h. dass viel Entwicklungsaufwand getrieben werden muss, der später jedoch kaum einen Einfluss auf das Ergebnis hat. OLSR-basierte Systeme schnitten sogar noch schlechter ab: Hier waren die vorgeschlagenen Optimierungen nicht umsetzbar oder haben direkt zu Fehlverhalten geführt.

Als beste Lösungen haben sich zweistufige Single-Path-Routingansätze erwiesen. Die erste Stufe ist darauf optimiert, Nachrichten zwischen weit voneinander entferntem Sender und dem Zielgebiet zu transportieren. Die zweite Stufe dagegen verteilt eine Nachricht optimal innerhalb des begrenzten Zielgebietes.

Als Protokoll für die zweite Stufe wurde geolimitierter Broadcast präsentiert. Damit können Nachrichten innerhalb einer nicht-partitionierten Geocast-Domäne geflutet werden. Für die erste Stufe wurden zwei unterschiedliche Protokolle entworfen. Für Netze, die AODV als Unicast-Routingprotokoll einsetzen, eignet sich der reaktive Geocast, der durch eine direkte Anpassung aus Unicast-AODV entstanden ist. Die wesentliche Verbesserung gegenüber Unicast-AODV war der Einsatz von Expanding Ring Search, um das Zielgebiet zu finden. Für auf OLSR aufsetzende Netze wurde ein wesentlich komplexeres System, basierend auf

Arbeiten für ein Overlay-Netzwerk zur Verteilung von Geocast-Nachrichten im Internet, entwickelt. Dabei wurde der Flutungsaufwand begrenzt, indem per geolimitiertem Broadcast stets nur innerhalb einer Geocast-Domäne geflutet wurde. Dies führt dazu, dass, ein stabiles Overlay-Netzwerk vorausgesetzt, der netzweite Nachrichtenaufwand zum Aufbau des Overlay-Netzwerkes nur einem kleinen Vielfachen des Aufbaus einer Route im Fall von reaktivem Geocast entspricht.

Beide Protokolle wurden zur Evaluierung im Netzwerksimulator ns2 implementiert. In den Simulationen zeigte sich, dass der reaktive Geocast in AODV-basierten Netzen sehr schnell durch die Netzlast begrenzt wurde, aber auch, dass das Netz eine gewisse Dichte aufweisen muss, damit ein nennenswerter Anteil an Geocast-Nachrichten erfolgreich zugestellt werden kann. Da dieses Geocast-Protokoll nur ein abgewandeltes Unicast-AODV-Protokoll darstellt und die Zielgröße von Netzen, in denen Unicast-AODV eingesetzt werden soll, deutlich unter der Größe von Metropolitan Area Networks liegt, wird der reaktive Geocast alleine kaum tauglich sein, um Geocast in Metropolitan Area Networks zu realisieren. Ein klarer Vorteil ist jedoch, dass die Implementierung von reaktivem Geocast sehr einfach ist. Reaktiver Geocast könnte daher wichtig werden, wenn es um die Bereitstellung von Geocast-Diensten innerhalb beschränkter Netzbereiche geht. Ein Beispiel dafür sind große Netze, in denen einzelne Mesh-Parzellen über ein verkabeltes Backbone-Netz verbunden sind.

Die Ergebnisse von hierarchischem Geocast deuten zunächst auf ein grundlegendes Problem mit dem Unicast-Routingprotokoll hin. Bereits in dem im Vergleich zum anvisierten Zielsystem relativ kleinen simulierten Netzwerk war der Übertragungskanal völlig überlastet. Es lässt sich zwar ein Trend ausmachen, dass Systeme mit vielen statischen Knoten besser funktionieren als Systeme mit rein mobilen Knoten. In allen Fällen war allerdings ein sehr großer Anteil der Fehler auf fehlende Unicast-Routen zurückzuführen. Um dieses Overlay-System einzusetzen, muss also zunächst ein geeignetes Underlay-Netzwerk vorhanden sein. Da die Tests im Simulator nur mit einer als veraltet anzusehenden und bekanntermaßen ineffizienten RFC-konformen OLSR-Implementierung durchgeführt werden konnten, existiert ein besseres Underlay-Netzwerk in der Form aktueller OLSR-Varianten möglicherweise schon.

7.2 Offene Fragen

Der Fokus dieser Arbeit lag auf dem grundlegenden Entwurf von Geocast-Routingprotokollen für Mesh-Netze. Aufgrund des beschränkten Umfanges konnten nicht alle aufkommenden weitergehenden Fragestellungen behandelt werden. Abschließend sollen hier die Punkte aufgeführt werden, deren weitere Untersuchung die größten Verbesserungen bei der Weiterentwicklung von Geocast-Routingprotokollen verspricht.

Tests in realen Systemen Wie Erfahrungen bei Entwicklung und Einsatz von Routingprotokollen für Mesh-Netze gezeigt haben [Aic], ist es unerlässlich ein vorgeschlagenes Protokoll in einem realen System weiteren Tests zu unterziehen um festzustellen, welche in der Simulation gemachten Abstraktionen zu Problemen in realen Systemen führen.

Optimiertes Fluten Bei der Untersuchung einzelner Simulationsläufe in AODV-basierten Netzen hat sich gezeigt, dass es beim Fluten in allen Varianten (Expanding Ring Search, geolimitierter Broadcast,...) zu einer erhöhten Kollisionswahrscheinlichkeit zwischen Nachrichten kommt. Für die Zwecke der Simulation ließ sich die Kollisionswahrscheinlichkeit mit Hilfe einer zufälligen Verzögerung beim Senden von Nachrichten einschränken.

Unicast-OLSR bietet über die Multi-Point-Relays einen Mechanismus zum optimierten Fluten, der genau dem beschriebenen Problem entgegenwirken soll. Eine Anpassung an den reaktiven Geocast für AODV-basierte Netze könnte daher die Effizienz der Verteilung weiter verbessern. Möglicherweise kann auch der Einsatz probabilistischer Protokolle zu Verbesserungen führen.

Feedback Rückmeldungen über den Verbleib oder Verlust einer Nachricht können für mehrere Zwecke benutzt werden. Neben aufwändiger zu realisierenden Eigenschaften, wie Überlastvermeidung und Flusskontrolle, können Rückmeldungen im einfachsten Fall dazu verwendet werden, um die Übertragung verloren gegangener Nachrichten zu wiederholen.

Reaktiver Geocast bietet lediglich negatives Feedback, falls ein Routingfehler aufgetreten ist. Nachrichtenverluste erzeugen kein Feedback. Hierarchischer Geocast bietet gar kein Feedback. Konzeptionell ist es jedoch relativ einfach, solches Feedback nachzurüsten. Das Hauptproblem ist, einen einzelnen Knoten auszuwählen, der den Empfang einer Nachricht bestätigt. Andernfalls kann es durch die anfallenden Mengen an Feedback-Nachrichten zu Implosionsproblemen und Netzüberlastungen kommen.

Für hierarchischen Geocast bietet es sich an, den Designated Router mit der Aufgabe zu betrauen, beim Empfang einer Geocast-Nachricht für sein Zielgebiet ein `ACK` an den Absender zu senden. Bei reaktivem Geocast kann als bestätigender Knoten derjenige ausgewählt werden, der eine Nachricht von der Unicast-Übertragung zur Übertragung per geolimitiertem Broadcast konvertiert.

Weitere Routen-Wiederverwendung bei reaktiven Geocast Sowohl Unicast-AODV als auch die Geocast-Erweiterungen versuchen, bereits bekannte Informationen soweit wie möglich wiederzuverwenden. In der vorliegenden Arbeit beschränkt sich die Wiederverwendung von Geocast-Routen auf genau passende Routen.

Jedoch können prinzipiell auch noch genauere Einträge verwendet werden: Existiert auf einem Knoten ein genauerer Eintrag, d.h. ein Eintrag für den Präfix einer Lokation, die der angefragten Lokation untergeordnet ist, so können Nachrichten an die angefragte Lokation über die bestehende Route weitergeleitet werden – es wird auf jeden Fall ein Knoten im adressierten Zielgebiet erreicht.

Stochastische Routenoptimierung bei reaktivem Geocast Die Effektivität der Routenoptimierung bei reaktivem Geocast ist durch den Radius des limitierten Flutens, d.h. durch die in Kauf genommene zusätzliche Netzlast, beschränkt. Wird eine stochastische Methode

verwendet, um den Flutungsradius zu wählen oder um zu entscheiden, ob eine Nachricht noch weitergeleitet werden soll, kann die Effektivität der Optimierung gesteigert werden, ohne die Netzlast zu stark zu erhöhen.

Blocking Expanding Ring Search Expanding Ring Search (ERS) ermöglicht es, die Netzlast bei Flutungen zu senken, falls zu erwarten steht, dass sich das gesuchte Ziel in der näheren Umgebung des suchenden Knotens aufhält. Falls das Ziel sich jedoch einmal nicht in der Nähe befindet, verursacht ERS häufige Neuübertragungen der Suchnachrichten in der Umgebung des suchenden Knotens.

Mit Hilfe der Blocking Expanding Ring Search (B-ERS, [PKP06]) und der Verbesserung Blocking Expanding Ring Search Plus (B-ERS-Plus, [ARMOK08]) können diese Duplikate vermieden werden. Eine B-ERS-Nachricht wird dabei von jedem Knoten zwischengespeichert und erst nach einer gewissen Zeit weitergeleitet. Diese Zeit ist abhängig davon, wie viele Hops die Nachricht bereits passiert hat. Ein vorgeschlagener Wert ist $2 * \text{Hopcount} * \text{Übertragungszeit zwischen benachbarten Knoten}$. Wenn ein Knoten eine Antwort auf die Nachricht versenden kann, wird gleichzeitig noch eine Stopp-Nachricht abgeschickt. Stopp-Nachrichten werden ohne Verzögerung weitergeleitet und weisen Knoten an, weitere Neuübertragungen der B-ERS-Anfrage zu unterlassen. B-ERS-Plus optimiert die Verteilung der Stopp-Nachrichten nochmals im Hinblick auf die Mobilität von Knoten.

Der Einsatz von B-ERS kann die Netzlast und Latenz der Routensuche von reaktivem Geocast möglicherweise nochmals senken.

Mehrstufiges Geocast-Routing mit reaktivem Geocast Reaktiver Geocast auf Basis von AODV ist nach den bei der Implementierung der Simulation gemachten Erfahrungen sehr einfach zu implementieren. Die Größe von Netzwerken, die AODV als Unicast-Routingprotokoll verwenden, ist jedoch beschränkt, so dass ein direkter Einsatz in Metropolitan Area Networks nicht möglich erscheint. Falls jedoch ein großes Netz in kleinere Parzellen aufgeteilt werden kann, könnte reaktiver Geocast innerhalb dieser Parzellen zum Einsatz kommen.

Literaturverzeichnis

[Aic] Aichele, Corinna „Elektra“: *The OLSR Story*. https://www.open-mesh.net/wiki/the-olsr-story (Zitiert auf den Seiten 18, 26 und 112)

[ARMOK08] Al-Rodhaan, M. ; Mackenzie, L.M. ; Ould-Khaoua, M.: Improvement to Blocking Expanding Ring Search for MANETs / University of Glasgow, Department of Computing Science. Dept of Computing Science, University of Glasgow, 2008. – Forschungsbericht. – 13 S. – ISBN TR–2008–286 (Zitiert auf Seite 114)

[Bah06] Bahr, Michael: Proposed routing for IEEE 802.11s WLAN mesh networks. In: *WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet*. New York, NY, USA : ACM, 2006. – ISBN 1–59593–510–X, S. 5 (Zitiert auf Seite 16)

[CD09] Clausen, T. ; Dearlove, C.: *The Optimized Link State Routing Protocol version 2*. http://tools.ietf.org/id/draft-ietf-manet-olsrv2-08.txt. Version: März 2009 (Zitiert auf Seite 19)

[CJ03] Clausen, T. ; Jacquet, P. ; Clausen, T. (Hrsg.) ; Jacquet, P. (Hrsg.): *Optimized Link State Routing Protocol (OLSR)*. United States, 2003 (Zitiert auf Seite 17)

[DC04] De Couto, Douglas S. J.: *High-Throughput Routing for Multi-Hop Wireless Networks*, Massachusetts Institute of Technology, Diss., Juni 2004. http://pdos.csail.mit.edu/papers/grid:decouto-phd/thesis.pdf (Zitiert auf Seite 26)

[DR03] Dürr, Frank ; Rothermel, Kurt: On a Location Model for Fine-Grained Geocast. In: Dey, Anind K. (Hrsg.) ; Schmidt, Albrecht (Hrsg.) ; McCarthy, Joseph F. (Hrsg.): *Proceedings of the Fifth International Conference on Ubiquitous Computing : UbiComp '03 ; Seattle, WA, Oct. 12-15, 2003* Bd. 2864, Springer-Verlag, Oktober 2003 (Lecture Notes in Computer Science). – ISBN 3–540–20301–X, 18–35 (Zitiert auf den Seiten 22, 23 und 29)

[DR06] Dürr, Frank ; Rothermel, Kurt: An Overlay Network for Forwarding Symbolically Addressed Geocast Messages. In: *Proceedings of the 15th International Conference on Computer Communications and Networks (ICCCN '06); Arlington, VA, October 9-11, 2006*, IEEE, Oktober 2006. – ISBN 0–4244–0572–6, 427–434 (Zitiert auf den Seiten 22, 23, 24, 29, 57 und 61)

[DR08] Dürr, Frank ; Rothermel, Kurt: An Adaptive Overlay Network for Worldwide Geographic Messaging. In: *Proceedings of the 22nd IEEE International Conference on Advanced Information Networking and Applications (AINA 2008); Gino-wan, Okinawa, Japan, March 25-28, 2008*, IEEE, März 2008, 875–882 (Zitiert auf den Seiten 22, 24, 29, 57, 61, 79 und 80)

[IN96] Imieliński, Tomasz ; Navas, Julio C.: *RFC 2009 – GPS-Based Addressing and Routing (Experimental)*. http://tools.ietf.org/html/rfc2009. Version: November 1996 (Zitiert auf den Seiten 22 und 37)

[IN97] Imieliński, Tomasz ; Navas, Julio C.: GeoCast – Geographic Adressing and Routing / Rutgers,, The State University. 1997. – Forschungsbericht (Zitiert auf Seite 22)

[IN98] Imieliński, Tomasz ; Navas, Julio C.: Multi-Hop Dynamic Geographic Routing. 1998. – Forschungsbericht (Zitiert auf Seite 22)

[IN99] Imieliński, Tomasz ; Navas, Julio C.: GPS-based geographic addressing, routing, and resource discovery. In: *Commun. ACM* 42 (1999), Nr. 4, S. 86–92. http://dx.doi.org/10.1145/299157.299176. – DOI 10.1145/299157.299176. – ISSN 0001–0782 (Zitiert auf Seite 22)

[Inc08] Inc., Google: *Google WiFi for Mountain View*. Website. http://wifi.google.com/. Version: 2008 (Zitiert auf Seite 11)

[JMC+01] Jacquet, P. ; Muhlethaler, P. ; Clausen, T. ; Laouiti, A. ; Qayyum, A. ; Viennot, L.: Optimized link state routing protocol for ad hoc networks, 2001, 62–68 (Zitiert auf Seite 17)

[LJM+03] Laouiti, Anis ; Jacquet, Philippe ; Minet, Pascale ; Viennot, Laurent ; Clausen, Thomas ; Adjih, Cédric: Multicast Optimized Link State Routing / INRIA. Version: 2003. http://hal.inria.fr/inria-00071865/en/. 2003 (RR-4721). – Research Report (Zitiert auf Seite 21)

[Lop04] Lopatic, Thomas: *olsrd Link Quality Extensions*. Online. http://www.olsr.org/docs/README-Link-Quality.html. Version: Dezember 2004 (Zitiert auf Seite 26)

[NAL07] Neumann, Axel ; Aichele, Corinna „Elektra“ ; Lindner, Marek: B.A.T.M.A.N. Status Report. Version: Juni 2007. http://downloads.open-mesh.net/batman/misc/batman-status.pdf. 2007. – Forschungsbericht (Zitiert auf Seite 19)

[NALS08] Neumann, A. ; Aichele, C. ; Lindner, M. ; S.Wunderlich: *Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)*. http://tools.ietf.org/id/draft-wunderlich-openmesh-manet-routing-00.txt. Version: April 2008 (Zitiert auf Seite 19)

[Nav01] Navas, Julio C.: *Geographic Routing in a Datagram Internetwork*, Graduate School-New Brunswick, Rutgers, The State University of New Jersey, Diss., 2001 (Zitiert auf Seite 22)

[PKP06] Park, Incheon ; Kim, Jinguk ; Pu, Ida: Blocking Expanding Ring Search Algorithm for Efficient Energy Consumption in Mobile Ad Hoc Networks. In: *WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services*. Les Ménuires (France), 01 2006, 191-195. – http://citi.insa-lyon.fr/wons2006/index.html Session 6 : Energy-aware protocols C.: Computer Systems Organization, C.: Computer Systems Organization/C.2: COMPUTER-COMMUNICATION NETWORKS (Zitiert auf Seite 114)

[PR99] Perkins, C. ; Royer, E.: Ad-hoc on-demand distance vector routing. In: *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999, 90–100 (Zitiert auf Seite 16)

[Pro08] Projekt, NEXUS: *SFB 627: Nexus. Umgebungsmodelle für Mobile Kontextbezogene Systeme*. online. `http://www.nexus.uni-stuttgart.de/`. Version: 04 2008 (Zitiert auf Seite 91)

[Pro09] Projekt, Freifunk: *Homepage des Freifunk Projekt*. `http://www.freifunk.net/`. Version: Mai 2009 (Zitiert auf den Seiten 11, 16 und 18)

[Ros09] Ros, Francisco J.: *UM-OLSR*. Website. `http://masimum.inf.um.es/?Software:UM-OLSR`. Version: January 2009 (Zitiert auf den Seiten 65 und 78)

[RP99] Royer, Elizabeth M. ; Perkins, Charles E.: Multicast operation of the ad-hoc on-demand distance vector routing protocol. In: *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA : ACM, 1999. – ISBN 1581131429, 207–218 (Zitiert auf Seite 20)

[RP00] Royer, Elisabeth M. ; Perkins, Charles E.: *Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing*. online. Version: Juli 2000. `http://tools.ietf.org/html/draft-ietf-manet-maodv-00.txt` (Zitiert auf Seite 20)

[Ste05] Stepanov, Illya: *CANU Mobility Simulation Environment (CanuMobiSim)*. `http://canu.informatik.uni-stuttgart.de/mobisim/`. Version: November 2005 (Zitiert auf Seite 91)

[TLG+08] Tønnesen, Andreas ; Lopatic, Thomas ; Gredler, Hannes ; Petrovitsch, Bernd ; Kaplan, Aaron ; Tücke, Sven-Ola u. a.: *OLSR Routing Daemon Homepage*. Website. `http://www.olsr.org/`. Version: 2008 (Zitiert auf Seite 18)

[Wik09] Wikipedia: *IEEE 802.11s — Wikipedia, Die freie Enzyklopädie*. `http://de.wikipedia.org/w/index.php?title=IEEE_802.11s&oldid=58945746`. Version: 2009. – [Online; Stand 20. Mai 2009] (Zitiert auf den Seiten 16 und 17)

Alle URLs wurden zuletzt am 11.11.2009 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Damian Cassius Philipp)