



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Eine systematische Literaturstudie zu beaconless Algorithmen für drahtlose Ad hoc und Sensornetze

Bachelorarbeit
zur Erlangung des Grades
BACHELOR OF SCIENCE
im Studiengang Informatik

vorgelegt von

Daniel Vivas Estevao

Betreuer: Florentin Neumann, M. Sc., Institut für Informatik,
Universität Koblenz-Landau

Erstgutachter: Prof. Dr. Hannes Frey, Institut für Informatik,
Universität Koblenz-Landau

Zweitgutachter: Florentin Neumann, M. Sc., Institut für
Informatik, Universität Koblenz-Landau

Koblenz, im Oktober 2014

Kurzfassung

Drahtlose Sensornetzwerke (DSN) sind Rechnernetze von Sensorknoten. In positionsbasierten Protokollen senden solche Knoten ihre Positionsdaten periodisch als Nachrichten (Beacon) über Funk an umliegende Knoten. Beacons bringen jedoch Nachteile mit sich, die sog. „beaconless“ Algorithmen zu lösen versuchen. Diese Algorithmen benutzen keine Beacons, sollen aber die gleichen Einsatzgebiete abdecken wie ihre Beacon-behafteten Gegenstücke.

Trotz ihrer Wichtigkeit im Forschungsgebiet der Rechnernetze sind beaconless Verfahren nach unserem derzeitigen Kenntnisstand noch nicht systematisch untersucht worden.

Ziel dieser Arbeit ist eine möglichst umfassende und systematische Übersicht von beaconless Algorithmen seit der Jahrtausendwende. Relevante Artikel werden anhand ihrer Gemeinsamkeiten kategorisiert und ihre Unterschiede werden gegenübergestellt. Diese Arbeit dient somit als Nachschlagewerk zum aktuellen Forschungsstand von beaconless Algorithmen sowie als Basis zum Aufdecken von Forschungslücken.

Abstract

Wireless sensor networks (WSN) are computer networks consisting of sensor nodes. In position-based protocols nodes broadcast messages containing their position information periodically. These messages are called beacons. The use of beacons however has downsides, which led to the emerging of beaconless algorithms.

Despite their importance in the research area of computer networks and to the best of our knowledge, beaconless algorithms have not been systematically surveyed.

The aim of this work is a comprehensive, systematic review of beaconless algorithms since the turn of the millennium. Relevant articles are categorized based on their similarities and their differences are compared. Thus, this work serves as a reference to the current state of research of beaconless algorithms and as a basis for finding research gaps.

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – ins besondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-Rom).

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja nein

Koblenz, den 15. Oktober 2014

Inhaltsverzeichnis

1	Einleitung	11
2	Grundlagen	15
2.1	Weiterleitung	15
2.1.1	Geographisches Routing	17
2.1.2	Greedy Routing	17
2.1.3	Recovery Routing	17
2.2	VANET	20
3	Methoden und Werkzeuge	21
3.1	Methode zur Literaturrecherche	21
3.1.1	Suchprozess	21
3.1.2	Auswahlkriterien	22
3.1.3	Werkzeuge	23
3.2	Methode zur Kategorisierung der Verfahren	23
4	Topologiekontrolle	25
4.1	Graphenmodelle	25
4.1.1	Unit Disk Graph	26
4.1.2	Graph der Partiellen Delaunay Triangulation (PDT)	26
4.1.3	Gabriel Graph	27
4.1.4	Relative Neighborhood Graph	27
4.2	Verfahren	27
5	Unicast Routing	31
5.1	Greedy Routing	32
5.1.1	Aggressive contention	32
5.1.2	Non-Aggressive Contention	37
5.1.3	Hybrid contention	43
5.2	Recovery Routing	45

6	Multicast Routing	57
7	VANET	63
7.1	Routing	63
7.1.1	Aggressive Contention	64
7.1.2	Non-Aggressive Contention	64
7.2	Sicherheit	66
8	Fazit	69
A	Appendix	71
A.1	Unicast Routing	71
A.1.1	Aggressive Contention	71
A.1.2	Non-aggressive Contention	80
A.1.3	Hybrid Contention	93
A.1.4	Recovery Routing	96
A.2	Multicast Routing	98
A.3	VANET	100
A.3.1	Aggressive Contention	100
A.3.2	Non-aggressive Contention	101
B	Inhalte der CD-ROM	103

Tabellenverzeichnis

4.1	Übersicht Topologiekontrolle-Protokolle	30
5.1	Übersicht Aggressive Contention Unicast Routing Protokolle	38
5.2	Übersicht Non-Aggressive Contention Unicast Routing Protokolle	44
5.3	Übersicht Hybrid Contention Unicast Routing Protokolle	46
5.4	Übersicht Recovery Unicast Routing Protokolle	55
6.1	Übersicht Multicast Routing Protokolle	62
7.1	Übersicht VANET Protokolle	67
A.1	BLR	71
A.2	CBGR	72
A.3	BGR	73
A.4	EABGR	73
A.5	BMR	74
A.6	BLR-E	74
A.7	DAF	75
A.8	DBO	75
A.9	ABC	76
A.10	RBRR	77
A.11	IFSA	78
A.12	CNB-GEDIR	78
A.13	MRR	79
A.14	SIF	80
A.15	OGF	81
A.16	GRoVar	82
A.17	IGF	83
A.18	CBRR	84
A.19	GDBF	85
A.20	BLGF	85
A.21	BOSS	86

A.22 EBGR	87
A.23 EBGRES	88
A.24 EEGR	88
A.25 MEGAN	89
A.26 PF	90
A.27 PFMAC	90
A.28 BGM	91
A.29 GeRaF	91
A.30 MACRO	92
A.31 PSGR	92
A.32 CBF	93
A.33 HCGR	94
A.34 CR	96
A.35 NB-FACE	96
A.36 BFP/AR	97
A.37 GEM	98
A.38 DMPB	98
A.39 BRUMA	99
A.40 RPBL	100
A.41 CBFS	100
A.42 BRAVE	101
A.43 CBBPF	101
A.44 IB	102

Abbildungsverzeichnis

1.1	Nachbarschaft eines Knotens	12
2.1	Geographisches Greedy Routing	16
2.2	Void Area	18
2.3	Face Routing	19
3.1	Artikel in yED (Ausschnitt)	24
4.1	Topologiekontrolle mit minimalem Spannbaum	26
4.2	PDT Graph	27
4.3	Gabriel Graph	28
5.1	Weiterleitungsbereiche	33
5.2	Winkel γ	36
5.3	MRR Weiterleitungsbereich	37
5.4	Optimal Relay Position	42
5.5	Clockwise Relaying	48
5.6	PSGR-Bypass	49
5.7	NB-FACE	51
5.8	Rotational Sweep: Sweep Circle	52
5.9	Rotational Sweep: Twisting Triangle	52
5.10	Weiterleitungsentscheidung in ARROW	53
6.1	Multicast Routing	58
6.2	Ineffizienter Multicast	59
6.3	Effizienterer Multicast	59
6.4	Geocast Variante 1	61
6.5	Geocast Variante 2	61
7.1	Weiterleitung in BRAVE	65
A.1	Optimal Reception Perimeter in ABC	75
A.2	HCGR Recovery Routing	95

Kapitel 1

Einleitung

Ein drahtloses Sensornetzwerk (DSN) ist ein Rechnernetz von *Sensorknoten*. Sensorknoten sind kleine Computer, die mittels Sensoren Informationen über ihre Umgebung sammeln können und untereinander über Funk kommunizieren können. Dazu besitzen sie eine oder mehrere Antennen, die zum Empfangen und/oder Versenden von Daten benutzt werden. Von einem Sensorknoten über Funk versendete Daten nennen wir *Datenpakete* oder *Nachrichten*. Den Bereich, den ein Knoten bei voller Sendeleistung über Funk erreichen kann, nennen wir seinen *Sendebereich*, *Senderadius* oder *Kommunikationsradius* (s. Abb. 1.1). Alle Knoten, die im Sendebereich eines Knotens S liegen, nennen wir *Nachbarknoten* von S .

Ein *Algorithmus* oder *Protokoll* beschreibt eine schrittweise Lösung eines bestimmten Problems, z.B. der Weiterleitung einer Nachricht von einem Knoten zu einem anderen in einem Netzwerk. Unter Umständen werden zur Durchführung des Algorithmus Daten benötigt; in diesem Beispiel können das die Positionsdaten aller Knoten sein, um die Nachricht in die geographische Richtung des Ziels weiterzuleiten.

Globale Algorithmen arbeiten in einem DSN mit Informationen zum gesamten Netzwerk. In einem Schritt könnte ein einziger Knoten ungeachtet seiner eigenen Position bspw. zur Weiterleitung einer Nachricht eine Kette von Knoten von der Quelle zum Ziel bestimmen, welche die Nachricht weiterleiten sollen. Auf der Basis globaler Netzwerkinformationen kann so die effizienteste Route hinsichtlich Anzahl Hops, benötigte Energie, etc. berechnet werden. Die *Skalierbarkeit* globaler Algorithmen ist jedoch schlecht, denn je größer und dichter ein Netzwerk ist, desto mehr im gesamten Netz verteilte Informationen müssen gesammelt und verarbeitet werden.

In *lokalen* Algorithmen stehen jedem Knoten nur begrenzte Informationen zum Netzwerk zur Verfügung; diese beschränken sich auf die *k-Hop-Nachbarschaftskennntnis* des Knotens: ein Knoten hat nur Informationen zu anderen Knoten, die maximal k Hops (Weiterleitungsschritte) von ihm entfernt sind, wobei k eine Kon-

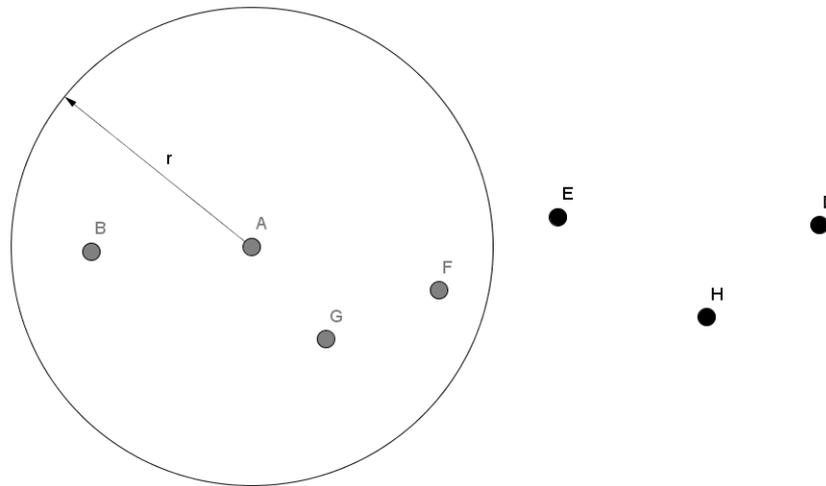


Abbildung 1.1: Knoten A soll eine Nachricht weiterleiten. Er kann Nachrichten maximal bis zum Rand des Kreises mit Radius r senden. r ist sein Sende- bzw. Kommunikationsradius. Alle Knoten innerhalb jenes Kreises sind Nachbarknoten von A (grauen Knoten).

stante ist. Statt Probleme von einem oder wenigen Recheneinheiten lösen zu lassen, wie es in globalen Algorithmen der Fall ist, wird das Problem in lokalen Algorithmen auf Teilprobleme aufgeteilt, welche jeweils von verschiedenen Knoten gelöst werden können. Da Teilprobleme ungeachtet der Größe und Dichte des Netzwerks immer auf Basis der k -Hop-Nachbarschaftskenntnis eines Knotens gelöst werden können, sind lokale Algorithmen besser skalierbar, d.h. sie haben ungeachtet der Netzwerkgröße eine gleichbleibende Effizienz.

Für die Erlangung von Informationen zur k -Hop-Nachbarschaft eines Knotens benutzen viele lokale Algorithmen *Beacons* – von jedem Knoten periodisch gesendete Nachrichten, die Informationen des Senders enthalten. Wie nachfolgend beschrieben hat die Verwendung von Beacons jedoch Nachteile, weswegen etwa seit der Jahrtausendwende sog. lokale *beaconless* Algorithmen entwickelt werden, welche keine Beacons benutzen.

Das Versenden von Beacons kann vor allem in dichten Sensornetzwerken zu *Kollisionen* führen. Eine Kollision findet statt, wenn zwei oder mehr Knoten Nachrichten aussenden, die gleichzeitig von einem Empfänger empfangen werden. Statt zwei separaten Nachrichten erhält er zwei sich überlappende Nachrichten, die er nicht lesen kann.

Das periodische Senden von Nachrichten begrenzt die Bandbreite des Netzwerkes, d.h. die Kommunikationskanäle sind permanent durch das Aussenden von Beacons belastet, wodurch die Datenrate für andere Nachrichtenpakete abnimmt.

Das Resultat sind Verzögerungen beim Senden und Empfangen von Daten – vor allem bei großen Datensätzen.

Der offensichtlichste Vorteil beim Verzicht auf Beacons ist die Energieersparnis. Mit einem beaconless Protokoll agieren Knoten *reaktiv*, d.h. nur wenn es eine Aufgabe für sie gibt. Ansonsten senden sie keine Nachrichten. Mit einem Beacon-behafteten Protokoll müssen Knoten periodisch Nachrichten aussenden, um auf dem aktuellen Stand des Netzwerks zu sein. Dabei ist es egal, ob eine Aufgabe für sie vorliegt oder nicht. Vor allem in drahtlosen Sensornetzwerken ist ein solches Verhalten nicht optimal, denn die Lebenszeit eines Sensorknotens hängt von der ihm zu Verfügung stehenden Energie ab. Ist seine Batterie erschöpft, kann er keine Nachrichten mehr verarbeiten oder aussenden und ist für das Netzwerk nutzlos. Beaconless Protokolle besorgen sich Netzwerkinformationen nach Bedarf und sind dementsprechend auf dem aktuellen Stand des Netzwerks, ohne Beacons zu benutzen. Diese Variante der Informationsbeschaffung ist sogar effizienter als die Beacon-Variante, da in letztgenannter die gesammelten Daten bereits veraltet sein könnten. Vor allem in hoch aktiven mobilen Netzen ändert sich das Netzwerk ständig. Beacon-behaftete Algorithmen stoßen hier auf ein Dilemma: eine höhere periodische Beacon-Rate sorgt für aktuellere Informationen und folglich bessere Ergebnisse z.B. bei der Datenweiterleitung. Die Energie eines Knotens wird so jedoch schneller verbraucht. Sollen die Sensorknoten möglichst energieeffizient arbeiten, muss die Beacon-Rate gesenkt werden, was aber schnell zu veralteten Informationen und somit falschen Berechnungen führt. Beaconless Algorithmen hingegen arbeiten sowohl energieeffizient als auch nur mit aktuellen Netzwerkinformationen.

Die Vorteile von beaconless Algorithmen für drahtlose Sensornetzwerke überwiegen ihre Nachteile. Da die Größe von Sensorknoten im Laufe der Jahre immer weiter abnimmt – das Ziel sind Staubkorn große Knoten [1] – und folglich die Größe von Batterie und Speichermodulen eines jeden Knoten auch minimiert werden müssen, sind energieeffiziente und zustandslose Algorithmen für Sensornetzwerke von höchster Wichtigkeit. Beacon-behaftete Protokolle haben diese Eigenschaften nicht, beaconless Protokolle bieten diese hingegen an.

Trotz ihrer Wichtigkeit im Forschungsgebiet der Rechnernetze existiert nach unserem derzeitigen Kenntnisstand keine vollständige, systematische Übersicht zu beaconless Verfahren. Diese Arbeit beschäftigt sich damit, einen möglichst umfassenden Überblick mit Beschränkung auf lokale beaconless Algorithmen für DSN bereitzustellen. Relevante Artikel werden anhand ihrer Gemeinsamkeiten kategorisiert und ihre Unterschiede werden strukturiert gegenübergestellt. Diese Arbeit dient somit als Nachschlagewerk zum aktuellen Forschungsstand von beaconless Algorithmen und eröffnet Perspektiven auf noch offene Fragestellungen.

Die Gliederung der Arbeit wurde wie folgt gestaltet: Die Suchmethoden und Auswahlkriterien für relevante Artikel werden in Kapitel 3 beschrieben. Die da-

nach folgenden Kapitel 4 bis 7 enthalten Beschreibungen der Vorgehensweisen der recherchierten Algorithmen. Jedes Kapitel steht dabei für eine Kategorie von Algorithmen, welche sich in ihren Verhaltensstrukturen merklich von Algorithmen anderer Kategorien unterscheiden. Innerhalb eines Kapitels werden Algorithmen anhand kleinerer Unterschiede in ihren Verhaltensmustern in weitere Unterkategorien aufgeteilt. Am Ende jedes dieser Kapitel werden Eigenschaften der Protokolle tabellarisch gegenübergestellt. Kapitel 2 erläutert grundlegende Begriffe und Vorgänge, die für das Verständnis zuvor genannter Kapitel notwendig sind. Schließlich wird in Kapitel 8 ein Fazit gezogen. Im Appendix A finden sich Tabellen mit detaillierteren Informationen zu einigen behandelten Protokollen.

Kapitel 2

Grundlagen

In diesem Kapitel werden einige grundlegende Begriffe und Vorgänge erläutert, die für das Verständnis der nachfolgenden Kapitel notwendig sind.

2.1 Weiterleitung

Ein Sensorknoten kann Daten auf verschiedene Arten verschicken: per *Unicast* wird die Nachricht gezielt an einen Nachbarknoten geschickt; per *Multicast* wird die Nachricht gezielt an mehrere Nachbarknoten geschickt; per *Broadcast* wird die Nachricht an alle Nachbarknoten geschickt. Die Begriffe Unicast, Broadcast und Multicast können auch für die Weiterleitung innerhalb eines *Netzwerkes* gemeint sein: in diesem Fall steht Unicast für die Weiterleitung einer Nachricht von einem *Quellknoten* zu einem bestimmten *Zielknoten* aus dem Netzwerk; Multicast steht für die Weiterleitung einer Nachricht von einem Quellknoten zu mehreren bestimmten Zielknoten; Broadcast steht für die Weiterleitung einer Nachricht von einem Quellknoten zu allen Knoten des Netzwerks. Die Nachricht wird in diesen Fällen vom Quell- zum Zielknoten über andere Knoten des Netzwerks geleitet.

Ein Netzwerk von Sensorknoten wird i.d.R. als Graph modelliert. Die Knoten des Graphen stehen dabei für die Sensorknoten; eine Kante zwischen zwei Knoten steht entweder für eine *unidirektionale* oder *bidirektionale* Verbindung zwischen den Knoten oder für eine Weiterleitung einer Nachricht von einem Knoten zum anderen. Die Bedeutung ist i.d.R. aus dem Kontext ersichtlich. Eine unidirektionale Kante bedeutet, dass ein Knoten im Sendebereich eines anderen Knoten liegt, der umgekehrte Fall aber nicht zutrifft; der erste Knoten kann lediglich Daten vom letzten Knoten empfangen. Im Falle einer bidirektionalen Kante liegen beide Knoten in ihren gegenseitigen Sendebereichen, d.h. jeder Knoten kann Daten des jeweils anderen Knoten empfangen. Für das Empfangen von Daten wird in die-

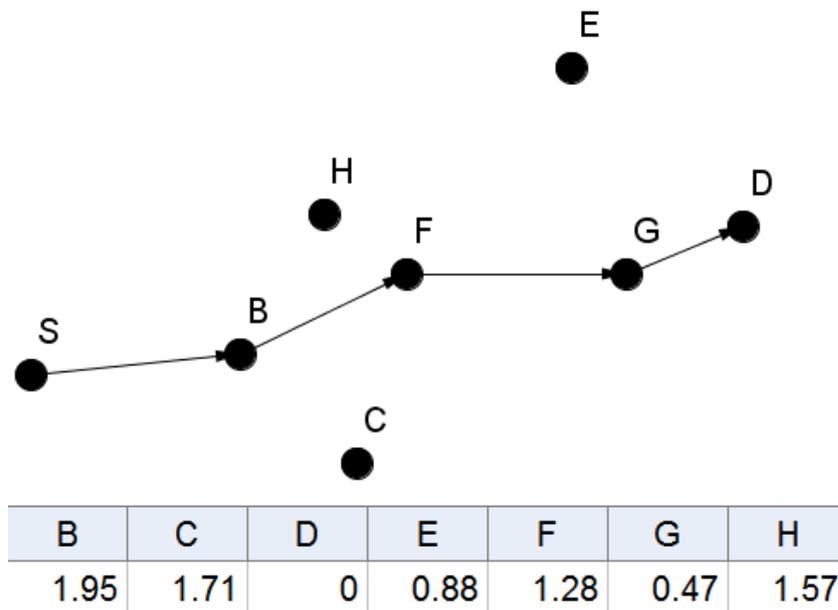


Abbildung 2.1: Geographisches Greedy Routing. Weiterleitung einer Nachricht von Quellknoten S bis zum Zielknoten D . Die Tabelle stellt den euklidischen Abstand eines jeden Knotens zum Zielknoten dar. Der Senderadius jedes Knotens beträgt in diesem Beispiel 1.

ser Arbeit synonym „*hören*“ benutzt. Wenn sich zwei Knoten „gegenseitig hören“ können, besteht eine bidirektionale Verbindung zwischen ihnen.

In der Regel wird in den in dieser Arbeit vorgestellten Protokollen vom *Unit Disk Graph* (UDG) Modell ausgegangen. In diesem Netzgraphenmodell haben alle Knoten den gleichen Senderadius. Dementsprechend gibt es in diesem Modell nur bidirektionale Kanten.

Beim Weiterleiten einer Nachricht über mehrere Knoten bezeichnen wir den Knoten, der aktuell die Nachricht weiterleiten soll, als *Forwarder*. Während es für eine Weiterleitung nur einen Quellknoten gibt, kann es zur selben Zeit mehrere Forwarder geben. Die Weiterleitung einer Nachricht von einem Forwarder an einen anderen Knoten bezeichnen wir als *Hop*; der zuletzt genannte Knoten kann als *nächster Hop* der Weiterleitung bezeichnet werden. Wenn von der „Anzahl Hops von Knoten A zu Knoten B “ gesprochen wird, ist damit die Anzahl von Weiterleitungen (Unicast oder Broadcast) gemeint, die benötigt wurde, um eine Nachricht von Knoten A zu Knoten B zu übertragen.

2.1.1 Geographisches Routing

In geographischen Routing Protokollen kennt ein Knoten, der eine Nachricht versenden will, die Position des Ziels. Jeder Knoten kann seine eigene Position durch ein Lokalisierungssystem, z.B. *GPS*, abfragen. Die Nachricht wird nur auf Basis der Positionsdaten der Knoten in Richtung Ziel weitergeleitet.

Die meisten geographischen Routing Protokolle verwenden ein *Greedy-Recovery-Greedy* Schema: Die Nachricht wird so lange mit *Greedy Routing* (s. Kap. 2.1.2) weitergeleitet, bis sie das Ziel erreicht hat oder kein passender Kandidatenknoten gefunden wird. Geographische Greedy Routing Algorithmen versuchen, das zu versendende Datenpaket in jedem Weiterleitungsschritt näher an den Zielknoten zu bringen. Jeder Hop muss immer eine kleinere euklidische Distanz zum Zielknoten aufweisen als der vorherige Hop. Die Nachricht bewegt sich immer in die geographische Richtung des Ziels (s. Abb. 2.1).

Wird kein Kandidatenknoten gefunden, der die Greedy-Bedingung erfüllt, wird die Nachricht daraufhin mit *Recovery Routing* (s. Kap. 2.1.3) weitergeleitet, bis ein solcher Knoten gefunden wird. Im Recovery Routing werden Knoten ungeachtet der Greedy-Bedingung anhand anderer Eigenschaften als Forwarder gewählt.

Wird während des Recovery Routings ein Knoten gefunden, der die Greedy-Bedingung erfüllt, wird die Nachricht wieder mit Greedy Routing weitergeleitet.

2.1.2 Greedy Routing

Greedy Routing ist eine Art der Weiterleitung von einem Quell- zu einem Zielknoten, in der als nächster Hop nur Knoten in Frage kommen, welche besseren *Fortschritt* bieten als der aktuelle Forwarder. Was unter Fortschritt zu verstehen ist, ist vom jeweiligen Protokoll abhängig. Ist es das Ziel eines Protokolls, in jedem Weiterleitungsschritt die Nachricht räumlich so nah wie möglich in Richtung Ziel zu übertragen (geographisches Routing, s. Kapitel 2.1.1), ist der Fortschritt abhängig von der euklidischen Distanz eines Knotens zum Ziel, d.h. alle Knoten, die geographisch näher am Ziel liegen als der Forwarder, bieten einen besseren Fortschritt oder *Fortschrittswert* als der Forwarder an. Solche Fortschritt bringenden Knoten werden als *Kandidatenknoten* bezeichnet, d.h. sie sind *Kandidaten* für den nächsten Hop.

2.1.3 Recovery Routing

Recovery Routing wird in Unicast Routing Protokollen angewendet, wenn Greedy Routing fehlschlägt. Das passiert, wenn das Datenpaket den Zielknoten noch nicht erreicht hat und der Forwarder keinen geeigneten Nachbarn zur Weiterleitung hat. Eine solche Situation nennt man u.A. *lokales Minimum*, *Deadend* oder *Void*, ein

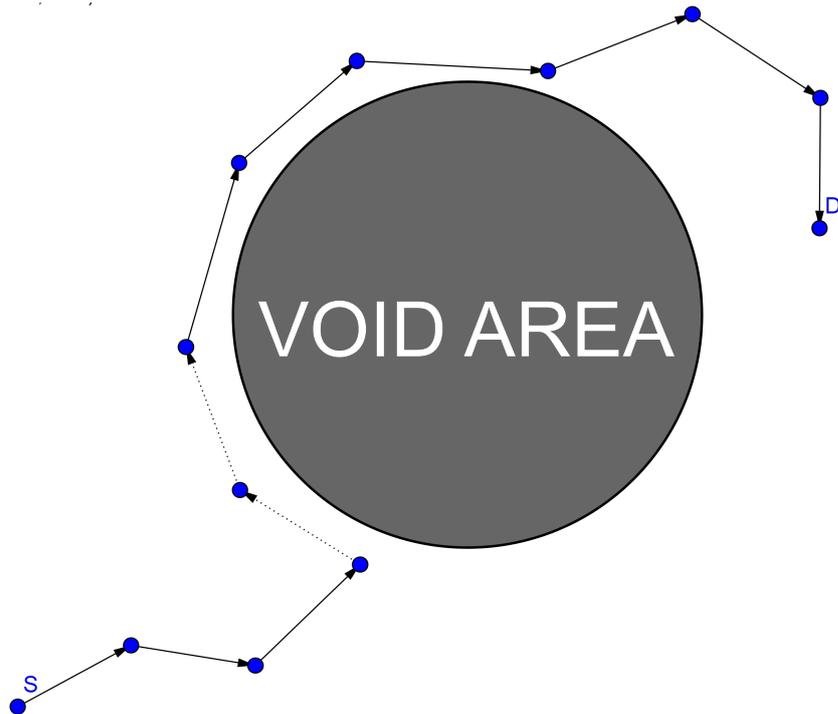


Abbildung 2.2: Knoten S will eine Nachricht an Zielknoten D mit Greedy Routing weiterleiten. Während der Weiterleitung wird ein Knoten erreicht, der keine Nachbarn hat, die näher am Ziel liegen als er selbst. Dieser muss die Nachricht über Recovery Routing (punktierter Pfeile) an Knoten weiterleiten, die die Greedy Routing Bedingung nicht erfüllen. So kann die Nachricht im besten Fall um den Bereich ohne Knoten (*Void Area*) geleitet werden, bis sich wieder Knoten finden, welche Greedy Routing betreiben können (durchgezogene Pfeile).

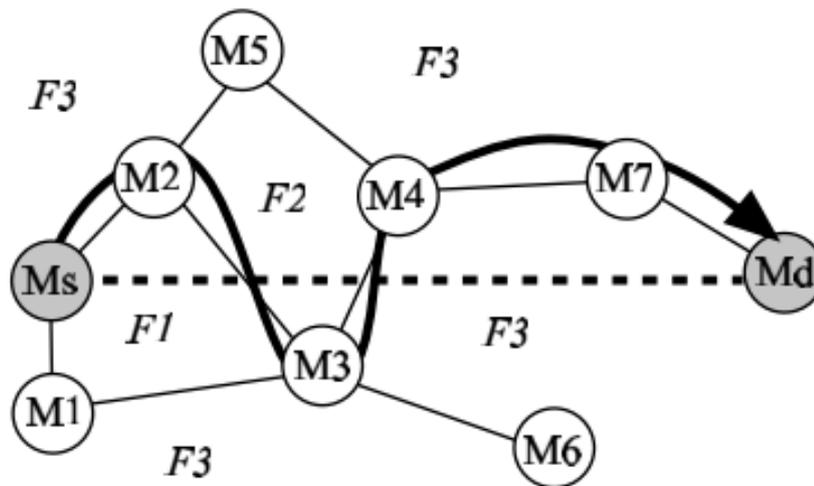


Abbildung 2.3: Face Routing von Quellknoten M_s zu Zielknoten M_d nach rechter-Hand-Regel. Andere Knoten sind als Kreise M_1 bis M_7 dargestellt. Die Bereiche F_1 bis F_3 sind von Kanten eingegrenzte Bereiche, sog. Faces. Die Strecke vom Quell- zum Zielknoten (gestrichelte Linie) schneidet bestimmte Faces; entlang der geschnittenen Faces wird die Nachricht weitergeleitet (durchgezogene Linie/Pfeil). Bild entnommen aus [2].

Gebiet ohne Kandidatenknoten nennt man entsprechend *Void-Region* oder *Void-Area*. Mithilfe von Recovery Routing Verfahren wird versucht, die Nachricht um die Void-Region herum zu leiten, bis man einen Knoten erreicht, der Fortschritt in Richtung Zielknoten bringt (s. Abb. 2.2).

FACE-Routing

FACE-Routing [3] ist ein häufig für Recovery Routing verwendetes Weiterleitungsverfahren. Es sollte allerdings nur auf *planaren* Graphen ausgeführt werden, andernfalls hat es keine *Auslieferungsgarantie*, d.h. die Nachricht erreicht u.U. niemals ihr Ziel. In einem planaren Graphen kreuzen sich keine Kanten. Graphenmodelle, die diese Eigenschaft besitzen, sind u.A. der *Gabriel Graph* und der Graph der *Partiellen Delaunay Triangulation* (PDT) (s. Kap. 4.1). Als *Face* bezeichnet man eine Region eines planaren Graphen, welche von Kanten umschlossen wird und in die keine Kanten hineinragen. Beim FACE-Routing wird eine imaginäre Linie vom aktuellen Forwarder zum Zielknoten gezogen und die Nachricht wird entlang der geschnittenen Faces weitergeleitet (s. Abb. 2.3).

2.2 VANET

Ein „*Vehicular Ad-Hoc Network*“ (*VANET*) ist eine spezialisierte Variante von drahtlosen Sensornetzwerken mit mobilen Knoten (*MANET*). In *VANETs* sind Sensorknoten in Vehikeln wie Personen- oder Lastkraftwagen verbaut. Dementsprechend beschreiben *VANET*-Artikel Protokolle für den Einsatz auf Straßen in urbanen Gebieten mit Kreuzungen oder auf Autobahnen und Straßen ohne Kreuzungen. Durch die hohe Mobilität der Knoten in *VANETs* ist herkömmliches Recovery Routing oft nicht anwendbar. Stattdessen bedienen sich viele Protokolle eines *store-carry-forward*-Ansatzes: findet ein Forwarder keine Kandidatenknoten, speichert er das Datenpaket und wartet einige Zeit, bevor er einen neuen Weiterleitungsversuch startet. In der Zwischenzeit kann sich sein Vehikel in eine günstigere Position bewegt haben, oder die Vehikel anderer Knoten haben sich in seinen Sendebereich bewegt.

Kapitel 3

Methoden und Werkzeuge

In diesem Kapitel wird vorgestellt, wie nach relevanter Literatur gesucht wurde, nach welchen Kriterien ein Artikel für diese Arbeit relevant befunden wurde und mit welchen Werkzeugen (Programmen) die relevante Literatur archiviert und eine Übersicht dazu gestaltet wurde.

3.1 Methode zur Literaturrecherche

3.1.1 Suchprozess

Die Suche nach relevanter Literatur gliederte sich in zwei Phasen: die Datenbanksuche und die „Schneeballsuche“.

Datenbanksuche

Die Datenbanken, die durchsucht wurden, sind *IEEE Xplore*, *ACM Portal*, *SpringerLink* und *ScienceDirect*. In jeder Datenbank wurde nach den Begriffen „beaconless“ (mit Bindestrich) und „beaconless“ gesucht. Falls sich aus dem Titel eines Artikels nicht direkt erkennen ließ, dass der Artikel eindeutig nicht relevant ist, wurde die Kurzfassung (Abstract) gelesen. Ließ sich aus der Kurzfassung auch nicht erkennen, dass der Artikel eindeutig nicht relevant ist, wurde der Artikel überflogen und bei kritischen Fällen ggf. ein zweites Peer-Review eines Experten aus dem Feld hinzugezogen.

Es sollte hier noch einmal angemerkt werden, dass nicht nach relevanten Artikeln gesucht wurde, sondern nicht relevante Artikel aussortiert wurden. Würden nur Artikel als relevant befunden werden, welche anhand des Titels oder der Kurzfassung eindeutig als beaconless identifiziert werden können, wäre die Literaturliste u.U. nicht so umfangreich geworden.

Schneeballsuche

Nach Abschluss der Datenbanksuche wurden alle Literaturlisten der für relevant befundenen Artikel nach dem selben Schema wie in der Datenbanksuche untersucht. Auf diese Weise gefundene, relevante Artikel wurden wiederum auf die selbe Weise untersucht. Zum Suchen der Artikel im Internet wurde die Suchmaschine *Google Scholar* benutzt.

Bei mehreren Versionen eines Artikels wurde nur bei der für wichtiger befundenen Version die Schneeballsuche durchgeführt. Mehr dazu im Kapitel 3.1.2.

3.1.2 Auswahlkriterien

Ziel der Literaturrecherche war es, jeden Artikel aus dem Themengebiet der drahtlosen Netzwerke zu finden, der einen lokalen beaconless Algorithmus beschreibt. Solche Algorithmen haben nach unserer Definition die Eigenschaft, dass keine periodischen „Beacon“ oder „Hello“-Nachrichten von Sensorknoten ausgesendet werden; mit „Beaconing“ beschreiben wir das Aussenden einer Nachricht durch einen Knoten, dem jeder Empfänger der Nachricht bedingungslos eine Antwort schickt. Beaconing sollte nie ein Kernaspekt eines beaconless Protokolls sein, sondern sich z.B. auf Recovery Routing beschränken (s. Kapitel 5.2). Außerdem werden Aufgaben ohne globales Wissen über das Netzwerk gelöst, sondern nur mit lokalen Informationen, die ein Knoten entweder selbst beschaffen oder von einem anderen Knoten, der diese Informationen lokal beschafft hat, erhalten kann.

Wir haben aus der Suche Verfahren ausgeschlossen, die Beacons zwangsläufig zur Ausführung ihres Hauptprotokolls benötigen; zusätzlich wurden Verfahren ausgeschlossen, die zu irgendeinem Zeitpunkt während ihrer Ausführung das gesamte Netz *fluten* müssen, d.h. eine Nachricht ausgesendet wird, die von jedem Knoten im Netzwerk erhalten werden muss. Dazu zählen auch einige beaconless Protokolle, die in der initialen Ausführungsrunde Informationen über das gesamte Netzwerk sammeln müssen, z.B. um den kürzesten Weg von einem Quell- zu einem Zielknoten zu finden. Ein solches Vorgehen verstößt gegen unser Kriterium der Lokalität. Auch beaconless Broadcast-Protokolle, deren Ziel es ist, das Netzwerk im Hinblick auf bestimmte Aspekte möglichst effizient zu fluten, haben wir nicht weiter beachtet. Dazu zählen unter anderem die Verfahren aus [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] und [14].

Wurden mehrere Versionen zu einem Verfahren gefunden, z.B. ein Artikel aus einem Journal und ein Artikel aus einer Konferenz, wurde immer der bekanntere Artikel für diese Arbeit als relevant befunden. Für gewöhnlich wurde immer die Journal-Version bevorzugt (falls vorhanden), in seltenen Fällen, bspw. IGF [15], wurde die Version für relevant befunden, die am häufigsten in der Literatur zitiert

wurde – im Falle von IGF wurde so der Technical Report der Journal-Version vorgezogen.

3.1.3 Werkzeuge

Relevante Artikel wurden direkt zur einfachen Bearbeitung (Vermerke, Notizen, Textmarkierungen) in *Mendeley*¹ aufgenommen.

Während der Suche wurden nicht relevante Artikel in einer einfachen Textdatei notiert, um mehrere Reviews des selben Artikels z.B. aus verschiedenen Datenbanken zu vermeiden.

Um einen einfachen Überblick über die kategorisierten Artikel, die jeweilige Veröffentlichungsart und den Bearbeitungsstatus zu haben, wurde der Graph Editor *yEd*² benutzt (s. Abb. 3.1).

3.2 Methode zur Kategorisierung der Verfahren

Relevante Algorithmen wurden in dieser Arbeit auf Basis ihrer Einsatzgebiete (z.B. Algorithmen für VANETs, s. Kap. 7) und/oder ihrer Verwendung bestimmter grundlegender Verhaltensmuster (z.B. aggressive- und non-aggressive contention in Kap. 5) in Kategorien eingeordnet.

¹Mendeley Desktop, Ver. 1.11, Mendeley Ltd. Verfügbar auf: <http://www.mendeley.com/download-mendeley-desktop/> Letzter Zugriff: 26.09.2014, 16:58 Uhr

²yEd, Ver. 3.11.1, yWorks GmbH. Verfügbar auf: <http://www.yworks.com/de/downloads.html#yEd> Letzter Zugriff: 26.09.2014, 17:02 Uhr



Abbildung 3.1: Artikel in yED (Ausschnitt). Relevante Artikel wurden in Form eines Graphen dargestellt und verwaltet. Rechtecke sind Kategorien und Unterkategorien von Artikeln, Artikel selbst sind Kinderknoten von Kategorien. Sie werden als Ellipsen dargestellt. Die Art eines Artikels (Journal, Proceedings, Technical Report, etc.) wird farblich als Hintergrund der Ellipse kodiert; zusätzliche Informationen und der Bearbeitungsstand eines Artikels werden als Kinderknoten von Artikeln in Form von Rechtecken mit abgeknickter oberen rechten Ecke dargestellt.

Kapitel 4

Topologiekontrolle

Topologiekontrolle bezeichnet eine Technik zur Änderung von *Topologien* von Netzwerken (die den Netzwerken zugrundeliegenden Netzwerkgraphen), um z.B. Algorithmen hinsichtlich verschiedener Aspekte effizienter im Netzwerk ausführen zu können. Für gewöhnlich werden Netzwerkgraphen, auf denen Topologiekontrolle ausgeführt wird, *ausgedünnt*, d.h. die Kantenmenge wird reduziert (s. Abb. 4.1). In einem drahtlosen Sensornetzwerk bedeutet das beispielsweise, dass Knoten Daten nicht immer mit voller Sendeleistung per Broadcast an alle Nachbarknoten aussenden, sondern je nach benutzter Topologie Listen ihrer Nachbarn im ausgedünnten Netzwerkgraphen führen und jedem dieser Nachbarn die Nachricht per Unicast senden. Daraus resultiert eine Reduzierung des Energieverbrauchs und der Interferenzen innerhalb eines Netzwerks, weil Nachrichten nur noch mit der nötigen Sendeleistung verschickt werden müssen, um bestimmte Nachbarn zu erreichen und die dadurch verringerten Senderadien die Wahrscheinlichkeit verringern, dass sich Sendebereiche überlappen.

4.1 Graphenmodelle

Hier werden verschiedene Graphenmodelle vorgestellt, die für Topologiekontrolle benutzt werden können.

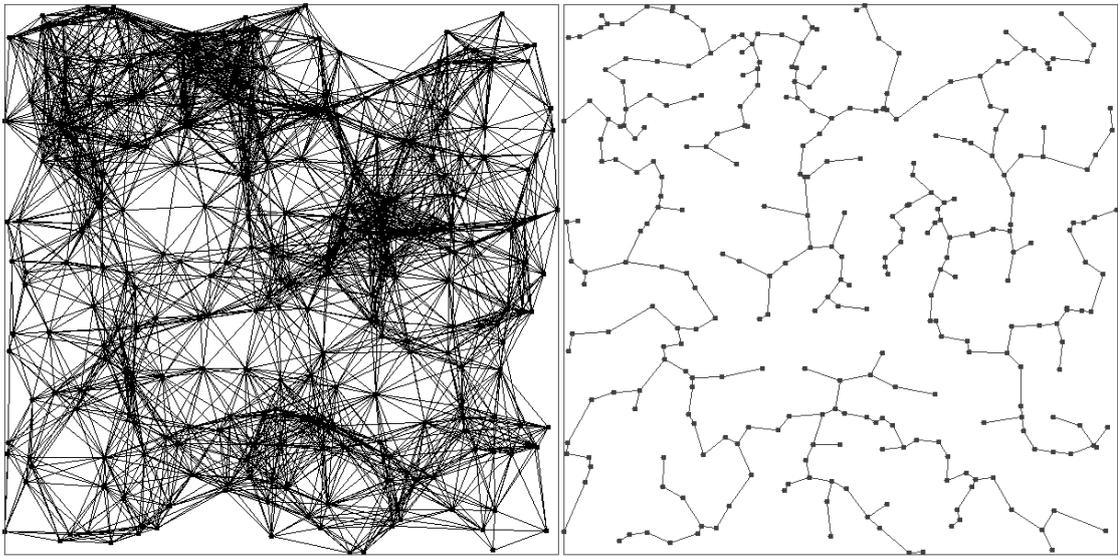


Abbildung 4.1: Links¹: Beispielnetzwerk ohne Anwendung von Topologiekontrolle. Rechts²: Das gleiche Netzwerk mit reduzierter Topologie durch Anwendung eines minimalen Spannbaum-Algorithmus.

4.1.1 Unit Disk Graph

Alle Knoten haben einen uniformen Senderadius der Länge einer *Unit*. Liegt ein Knoten im Sendebereich eines anderen Knoten, haben sie eine gemeinsame Kante.

4.1.2 Graph der Partiellen Delaunay Triangulation (PDT)

Eine Teilmenge des Unit Disk Graph. Enthält alle Kanten des Gabriel Graphs (s. Kapitel 4.1.3) sowie zusätzlich jede Kante zwischen zwei Knoten u und v , falls der den Winkel $\angle uvw$ maximierende Knoten w so liegt, dass der Kreis durch alle drei Knoten keine anderen Knoten enthält und außerdem vollständig innerhalb der *Unit Disk* von u liegt (s. Abb 4.2). Die Unit Disk eines Knotens ist der Sendebereich des Knotens im Unit Disk Graph Modell.

Im Gegensatz zum Gabriel Graph ist der Graph der PDT ein *euklidischer Spanner* mit einer konstanten *Spanning Ratio*. Das bedeutet, dass die Länge eines Pfads von einem Knoten A zu einem anderen Knoten B nach der Vereinfachung des Graphen gemäß PDT nur um einen konstanten Faktor von der Pfadlänge

¹Quelle: http://upload.wikimedia.org/wikipedia/commons/8/82/Capture_all.PNG, letzter Zugriff: 29.09.2014, 21:45 Uhr

²Quelle: http://upload.wikimedia.org/wikipedia/commons/c/c3/Capture_mst.PNG, letzter Zugriff: 29.09.2014, 21:48 Uhr

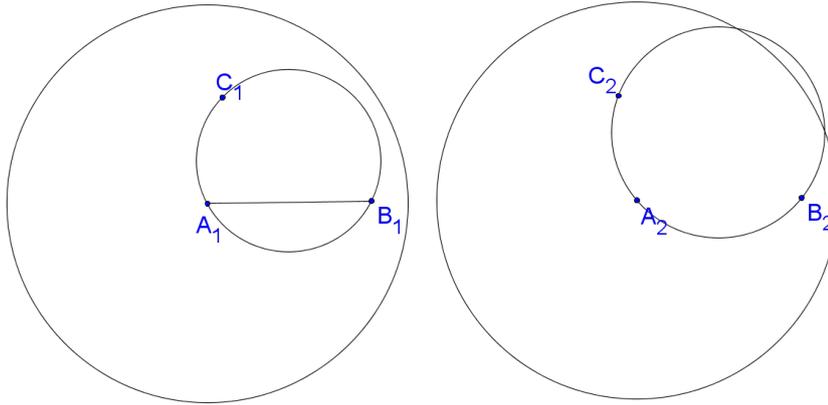


Abbildung 4.2: Vereinfachtes Beispiel zur Darstellung der PDT Graph-Bedingung. Links: Der durch A_1 , B_1 und C_1 aufgespannte Kreis enthält keine anderen Knoten und befindet sich vollständig innerhalb der Unit Disk von A_1 ; folglich existiert eine Kante zwischen A_1 und B_1 mit winkelmaximierendem Knoten C_1 . Rechts: Der gleiche Kreis wie im linken Bild ragt aus der Unit Disk von A_2 hinaus, was eine PDT-Regel verletzt. Es existiert keine Kante zwischen Knoten A_2 und B_2 mit winkelmaximierendem Knoten C_2 .

des kürzesten Pfades im Ursprungsgraphen entspricht. Dies ist eine gewünschte Eigenschaft, die der Gabriel Graph nicht bietet.

4.1.3 Gabriel Graph

Eine Teilmenge der Partiellen Delaunay Triangulation. Zwei Knoten haben genau dann eine gemeinsame Kante, wenn der Kreis, dessen Durchmesser die Strecke der beiden Knoten ist und der durch diese Knoten läuft, keine anderen Knoten enthält (s. Abb. 4.3). Der Gabriel Graph hat eine Spanning Ratio von $\Theta(\sqrt{n})$ [16].

4.1.4 Relative Neighborhood Graph

Eine Teilmenge des Gabriel Graph. Zwei Knoten A und B haben nur dann eine gemeinsame Kante, wenn es keinen anderen Knoten C gibt, der eine geringere Distanz zu A und zu B hat, als die Distanz von A zu B . Dieser Graph hat eine konstante Spanning Ratio ($O(1)$) [16].

4.2 Verfahren

Der Recovery-Modus von GDBF [17] (s. auch Kapitel 5.1.2 und Tabelle A.19) ist ein Topologiekontrollverfahren, das alle Nachbarn des Forwarders im Gabriel

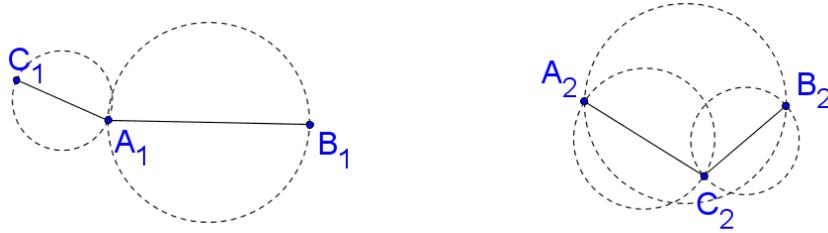


Abbildung 4.3: Links: Der durch A_1 und C_1 aufgespannte Gabrielkreis (gestrichelt) enthält keine anderen Knoten, darum sind die genannten Knoten Nachbarn im Gabriel Graph. Das Gleiche gilt für die Knoten A_1 und B_1 . Rechts: Der durch A_2 und B_2 aufgespannte Gabrielkreis enthält den Knoten C_2 , darum sind A_2 und B_2 keine Nachbarn im Gabriel Graph. Ihre jeweiligen, mit C_2 aufgespannten, Gabrielkreise hingegen erfüllen die Gabriel-Graph Bedingung.

Graph-Modell (s. Kap. 4.1) berechnet. Die Nachbarknoten des Forwarders setzen ihre Zähler abhängig von ihrer Distanz zu diesem, wobei für eine kürzere Distanz entsprechend ein kürzerer Zähler berechnet wird. Hört ein Kandidatenknoten X eine Antwort eines Knotens Y für Forwarder S , bricht er seinen Zähler ab, falls Y im Gabrielkreis von S und X liegt.

Das Verfahren BFP [18] berechnet die Kanten eines Knotens zu seinen Nachbarknoten für ein planares Subgraphenmodell. Das Subgraphenmodell muss mindestens alle Kanten des Relative Neighborhood Graph und höchstens alle Kanten des Gabriel Graph enthalten. Das Verfahren wird in zwei Phasen ausgeführt: In der ersten Phase sendet ein Knoten v eine Nachricht per Broadcast an alle Nachbarknoten n aus; diese berechnen abhängig von ihrer Distanz zu v einen Verzögerungswert und schicken nach entsprechender Wartezeit eine Antwort. Andere Nachbarknoten, die diese Nachricht hören und selbst noch nicht geantwortet haben, verwerfen ihre geplante Antwort. Falls die gehörte Nachricht von einem Knoten n' stammt und die Kante (v, n') eine Regel des gewählten Subgraphenmodells verletzt, melden Knoten, die die Antwort von n' gehört haben und den Regelbruch erkannt haben, diesen Knoten dem Knoten v . Dieses Melden fälschlicherweise antwortender Knoten geschieht in einer zweiten Phase, einer sog. *Protestphase*. Nur Knoten, welche in der ersten Phase geantwortet haben und die in der zweiten Phase nicht gemeldet wurden, sind Nachbarknoten von v im gewählten Graphenmodell.

Das Reactive PDT [19] Verfahren berechnet reaktiv die Kanten eines Knotens im PDT Graphmodell. Wie schon in BFP sendet ein Knoten u eine Anfrage-Nachricht per Broadcast und wartet für eine bestimmte Zeit auf Antworten seiner Nachbarn; diese warten eine bestimmte, von ihrer Distanz zu u abhängige Zeit lang, bevor sie eine Antwort schicken. Hört ein Knoten v eine Antwort eines anderen Knotens z , merkt er sich diesen Knoten als Nachbar und prüft, ob auf Basis aller von ihm gehörten Nachbarknoten eine PDT-Regel verletzt wird. Sollte dies der

Fall sein, schickt v ggf. keine Antwort an u ; sonst verlängert v ggf. seine Wartezeit für das Senden der Antwort. Alle antwortenden Knoten sind Nachbarknoten von u im Graphen der Partiellen Delaunay Triangulation.

Die drei untersuchten Verfahren werden in Tabelle 4.1 verglichen. Darin stellt die Spalte zur *k-Lokalität* dar, aus welcher k-Hop-Nachbarschaft ein Knoten Informationen für ein Ausführen des Verfahrens beziehen muss; die *Komplexität im schlechtesten Fall* bezieht sich auf die Nachrichtenkomplexität der Verfahren; in der Spalte *Metrik* wird beschrieben, nach welcher Methode in einem Verfahren die Zähler der Kandidatenknoten berechnet werden; die Spalte *TK-Struktur* steht jeweils für den Subgraphen, dessen Kanten ein Verfahren berechnet; *Protestphase* und *UDG* veranschaulichen, welche Verfahren Protestphasen benutzen bzw. auf dem Unit Disk Graph Modell basieren.

Table 4.1: Übersicht Topologiekontrolle-Protokolle

Name	Ref.	k-Lokalität	Komplexität im schlechte- sten Fall	Metrik	TK-Struktur	Protestphase	UDG	Jahr
GDBF	[17]	1	$O(n)$	D	GG	✓	✓	2006
BFP (GG)	[18]	1	$O(n)$	D	GG	✓	✓	2010
BFP (CNG)			$O(N_{CNG}(u))$		CNG			
BFP (RNG)			$O(N_{RNG}(u))$		RNG			
Reactive	[19]	1	$O(N_{PDT}(u))$	DW	PDT		✓	2013
PDT								

Legende

Metrik: D = Distanz, W = Winkel $\angle uvz$ mit $u = \text{Forwarder}$, z und $v = \text{Kandidatenknoten}$
 Komplexität: $n = \text{alle Nachbar-knoten}$, $u = \text{Forwarder}$, $|N_G(u)| = \text{Anzahl der Nachbar-knoten von Knoten } u \text{ im Graphmodell } G$
 TK-Struktur: GG = Gabriel Graph, CNG = Circular Neighborhood Graph, RNG = Relative Neighborhood Graph, PDT = Partielle
 Delaunay Triangulation

Kapitel 5

Unicast Routing

Unicast Routing bezeichnet das Weiterleiten einer Nachricht von einem Quellknoten über andere Knoten zu einem Zielknoten.

Um den optimalen nächsten Hop bestimmen zu können, verwenden beaconless Schemata Verzögerungsmechanismen. Typischerweise wird eine Anfrage vom aktuellen Forwarder an so viele Nachbarn wie möglich per Broadcast versandt, woraufhin sich die Empfänger der Nachricht um den Platz als nächster Hop bewerben. Je nachdem, wie gut ein Knoten im Sinne des verwendeten Algorithmus als nächster Hop geeignet ist, berechnet er einen dazu passenden Verzögerungswert und startet einen Zähler mit diesem Wert. Nach Ablauf des Zählers sendet der Knoten eine Antwort. Der Knoten mit der besten Weiterleitungsposition berechnet den kürzesten Verzögerungswert unter allen Bewerbern und unterdrückt mit seiner Antwort andere Knoten, deren Zähler noch laufen. Die Formel zur Berechnung des Verzögerungswerts kann unterschiedliche Aspekte berücksichtigen, von euklidischen Abständen von Knoten bis hin zur verbleibenden Energie eines Knotens. Je nach Ziel eines Algorithmus kann sich dessen Verzögerungsformel erheblich von Formeln anderer Unicast Routing Algorithmen unterscheiden.

Einige Protokolle verfügen über keine strikte Trennung von MAC-Schicht und Vermittlungsschicht. Auf solchen Protokollen aufbauende Routing Algorithmen können für gewöhnlich nur auf einem speziell für sie ausgelegten MAC-Protokoll ausgeführt werden. Solche Protokolle werden als MAC-Cross-Layer Protokolle bezeichnet.

Beaconless Unicast Routing Protokolle sind in den meisten Fällen *source-to-sink* basiert, d.h. eine Nachricht wird von einer Datenquelle, z.B. einem mobilen Sensorknoten, zu einer Senke, z.B. einem statischen Knoten, weitergeleitet; sie finden aber u.A. auch in Anfrage-Verfahren Anwendung, in denen ein Knoten eine bestimmte Information aus dem Netzwerk erfragt. In CERS [20] beispielsweise wird das Netzwerk in Ringe unterteilt und über den Median eines jeden Rings

wird beaconless eine Anfrage-Nachricht weitergegeben, bis ein gesuchter Knoten gefunden oder jeder Ring durchsucht wurde.

Beaconless Unicast Routing Protokolle lassen sich für gewöhnlich entweder dem Greedy- oder Recovery Routing zuordnen.

5.1 Greedy Routing

Beaconless Greedy Routing lässt sich in zwei große Kategorien einteilen: Aggressive- und Non-Aggressive Contention.

5.1.1 Aggressive contention

Ein typischer Weiterleitungsschritt eines Greedy Routing Protokolls, das Aggressive Contention verwendet, wird in der folgenden Liste beschrieben:

1. Der Forwarder sendet das weiterzuleitende Datenpaket direkt über Broadcast an alle Nachbarn.
2. Jeder Empfänger der Nachricht berechnet und startet einen Zähler. Welche Faktoren bei der Berechnung dieses Zählers eine Rolle spielen, hängt vom Ziel des jeweiligen Protokolls ab. Generell gilt jedoch, je kürzer der Zähler, desto besser ist dieser Knoten als nächster Hop geeignet.
3. Nach Ablauf seines Zählers leitet ein Knoten das Datenpaket als Broadcast weiter.
4. Knoten, die ein Datenpaket erhalten, während ihr Zähler noch läuft, brechen diesen ab und senden nicht.

Im Aggressive Contention Schema kann es passieren, dass zwei Knoten, die sich nicht gegenseitig hören können, das selbe Paket weiterleiten. Darum müssen die Weiterleitungsbereiche festgelegt werden. Das sind räumliche Gebiete, in denen Kandidaten zur Weiterleitung gesucht werden. Knoten außerhalb eines solchen Gebiets scheiden als Kandidaten für den nächsten Hop aus. Weiterleitungsbereiche werden häufig so gesetzt, dass alle Knoten in diesem Bereich direkt miteinander kommunizieren können. Diese Begrenzung kann allerdings dazu führen, dass Knoten außerhalb des Weiterleitungsbereiches übersehen werden, die eine bessere Position zur Weiterleitung des Datenpakets haben; folglich wird die Paketauslieferungsrates – die Wahrscheinlichkeit, dass ein Datenpaket sein Ziel erreicht – durch die Begrenzung des Weiterleitungsbereiches limitiert [21]. Einen Kompromiss zwischen diesen beiden Faktoren zu finden, ist ein Kernproblem solcher Protokolle.

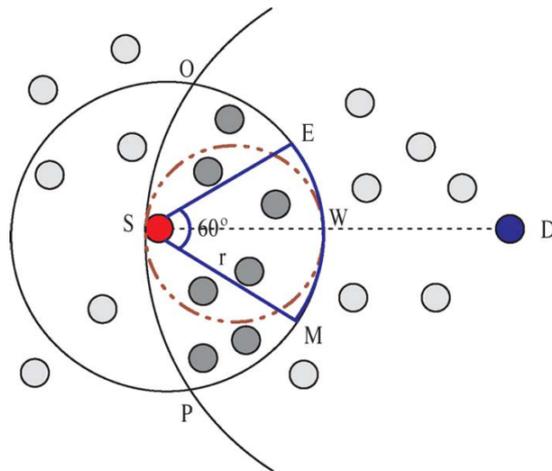


Abbildung 5.1: Drei typische Weiterleitungsbereiche für zählerbasiertes, geographisches Routing. Der Forwarder S sucht im Weiterleitungsbereich einen Kandidatenknoten zur Weiterleitung einer Nachricht in Richtung des Zielknotens D . Dargestellt sind ein kreisförmiger Weiterleitungsbereich (gestrichelt-punktierte Linie), ein Weiterleitungsbereich in Form eines Reuleaux Dreiecks mit einem Öffnungswinkel von 60° (die Strecken $|SE|$, $|SM|$ und die Kurve EWM) und der *Greedy*-Bereich, der alle Knoten enthält, die noch im Sendebereich von S liegen und sich näher am Zielknoten D befinden als S selbst (die Kurven von S nach O bzw. P und begrenzt durch den Sendebereich von S). Bild entnommen aus [22].

Der Artikel aus [22] analysiert die Eigenschaften dreier typischer Weiterleitungsbereiche (s. Abb. 5.1), d.h. wie sich die Wahl eines bestimmten Bereiches in einem zählerbasierten, geographischen Routingprotokoll auf Metriken, wie beispielsweise der maximalen Anzahl von Hops bis zum Ziel, auswirkt.

Die Algorithmen BLR [23], CBGR [24], BGR [25], EABGR [26], ABC [27] und CNB-GEDIR [28] versuchen, die Anzahl der Hops vom Quellknoten bis zum Zielknoten möglichst gering zu halten. Dabei bietet BLR [23] drei verschiedene Formeln zur Berechnung des Verzögerungswerts an, von denen nur eine (1) ausschließlich auf die Minimierung der Hopanzahl ausgelegt ist (s. Tab. A.1).

Eine Besonderheit des BLR-Verfahrens aus [23] ist das sogenannte „Reactive Local Routing“. Es tritt in der Nähe des Zielknotens in Kraft und flutet ein Gebiet um diesen Knoten mit der Datennachricht, um mögliche Ungenauigkeiten in den Positionsdaten des Zielknotens auszugleichen. Die Formel (2) (s. Tab. A.1) zur Berechnung des Verzögerungswerts aus BLR [23] wählt immer den Knoten mit dem niedrigsten Fortschritt aus, was den totalen Energieverbrauch beim Routing von Quelle zu Ziel verringern soll. Die dritte Formel (3) ist komplexer als die anderen beiden Formeln und beachtet neben der Nähe zum Zielknoten auch, ob sich ein Knoten auch nah an der Strecke vom Forwarder zum Ziel befindet.

CBGR [24] beachtet bei der Weiterleitung, wie weit der nächste Hop das Paket senden kann. Der Kandidatenknoten, dessen Rand des maximalen Senderadius unter allen Kandidatenknoten am nächsten zum Zielknoten ist, wird der nächste Forwarder. Im „Optional Mode“ merkt sich ein Knoten, welchen Nachbarknoten er in der Vergangenheit als nächsten Hop für einen bestimmten Zielknoten gewählt hat. Er leitet weitere Nachrichten für den selben Zielknoten an den selben Nachbarknoten weiter, ohne einen Wettbewerb zu starten. Das spart Energie und Zeit. Um Knotenmobilität in Betracht zu ziehen, wechselt ein Knoten nach einem bestimmten Zeitintervall wieder in den üblichen „Basic Mode“.

Um zu verhindern, dass mehrere Kandidatenknoten identische Verzögerungswerte berechnen und gleichzeitig ein Paket weiterleiten, wird in BGR [25] in den Paketheader die Anzahl bereits zurückgelegter Hops geschrieben. Üblicherweise würde ein Knoten das erste Paket empfangen und dabei einen Zähler starten, der direkt durch den Empfang des zweiten Pakets abgebrochen werden würde. Es würde so keine Weiterleitung mehr stattfinden. Hier greift die in [25] vorgeschlagene „Avoidance of Simultaneous Forwarding“ ein. Dabei vergleicht ein Empfängerknoten die Hop-Anzahl zweier direkt nacheinander ankommender Pakete und bricht bei gleicher Hop-Anzahl seinen Timer nicht ab.

Der Algorithmus aus EABGR [26] beachtet bei der Wahl des nächsten Hops auch die verbleibende Energie eines Knotens. Dabei werden Knoten mit mehr verbleibender Energie und kleinerem Abstand zum Zielknoten bevorzugt. Es ist auf dreidimensionale Sensornetzwerke ausgelegt.

In ABC [27] werden Knoten abhängig von ihrem Abstand zum Zielknoten Zonen zugeordnet. Jeder Zone werden bestimmte Zeitslots zum Antworten zugeteilt. Die Zählerberechnung basiert nur auf der Distanz zum Ziel, d.h. Knoten, die näher am Zielknoten liegen, antworten früher.

Das CNB-GEDIR-Verfahren aus [28] ist im ursprünglichen Sinne darauf ausgelegt worden, eine Route von einem Quell- zu einem Zielknoten zu finden. Dazu benutzt es ein Aggressive Contention-Schema, in dem es mit Kontrollnachrichten in jedem Schritt den Nachbarknoten findet, der am nächsten zum Zielknoten liegt. Erweitert man diese Kontrollnachricht um das Datenpaket, erhält man ein typisches beaconless Routingverfahren.

BMR [29] wählt den nächsten Hop zählerbasiert anhand der verbleibenden Energie eines Knotens und dessen ZonenID. Dabei haben Knoten, die sich näher am Ziel befinden, eine kleinere ZonenID. Die Gewichtung der beiden Kriterien ist implementationsabhängig. Dieses Verfahren funktioniert nur in Netzen mit statischen Knoten. Die Basisstation, d.h. der Zielknoten, sendet zu Beginn des Protokolls Nachrichten mit unterschiedlichen Übertragungsstärken aus. Je nach Übertragungsstärke wird ein unterschiedlich großes Gebiet um die Basisstation erreicht. Empfänger der Nachricht setzen ihre ZonenID auf die in der Nachricht

enthaltenen ID; wird mehr als eine Nachricht empfangen (dies ist z.B. der Fall bei Knoten, die sich sehr nah an der Basisstation befinden), wird immer die kleinere ID übernommen. Je niedriger die gesendete ID ist, desto schwächer war die Übertragungsstärke und desto näher befindet sich die Zone mit der entsprechenden ID an der Basisstation. Eine Nachricht eines Knotens an die Basisstation wird von einer hohen Zone zu einer niedrigeren Zone in Richtung Ziel weitergeleitet. Dieses Verfahren unterscheidet sich erheblich von den anderen vorgestellten Verfahren in der Hinsicht, dass ein potentieller Forwarder im Sendebereich des Zielknotens liegen muss.

Im Artikel von Mishra und Gore [30] wird kein eigenständiges Routingverfahren vorgestellt, sondern eine Methode zur Bestimmung des nächsten Forwarders. Diese soll energieschonend sein und ist an drei-dimensionale Netze angepasst; neben der verbleibenden Energie eines Knotens spielt auch seine Nähe zum maximalen Senderadius des Forwarders und zur Strecke zwischen Forwarder und Ziel eine Rolle. Da explizit verschiedene Weiterleitungsbereiche untersucht werden, bezieht sich dieser Artikel auf Aggressive Contention-Verfahren.

Das Verfahren aus I-PBBLR [31] ist kein Routing Algorithmus, sondern eine Erweiterung für bereits existierende Verfahren. Obwohl seine Anwendung im Artikel nur für Aggressive Contention-Schemata beschrieben wird, ist er theoretisch auf alle Routing Verfahren mit einem Verzögerungsmechanismus anwendbar. Die Kernidee von I-PBBLR ist es, den implementationsabhängigen Fortschrittswert eines jeden Knoten mit dem Cosinus des Winkels der Strecke von Forwarder zu Kandidatenknoten und der Strecke von Forwarder zu Zielknoten (s. Abb. 5.2) zu multiplizieren. Durch die Erweiterung der Fortschrittswertberechnung sollen Routing Protokolle eine bessere Leistung in Bezug auf Distanz und Richtung des weitergeleiteten Pakets erzielen.

Der BLR-E Algorithmus [32] ist eine Erweiterung zu BLR [23]. Er ändert die Formel zur Verzögerungsberechnung so, dass Knoten bevorzugt werden, die in Gebieten mit höherer Knotendichte liegen. Diese Veränderung verringert die Wahrscheinlichkeit, an ein lokales Minimum zu gelangen. Daraus resultiert laut den Autoren eine Verbesserung der globalen Weiterleitungseffizienz.

Auch der Algorithmus aus DAF [33] ist eine Erweiterung zu BLR [23]. Er versucht, die totale Verzögerungszeit von BLR zu verbessern. Im Gegensatz zur originalen Zählerberechnung, bei der zur Wahl eines Forwarders der Abstand des aktuellen Forwarders zur Projektion der Position des Kandidatenknotens auf die Strecke zwischen Forwarder und Zielknoten betrachtet wird, betrachtet DAF den Abstand eines Kandidatenknotens zur Strecke zwischen Forwarder und Zielknoten. Daraus ergibt sich laut den Autoren eine niedrigere Verzögerungszeit pro Hop; die Hopanzahl kann sich aber erhöhen.

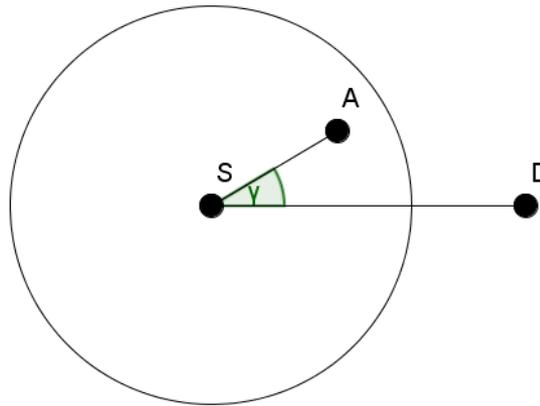


Abbildung 5.2: Winkel γ zwischen Kandidatenknoten A, Forwarder S und Zielknoten D.

Obwohl die Autoren von DBO [34] ihren Algorithmus als Broadcast-Protokoll bezeichnen, weist er Merkmale eines Routing Verfahrens auf. Das Gebiet zwischen einem Quell- und einem Zielknoten wird dabei in „Hop Zones“ aufgeteilt (ähnlich zu BMR [29]) und die Nachricht wird von einer Hop Zone an die jeweils niedrigere, dem Zielknoten näher gelegene Zone, weitergeleitet. Im besten Fall findet in jeder Hop Zone nur eine Weiterleitung statt, Paketduplikationen können aber auftreten. Der nächste Forwarder wird „zufällig“ gewählt; tatsächlich bewirbt sich jeder Paketempfänger um den Sendekanal und der Gewinner leitet die Nachricht weiter.

Im Gegensatz zu den meisten anderen Verfahren, bietet RBRR [35] zeitkritisches Routing an. Hier soll das Datenpaket bis zu einem bestimmten Zeitpunkt das Ziel erreicht haben. Das Protokoll versucht entsprechend das Paket so zeitlich schnell wie möglich von Quelle zu Ziel weiterzuleiten. Dazu berechnet jeder Knoten auf Basis seiner Distanz zu einem Kandidatenknoten, der benötigten Zeit zum Senden und Empfangen einer Nachricht sowie des gewünschten spätesten Auslieferungszeitpunkts einen „speed“-Wert. In jedem Weiterleitungsschritt wird ein Kandidatenknoten als nächster Hop ausgewählt, dessen Wert mindestens einem vom Quellknoten berechneten „speed“-Wert entspricht und zugleich möglichst nah am Zielknoten liegt. Kann kein solcher Knoten gefunden werden, wird die Nachricht das Ziel nicht rechtzeitig erreichen.

MRR [36] ist ein Multipath Routing Protokoll. Ein Forwarder bestimmt zu Beginn des Verfahrens einen Weiterleitungsbereich in Richtung Zielknoten. Der Weiterleitungsbereich ähnelt einem Rechteck (s. Abb 5.3). Seine Breite entspricht minimal dem uniformen Radius des Kommunikationsbereiches eines jeden Knotens und maximal dem doppelten Wert dieses Radius. Wird die Breite größer als der Senderadius gewählt, erlaubt das Protokoll für gewöhnlich die Koexistenz von mindestens zwei Weiterleitungsrouten, die sich nicht gegenseitig stören. An-

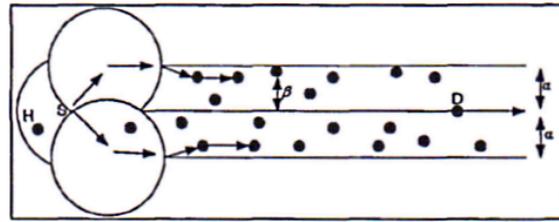


Abbildung 5.3: Weiterleitungsbereich des MRR [36] Protokolls. S ist der Forwarder und D ist der Zielknoten. Das Bild wurde dem Artikel [36] entnommen.

statt sich wie in den meisten anderen Protokollen an einem System wie GPS zur Positionsabfrage von Knoten zu bedienen, wird in diesem Artikel ein eigens dafür entwickelter Algorithmus verwendet. Dieser Algorithmus erfüllt nicht zwangsläufig alle Kriterien für beaconless Algorithmen, weswegen MRR unabhängig davon betrachtet wurde. Laut den Angaben der Autoren werden Knoten als nächster Hop bevorzugt, die Nachrichten von schwächeren Signalen empfangen, näher an den Rändern des Weiterleitungsbereiches liegen und mehr Energiereserven haben (s. dazu auch die Verzögerungszeitberechnung in Tab. A.13 zu MRR).

Die untersuchten Verfahren werden in Tabelle 5.1 verglichen. Darin stellt die Spalte zur *k-Lokalität* dar, aus welcher k -Hop-Nachbarschaft ein Knoten Informationen zur Berechnung des nächsten Hops beziehen muss; die *Pfadstrategie* gibt an, ob die Nachricht über einen oder mehrere Weiterleitungswege gleichzeitig zum Ziel geleitet wird; in der Spalte *Metrik* wird beschrieben, nach welcher Methode in einem Verfahren die Zähler der Kandidatenknoten berechnet werden; *Weiterleitungsbereich* gibt die möglichen Formen eines Weiterleitungsbereiches für das jeweilige Verfahren an; *UDG* veranschaulicht, welche Verfahren auf dem Unit Disk Graph Modell basieren.

5.1.2 Non-Aggressive Contention

Ein typischer Weiterleitungsschritt eines Greedy Routing Protokolls, das Non-Aggressive Contention verwendet, wird in der folgenden Liste beschrieben.

1. Der Forwarder sendet ein REQUEST-Paket über Broadcast an alle Nachbarn.
2. Jeder Empfänger der Nachricht berechnet und startet einen Zähler. Welche Faktoren bei der Berechnung dieses Zählers eine Rolle spielen, hängt vom Ziel des jeweiligen Protokolls ab. Generell gilt jedoch, je kürzer der Zähler, desto besser ist dieser Knoten als nächster Hop geeignet.

Tabelle 5.1: Übersicht Aggressive Contention Protokolle

Name	Ref.	Schleifenfrei	k-Lokalität	Pfadstrategie	Metrik	Zustands- behaftet	Recovery	MAC-cross- layer	Weiterlei- tungsbereich	UDG	Jahr
BLR	[23]	✓	1	S	D		e/CR		SRdK	✓	2004
BGR	[25]	? ^a	1	S	D	✓	e		? ^a	✓	2005
BLR-E	[32]	✓	2	S	K	✓			SRdK	✓	2007
NB-	[28]	✓	1	S	D				Gredy		2007
GEDIR								Area			
CBGR	[24]	✓	1	S	D	✓	e		S	✓	2008
DBO	[34]	✓	1	? ^b	R	✓		✓	- ^c	✓	2009
EABGR	[26]	✓	1	S	DE		e		3Dk	✓	2009
ABC	[27]	✓	2	S	D		e		S	✓	2010
DAF	[33]	✓	1	S	Z				S	✓	2010
BMR	[29]	✓	1	S	DE	✓			- ^c	✓	2011
IFSA	[30]	? ^a	1	S	DE				k ^d	✓	2011
MRR	[36]	? ^a		M	DES				RBVC ^e	✓	2011
RBRR	[35]	✓	1	S	DZk				Rd	✓	2013

Legende

Pfadstrategie: S = Single Path, M = Multipath

Metrik: D = Distanz, E = Energie, S = Signalstärke, Z = Zeit, K = Knotendichte, Zk = Zeitkritisch, R = Zone/Zufall

Recovery: e = eigenes Verfahren, CR = Clockwise Relaying (s. Kap 2.1.3)

Weiterleitungsbereich: S = Sektor, Rd = Reuleaux Dreieck, K = Kreis, k = Kegel, 3Dk = Kegel (3D Sektor)

^aImplementationsabhängig

^bSingle Path; Paketduplikationen können nicht selten auftreten, die genommenen Pfade konvergieren aber in Richtung Ziel

^cumgekehrter Weiterleitungsbereich: Das Gebiet um einen Zielknoten wird in kreisförmige Zonen aufgeteilt und eine Nachricht wird von Zone zu Zone in Richtung Ziel weitergeleitet.

^dVier mögliche, drei-dimensionale Formen: Kegel mit 60° Innenwinkel; Kegel mit 120° Innenwinkel (nicht empfohlen); parabolischer

Kegel; elliptischer Kegel.

^estatisches Rechteck mit festgelegter Breite; erstreckt sich von initialem Forwarder bis Zielknoten

3. Nach Ablauf seines Zählers antwortet ein Knoten dem Sender des REQUEST- mit einem RESPONSE-Paket.
4. Knoten, die ein RESPONSE-Paket erhalten, während ihr Zähler noch läuft, brechen diesen ab und antworten nicht.
5. Nach Erhalt des ersten RESPONSE-Paketes sendet der Forwarder ein SELECTION-Paket über Broadcast. Dieses kündigt an, welcher Knoten der nächste Hop werden wird; für gewöhnlich ist das der Knoten, der zuerst ein RESPONSE-Paket gesendet hat. In diesem Schema enthält entweder das REQUEST- oder das SELECTION-Paket das weiterzuleitende Datenpaket.

In diesem Schema kann der Weiterleitungsbereich voll ausgenutzt werden, da die Zähler von potentiellen nächsten Hops sowohl von RESPONSE- als auch von SELECTION-Paketen abgebrochen werden können und alle Knoten zwangsläufig im Empfangsbereich dieser Nachrichten liegen. Die Paketauslieferungsrates kann so maximiert werden. Jedoch benötigt dieses Schema viele Kontrollnachrichten, die zusätzlichen Datenverkehr innerhalb des Netzwerks und einen höheren Energieverbrauch eines jeden Knoten bedeuten. Viele Verfahren verwenden deshalb große Weiterleitungsbereiche, die meist im Greedy-Forwarding alle Knoten umfassen, die sich näher am Ziel befinden als der aktuelle Forwarder.

Die Algorithmen SIF [37], OGF [38], IGF [15], CBRR [39], GDBF [17] BLGF [40], BOSS [41], GeRaF [42] und PSGR [43] versuchen, die gesamte Hop-Anzahl von der Quelle bis zum Ziel möglichst gering zu halten. SIF, OGF, IGF und CBRR beachten dabei auch die verbleibende Energie eines Knotens, d.h. es werden Knoten bevorzugt, deren Energiereserven möglichst unverbraucht sind. SIF und IGF haben sonst keine erwähnenswerten Besonderheiten.

OGF [38] benutzt Weiterleitungstabellen, d.h. jeder Knoten merkt sich, an welchen Nachbarn er eine Nachricht für einen bestimmten Zielknoten weitergeleitet hat. Hat er für einen Zielknoten noch nie eine Weiterleitung getätigt, initialisiert er einen Wettbewerb unter seinen Nachbarknoten, um den geeignetsten nächsten Hop zu finden. OGF und SIF [37] haben keine feste Formel zur Berechnung des Verzögerungswertes, bieten aber jeweils eine Beispielfunktion an, die sich sehr ähnlich sind.

Die Zählerberechnung in CBRR [39] beachtet den Abstand eines Knotens zum Ziel, die verbleibende Energie des Knotens sowie die Anzahl von Paketen, die noch zu senden sind. Die Knoten sind nicht zustandslos; jeder Knoten n führt eine Nachbartabelle, in denen er Nachbarknoten einträgt, die schon mal einen von n gestarteten Wettbewerb zum Finden eines nächsten Forwarders gewonnen haben. Dies beschleunigt zukünftige Weiterleitungen mit ähnlichen Auslieferungsbedingungen, da kein neuer Wettbewerb gestartet werden muss. Das Protokoll kann wahlweise Tabellen zur ein-Hop oder zwei-Hop Nachbarschaft führen. CBRR ist,

wie RBRR [35] (s. Kapitel 5.1.1), ein zeitkritisches Protokoll. Bei der Initialisierung einer Weiterleitung wird eine „Deadline“ für das zu sendende Datenpaket festgelegt; das ist der Zeitpunkt, zu dem das Paket am Ziel angekommen sein muss. Anhand des „speed“-Werts eines Knotens – abhängig von der Distanz zum Ziel und der Verzögerung beim Weiterleiten der Nachricht – kann geprüft werden, ob ein Knoten als Forwarder geeignet ist. Liegt sein Wert unter dem verlangten Wert (s. Tab. A.18), kann eine pünktliche Auslieferung nicht garantiert werden und der Knoten wird vom Wettbewerb für den nächsten Hop ausgeschlossen.

Das GDBF [17] Protokoll sucht einfach den dem Ziel nächsten Knoten und wählt diesen als nächsten Forwarder aus. Im Falle einer Kollision zweier Nachrichten beim Empfänger durch zwei gleichzeitig antwortende Knoten bittet der Forwarder diese Knoten um eine erneute Übertragung. Die betroffenen Knoten setzen daraufhin zufällige Zähler, unabhängig von ihrer Position, um eine erneute Kollision zu vermeiden. Die Recovery-Methode von GDBF ist beaconless und wird in Kapitel 4 angesprochen.

BLGF [40] sucht zu jedem Hop auch einen sog. Relay-Knoten. Für den Fall, dass der nächste Hop ein korruptes Datenpaket empfängt, welches er nicht lesen kann, wird das Datenpaket vom aktuellen Forwarder über den Relay-Knoten zum nächsten Hop weitergeleitet. Der Umweg über den Relay-Knoten sollte eine höhere Stabilität aufweisen als der direkte Weg von Forwarder zu nächstem Hop. Es operiert sonst während des Greedy-Routings genauso wie BOSS. Im Gegensatz zu BOSS wird aber für das Recovery-Routing das beaconless BFP-Verfahren angewendet (s. Kap. 5.2).

BOSS [41] ist speziell an fehleranfällige Netzwerke angepasst. Es werden nur Knoten als Kandidatenknoten gewählt, deren Verbindung stabil genug ist, das gesamte Datenpaket vom Forwarder zu empfangen. Zusätzlich wird der Empfang von Kontrollnachrichten bestätigt und ggf. werden Kontrollnachrichten bis zu mehrere Male erneut übertragen. Der Weiterleitungsbereich wird in Zonen in Abhängigkeit zur Entfernung zum Ziel unterteilt. Die Verzögerungszeit eines Knotens hängt von der Zone ab, in der er sich befindet; so werden die Antwortzeitpunkte der Knoten zunächst grob getrennt. Die Antwortzeitpunkte von Knoten innerhalb der selben Zone werden zusätzlich durch einen hinzuaddierten Zufallswert unterschieden. Kollisionen durch gleichzeitig antwortende Knoten sollen durch dieses Vorgehen reduziert werden.

GeRaF [42] unterteilt den Weiterleitungsbereich eines Forwarders in Zonen. Nach dem Versenden einer RTS-Kontrollnachricht antworten andere Knoten abhängig von der Nähe der Zone, in der sie sich befinden, zum Ziel. Der Zeitpunkt, zu dem Knoten einer bestimmten Zone antworten, ist für diese Zone reserviert, allerdings antworten mehrere Knoten derselben Zone möglicherweise gleichzeitig und verursachen eine Kollision von Antworten; der Forwarder versteht die kol-

lidierten Nachrichten nicht. In diesem Fall senden die Knoten, die die Kollision verursacht haben, ihre Antworten erneut, wobei jeder Knoten jedoch mit einer Wahrscheinlichkeit von 0.5 nicht antwortet. Bei Bedarf kann die Prozedur mit den nun antwortenden Knoten wiederholt werden, bis nur noch eine einzige Antwortnachricht empfangen wurde. Der erste Knoten, dessen nicht-kollidierte Antwort empfangen wurde, wird der nächste Hop.

In [44] wird GeRaF um Funktionen zur Kollisionsvermeidung erweitert: Knoten prüfen vor dem Weiterleiten eines Pakets, ob der Kanal frei ist. Ist dies nicht der Fall, wartet ein Knoten eine bestimmte Zeit T_{sens} vor dem Senden des Pakets.

Wie GeRaF [42] unterteilt auch PSGR [43] den Weiterleitungsbereich eines Forwarders in Zonen. Es bietet zwei Heuristiken an, von denen die erste (DTD) den räumlichen Fortschritt in Richtung Ziel zu maximieren versucht und die zweite (modified DTD) so zeitlich schnell wie möglich das Ziel zu erreichen versucht. Dabei erhalten mit der ersten Heuristik Zonen, die näher am Zielknoten liegen, höhere Prioritäten; mit der zweiten Heuristik wird der Sendebereich in Sektoren unterteilt, bevor diese zusätzlich mit Zonen wie in der ersten Heuristik weiter partitioniert werden. PSGR wählt die Zonengröße dynamisch aus, um möglichst kleine Zone zu erhalten, die bestenfalls jeweils nur einen Knoten beherbergen. Da wie in GeRaF die Knoten zeitlich abhängig von ihrer Zone antworten, sollen so Kollisionen minimiert werden. Die Berechnung der Zonengröße wird anhand von Schätzungen zur Knotendichte getroffen.

EBGR [45], EBGRES [46], EEGR [47] und MEGAN [48] legen die höchste Priorität auf energieeffiziente Weiterleitung. Ein Forwarder berechnet dafür die nächste Hop-Position, die theoretisch die energieeffizienteste Weiterleitung ermöglicht, und wählt in einem zählerbasierten Wettbewerb den Knoten aus, dessen Position unter allen Kandidatenknoten die nächste zur berechneten optimalen Position ist (s. Abb 5.4). Die drei erstgenannten Artikel richten sich nach demselben Energiemodell und berechnen jene Position abhängig von der Energie, die zum Senden und Empfangen eines Datenbits benötigt wird (a_1), der Energie zur Verstärkung des Übertragungssignals (a_2) und eines Pfadverlustkoeffizienten k zu einer gegebenen Distanz x . Die Weiterleitung von der Quelle zum Ziel ist am effektivsten, wenn alle Hopdistanzen gleich d/N sind (mit $N - 1 =$ Anzahl aller weiterleitenden Knoten) und die Optimale Anzahl Hops $\lfloor d/d_0 \rfloor$ oder $\lceil d/d_0 \rceil$ beträgt mit

$$d_0 = \sqrt[k]{\frac{a_1}{a_2(k-1)}}$$

und d die Distanz von Quell- zu Zielknoten ist. Die Berechnungen für MEGAN sind komplexer, da sie noch die benötigte Energie zum Bewegen eines Knotens miteinbeziehen. Zudem sind die Zählerberechnungen der beiden erstgenannten Protokolle identisch. Im Gegensatz zu EBGR ist EBGRES an Sensornetze angepasst worden, deren Knoten ihre Energiereserven regenerieren können, beispielsweise über

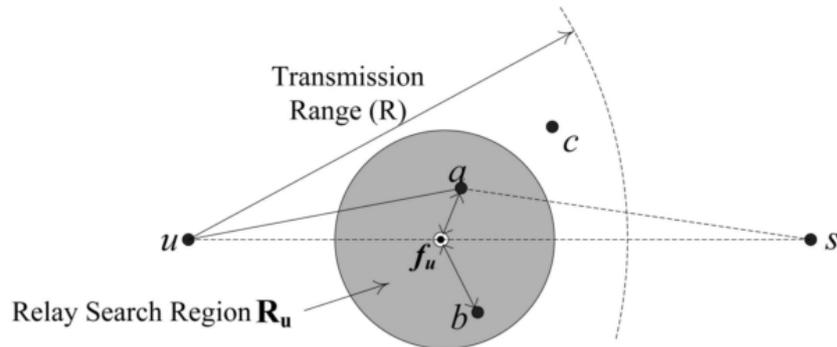


Abbildung 5.4: Weiterleitungsbereich (dunkelgrau ausgefüllter Kreis) in den Artikeln [45][46][47]. f_u ist die optimale Position des nächsten Hops des Forwarders u mit Zielknoten s . Bild entnommen aus [45].

Solarzellen. Zusätzlich verbessert EBGRES das Verhalten des Protokolls unter realistischen Bedingungen, ähnlich wie BOSS [41]. Auch MEGAN wurde für solche Netzwerke wie EBGRES entwickelt. Allerdings legt MEGAN die höchste Priorität auf die verbleibende Energie eines Kandidatenknotens nach seiner hypothetischen Weiterleitung. Es ist zudem das einzige uns bekannte Verfahren, welches Knotenmobilität aktiv nutzt. Vor der Weiterleitung bewegt sich der neue Forwarder zur optimalen Position.

Das GRoVar [49]-Verfahren wählt möglichst kleine Hops in Richtung des Ziels, um Energie zu sparen. Dazu regelt es die Sendestärke des Forwarders und erhöht diese schrittweise, solange kein Kandidatenknoten gefunden wird. Datenpakete erreichen laut den Autoren des Artikels schneller ihr Ziel als mit gewöhnlichen hop-minimierenden Verfahren, weil keine großen Umwege ausgewählt werden.

Im MACRO-Protokoll [50] sendet ein Forwarder ein Paket zunächst mit der geringsten Sendeleistung aus. Empfänger entnehmen dem Paket den Fortschrittswert (s. Kap. A.30) des Forwarders bezüglich eines Ziels und bereiten eine Antwort vor, falls sie besseren Fortschritt bieten. Diese Antwort wird zeitlich abhängig vom eigenen Fortschrittswert abgeschickt; je besser der Fortschritt, desto früher wird geantwortet. Der Forwarder sammelt solange Antworten, bis er schätzungsweise keine besseren Kandidaten finden kann; dann bricht er den Wettbewerb ab. Der beste Kandidat wird neuer Forwarder. Andernfalls wird die Sendeleistung erhöht und eine weitere Wettbewerbsrunde wird gestartet. Der Fortschrittswert ist sowohl abhängig von der Distanz der Knoten zum Ziel als auch von der aktuell verwendeten Sendestärke. Das Protokoll versucht die Anzahl Hops bis zum Ziel zu minimieren und gleichzeitig die benötigte Energie zur Forwardersuche zu reduzieren.

PF [51] und PFMAC [52] wählen möglichst weit entfernte Hops, welche möglichst dicht an der Strecke zwischen Forwarder und Zielknoten liegen. Beide Fakto-

ren haben die selbe Priorität. Eine Besonderheit dieser Protokolle: ihre Recovery-Strategien sind auch beaconless. Im Falle eines lokalen Minimums (s. Recovery Routing) wird einfach eine andere Berechnung der Verzögerungszeit benutzt.

BGM [53] ist ein Multipath Protokoll für Knoten-disjunkte Pfade. Es teilt den Weiterleitungsbereich in bestenfalls Knoten-disjunkte Zonen auf. Die Berechnung des Verzögerungswertes eines Knotens ist abhängig von der Zone, in der er sich befindet.

Die Verfahren aus SIF [37], OGF [38], IGF [15], BLGF [40], GeRaF [42], EBG-GR [45], EBGRES [46], MACRO [50] und PFMAC [52] haben MAC-Cross-Layer-Designs.

Die untersuchten Verfahren werden in Tabelle 5.2 verglichen. Darin stellt die Spalte zur *k-Lokalität* dar, aus welcher *k*-Hop-Nachbarschaft ein Knoten Informationen zur Berechnung des nächsten Hops beziehen muss; die *Pfadstrategie* gibt an, ob die Nachricht über einen oder mehrere Weiterleitungswege gleichzeitig zum Ziel geleitet wird; in der Spalte *Metrik* wird beschrieben, nach welcher Methode in einem Verfahren die Zähler der Kandidatenknoten berechnet werden; *Weiterleitungsbereich* gibt die möglichen Formen eines Weiterleitungsbereiches für das jeweilige Verfahren an; *UDG* veranschaulicht, welche Verfahren auf dem Unit Disk Graph Modell basieren.

5.1.3 Hybrid contention

Hybride Verfahren lassen sich nicht eindeutig einem Contention-Schema zuordnen. Die zwei einzigen Verfahren, die nach dieser Taxonomie zu dieser Algorithmenkategorie gehören, sind CBF [54] und HCGR [21]. Das erste Verfahren bietet verschiedene Ausführungsvarianten an, die sich nicht alle einer Contention-Strategie zuordnen lassen. Das zweite Verfahren greift je nach Position der Kandidatenknoten opportunistisch auf Aggressive- oder Non-Aggressive Contention zurück. Beide Algorithmen versuchen, die Anzahl der Hops vom Quellknoten bis zum Zielknoten möglichst gering zu halten.

CBF [54] bietet drei verschiedene sog. Unterdrückungsstrategien an, von denen die „Active Selection“-Strategie ein Non-Aggressive Contention-Schema verwendet. Die verbleibenden Strategien benutzen Aggressive Contention. Die Unterdrückungsstrategien beschreiben, in welchen Fällen Kandidatenknoten dem Forwarder nicht mehr antworten bzw. die erhaltene Nachricht nicht weiterleiten, nachdem der Forwarder eine Anfrage versendet hat. Nach der „Basic“-Strategie antworten Kandidatenknoten nicht mehr, falls ihre Wartezeit noch nicht abgelaufen ist und sie eine Weiterleitung durch einen anderen Kandidatenknoten hören. Um zu verhindern, dass mehrere Knoten die Nachricht weiterleiten, weil sie sich

Tabelle 5.2: Übersicht Non-Aggressive Contention Protokolle

Name	Ref.	Schleifenfrei	k-Lokalität	Pfadstrategie	Metrik	Zustands-behaftet	Recovery	MAC-cross-layer	Weiterlei-tungsbereich	UDG	Bemerkung	Jahr
GeRaF	[42]	✓	1	S	D			✓	GU	✓		2003
IGF	[15]	✓	1	S	DE		e	✓	Rd	✓		2003
SIF	[37]	✓	1	S	DE		e	✓	G	✓	statische Ziele	2005
GDBF	[17]	✓	1	S	D		e		sb	✓		2006
BOSS	[41]	✓	1	S	D		e		GU			2007
EEGR	[47]	✓	1	S	E				O	✓		2007
GROVar	[49]	✓	1	S	kh		e		sb			2007
MACRO	[50]	?	1	S	DE ^a			✓	sb			2007
OGF	[38]	✓	1	S	DE	✓	e	✓	G		statische Knoten	2007
PSGR	[43]	✓	1	S	D o. E		e		sb	✓		2008
CBRR	[39]	?	1-2 ^b	S	DEP	✓			sb	✓		2010
EBGR	[45]	✓	1	S	E		e	✓	O	✓		2010
PF/	[51]	?	1	S	DW		e	nein/	sbU			2010
PFMAC	[52]							✓				2011
BLGF	[40]	✓	1	S	D		BFP	✓	G			2011
BGM	[53]	✓	?	M	?	?			sbU	✓		2012
EBGRES	[46]	✓	1	S	E		e	✓	OU	✓		2012
MEGAN	[48]	✓	1	S	E				O		strikt mobile Knoten	2013

Legende

Pfadstrategie: S = Single Path, M = Multipath
 Metrik: D = Distanz, E = Energie, kh = kurze Hopdistanzen, P = Anzahl noch zu sendender Pakete, W = Winkel Kandidatenknoten - Forwarder - Ziel
 Recovery: e = eigenes Verfahren, BFP = Beaconless Forwarder Planarization (s. Kap. 2.1.3)
 Weiterleitungsbereich: Rd = Reuleaux Dreieck, sb = Sendebereich, U = Unterteilung in Untergebiete, G = Greedy Area/Positive Progress Area, O = Optimal Relay Position

^adurch Regulierung der Sendestärke

^bForwarding Table kann wahlweise Informationen zur 2-Hop-Nachbarschaft beachten

aufgrund zu großer Distanz von einander nicht hören, benutzt die „Area-based“-Unterdrückungsstrategie einen Weiterleitungsbereich, um die Distanz zwischen Kandidatenknoten zu begrenzen. Mit der „Active Selection“-Strategie leiten Knoten Nachrichten nicht direkt weiter, sondern antworten erst dem Forwarder, welcher dem zuerst antwortenden Knoten die Erlaubnis erteilt, die Weiterleitung durchzuführen.

Durch die Betrachtung des Winkels eines Knotens zur Strecke von Forwarder zu Zielknoten wird in HCGR [21] von jedem Kandidatenknoten ermittelt, ob er sich in der „Aggressive Area“ oder in der „Non-aggressive Area“ befindet. Je nachdem, in welcher dieser Zonen sich der erste antwortende Kandidatenknoten befindet, wird Aggressive Contention oder Non-Aggressive Contention benutzt. Dabei werden wartende Knoten aus der zuletzt genannten Zone von antwortenden Knoten aus der zuerst genannten Zone unterdrückt.

Die untersuchten Verfahren werden in Tabelle 5.3 verglichen. Darin stellt die Spalte zur *k-Lokalität* dar, aus welcher k-Hop-Nachbarschaft ein Knoten Informationen zur Berechnung des nächsten Hops beziehen muss; *Pfadstrategie* gibt an, ob die Nachricht über einen oder mehrere Weiterleitungswege gleichzeitig zum Ziel geleitet wird; in der Spalte *Metrik* wird beschrieben, nach welcher Methode in einem Verfahren die Zähler der Kandidatenknoten berechnet werden; *Weiterleitungsbereich* gibt die möglichen Formen eines Weiterleitungsbereiches für das jeweilige Verfahren an; *UDG* veranschaulicht, welche Verfahren auf dem Unit Disk Graph Modell basieren.

5.2 Recovery Routing

Recovery Routing wird in Unicast Routing Protokollen angewendet, wenn mit Greedy Routing eine Void Area erreicht wird. Die Nachricht wird mit Recovery Routing um die Void Area geleitet, bis ein Knoten gefunden wird, der die Bedingungen für Greedy Routing erfüllt.

Ein einfaches, von vielen Protokollen verwendetes, beaconless Recovery Verfahren ist z.B. „Forwarding Area Shift“ – das Verschieben des Weiterleitungsbereiches. Das findet in vielen Verfahren Anwendung, die den Weiterleitungsbereich für das Greedy Routing auf einen 60° Sektor mit der Strecke von Forwarder zu Ziel als Winkelhalbierende benutzen, wie es in vielen Aggressive Contention-Protokollen der Fall ist. Falls in diesem Sektor kein Kandidatenknoten gefunden wird, wird der Sektor gedreht und ggf. die Nachricht zur Weiterleitungsanfrage um die Information ergänzt, dass nun Recovery Routing betrieben wird. Die Weiterleitung findet so lange im Recovery Modus statt, bis ein Knoten gefunden wurde, der näher am Ziel

Tabelle 5.3: Übersicht hybrid contention Protokolle

Name	Ref.	Schleifenfrei	k-Lokalität	Pfadstrategie	Metrik	Zustands- behaftet	Recovery	MAC-cross- layer	Weiterlei- tungsbereich	UDG	Jahr
CBF	[54]	✓	1	S	D		e ^a		sb/Rd ^a /K ^a ✓	✓	2003
HCCR	[21]	✓	1	S	DW		e		GU	✓	2011

Legende

Pfadstrategie: S = Single Path

Metrik: D = Distanz, W = Winkel Kandidatenknoten - Forwarder - Ziel

Recovery: e = eigenes Verfahren

Weiterleitungsbereich: Rd = Reuleaux Dreieck, K = Kreis, sb = Sendebereich, U = Unterteilung in Untergebiete, G = Greedy Area/
Positive Progress Area

^anur im „Area-based Suppression“-Unterdrückungsschema

liegt als der Knoten, bei dem in den Recovery Modus gewechselt wurde. In diesem Fall wurde Fortschritt in Richtung des Zielknotens gemacht, d.h. das Resultat des zeitweiligen Recovery Routings entspricht einem Greedy Routing Schritt.

Die vorgestellten Recovery Routing Protokolle verfahren alle nach einem gemeinsamen Schema:

1. Der Forwarder sendet ein RTS-Paket mit Informationen zu seiner Position und ggf. mit Informationen zur Position des letzten Hops per Broadcast.
2. Jeder Empfänger des Pakets berechnet und startet einen Zähler. Für die Berechnung des Zählers spielen die Informationen aus dem RTS-Paket sowie die Position des Empfängers eine Rolle.
3. Nach Ablauf seines Zählers sendet ein Knoten ein CTS-Paket per Broadcast aus.
4. Gegebenenfalls folgt eine Protestphase, in welcher Knoten gegen die Wahl eines bestimmten neuen Forwarders protestieren, weil dieser protokoll- bzw. graphenspezifische Regeln verletzt.

Der Recovery-Modus von GDBF [17] (s. auch Kap. 5.1.2 und Tab. A.19) ist ein Topologiekontrollverfahren (s. Kap. 4), das alle Nachbarn des Forwarders im Gabriel Graph-Modell (s. Kap. 4.1) berechnet. Der Forwarder kennt so alle Nachbarn im Gabriel Graph – einem planaren Subgraphmodell – und kann darauf z.B. die nächste Kante für FACE-Routing bestimmen.

Die meisten Verfahren, die in den nächsten Absätzen genauer besprochen werden, benutzen ähnliche Vorgehensweisen. Meistens wird zu zwei gegebenen Knoten – Forwarder und Ziel oder Forwarder und letzter Hop – ein dritter Knoten aus der Menge aller Kandidatenknoten gesucht, so dass ein bestimmter Winkel minimiert wird. Bildhaft kann man sich vorstellen, dass eine Halbgerade durch den Forwarder gezogen und um ihn gedreht wird. Der erste Knoten, der von ihr getroffen wird, ist der nächste Forwarder. Lediglich wo die Halbgerade mit der Drehung startet, unterscheidet sich von Verfahren zu Verfahren. Die Richtung, in die sich die Halbgerade dreht, ist implementationsabhängig; die Drehrichtung muss jedoch innerhalb einer kompletten Weiterleitungsrunde beibehalten werden (*Linke-Hand-Regel* oder *Rechte-Hand-Regel*).

Zum BLR-Protokoll (s. auch Kapitel 5.1.1) existieren zwei Artikel, in denen verschiedene Recovery-Verfahren benutzt werden. Das erste, namenlose Verfahren [23], ist nicht beaconless (s. Tabelle A.1). Im zweiten Verfahren [55], Clockwise Relaying (CR), setzen Empfänger des RTS einen Zähler in Abhängigkeit zum Winkel $\angle dvw$ aus ihrer eigenen Position w , der Position v des Forwarders und der Position d des Ziels. Der zuerst antwortende Knoten wird zum Forwarder; es gibt

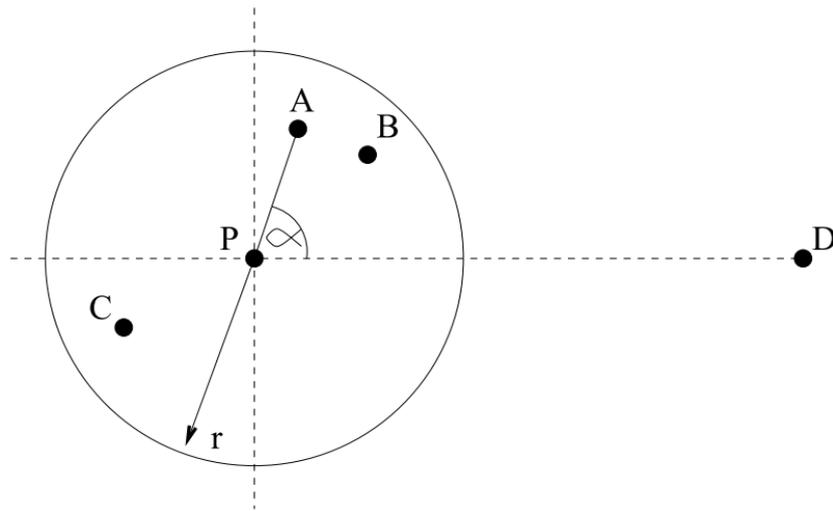


Abbildung 5.5: Weiterleitungsentscheidung in Clockwise Relaying. Knoten P ist Forwarder, Knoten D ist Zielknoten, die verbleibenden Knoten sind Kandidatenknoten. Der Winkel α minimiert die Funktion (s. Tab. A.34), darum wird A als Forwarder gewählt. Bild entnommen aus [55].

keine Protestphase. Ein Knoten w , dessen Winkel knapp unter 90° liegt, hat die kürzeste Verzögerungszeit (s. Bild 5.5 und Tab. A.34). Der alte Forwarder sendet daraufhin ein STN-Paket, um allen Knoten mitzuteilen, dass ein neuer Forwarder gefunden wurde. Sie brechen ihre Zähler daraufhin ab.

Im Recovery-Verfahren von PSGR [43] (s. auch Kapitel 5.1.2), PSGR-Bypass, setzen Empfänger des RTS einen Zähler in Abhängigkeit zum Winkel $\angle uvw$ aus ihrer eigenen Position w , der Position v des Forwarders und der Position u des letzten Hops (s. Abb. 5.6). Der Knoten w , der diesen Winkel minimiert, antwortet zuerst. Das Verfahren verläuft somit ähnlich zu CR in BLR [55].

NB-FACE [2] ist eine beaconless Umsetzung des FACE-Routing Protokolls [3]. Es berechnet den nächsten Gabriel-Graph-Nachbarn des Forwarders. Derjenige Nachbar w , für den der Winkel $\angle uvw$ (s. vorheriger Absatz) der kleinste ist, antwortet zuerst. Nachbarn, welche die Gabriel-Graph-Bedingung dieses Knotens und des Forwarders verletzen, melden dies gegebenenfalls während einer Protestphase. Die Reihenfolge, in der Nachbarknoten Nachrichten senden, ist nur vom Winkel abhängig; Bildlich kann man sich das so vorstellen, dass eine Halbgerade vom Forwarder aus gedreht wird und nacheinander Kandidatenknoten berührt. Wird ein Kandidatenknoten getroffen, antwortet oder protestiert er. Die Dauer der Drehung wird von einer Konstanten τ festgelegt. Eine volle Umdrehung bedeutet somit im schlimmsten Fall, dass jeder Nachbarknoten ein mal antwortet, was mit Beaconing gleichzusetzen ist; es wird aber mit Sicherheit ein Gabriel-Graph-Nachbar gewählt.

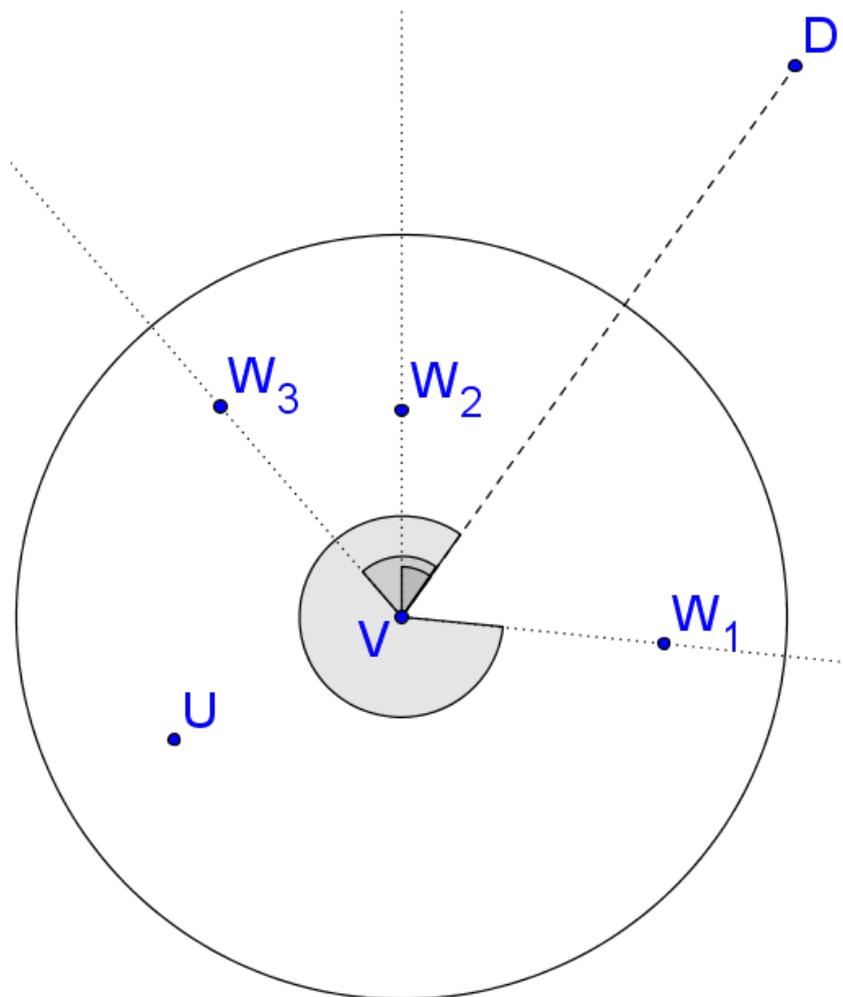


Abbildung 5.6: Weiterleitungsentscheidung in PSGR-Bypass. Bildlich gesprochen wird die Halbgerade von Forwarder V zu Zielknoten D (gestrichelte Linie) am Forwarder gedreht, bis der erste Kandidatenknoten W_i getroffen wird (punktirierte Linie durch die drei Knoten W_i); in diesem Fall ist dies Knoten W_2 . Dieser minimiert den betrachteten Winkel und wird der nächste Forwarder. Der Vollständigkeit halber wurden punktirierte Halbgeraden und Winkel zu allen Kandidatenknoten eingezeichnet.

Ist τ so klein, dass keine volle Umdrehung ausgeführt wird, kann ein Knoten fälschlicherweise als Gabriel-Graph-Nachbar erkannt werden, weil die Halbgerade einen Knoten, der gegen jenen Knoten protestieren will, nicht mehr erreicht (s. Abb. 5.7); NB-FACE hat in diesem Fall keine Auslieferungsgarantie.

Der Artikel von Rührup et al. [18] beschreibt zwei beaconless Verfahren. Das erste Verfahren, BFP, berechnet die Kanten eines Knotens zu seinen Nachbarknoten für ein planares Subgraphenmodell (z.B. Gabriel Graph). Das Verfahren wird in Kapitel 4 genauer betrachtet. Das zweite Verfahren, Angular Relaying (AR), berechnet die nächste Kante im Gabriel Graph für Face Routing, die für Recovery aus einem lokalen Minimum benutzt werden kann. Das Verfahren basiert auf einem zwei-stufigen Vorgehen, bei dem in der ersten Runde ein potentieller neuer Forwarder ausgewählt wird und in der zweiten Runde andere Knoten gegen diese Wahl protestieren können. AR funktioniert ähnlich wie NB-FACE [2], garantiert aber eine Auslieferung des Datenpakets ans Ziel. Anstatt die Protestphase wie in NB-FACE von einer Konstanten abhängig zu machen, wird in AR jedes mal eine neue Protestphase gestartet, wenn gegen einen Knoten protestiert wurde. Der protestierende Knoten wird zum vorläufigen nächsten Hop und die neue Protestphase bietet allen Kandidatenknoten die Möglichkeit, gegen ihn zu protestieren. Der Forwarder wartet, anders als in NB-Face, immer auf alle möglichen Antworten während einer Protestphase.

Der SC- und der TT Algorithmus [56] finden den nächsten Hop im Recovery Routing abhängig von dem Winkel zwischen dem letzten Hop, dem Forwarder und dem gewählten Kandidatenknoten. Bildlich gesprochen wird dabei ein Kreis (in SC, s. Abb. 5.8) oder ein Reuleaux Dreieck (in TT, s. Abb. 5.9), welches von dem Forwarder und dem letzten Hop aufgespannt wird, gegen den Uhrzeigersinn rotiert. Der nächste Knoten, der bei dieser Rotation von der Form berührt wird, ist der nächste Hop. Der Zählerwert wird in Abhängigkeit des Winkels $\angle uvw$ (s. Absatz zu PSGR) gesetzt. SC und TT benötigen keine Protestphase.

RS-Shortcut [57] ist eine Erweiterung zu SC und TT aus [56]. Während man mit SC immer einen Nachbarn im PDT-Graphen als nächsten Hop erhält, ist es in TT nicht immer gewährleistet. RS-Shortcut berechnet sowohl den nächsten Hop nach SC als auch nach TT. Sollten die beiden so gefundenen Knoten nicht identisch sein, werden für den mit TT gefundenen Knoten weitere Berechnungen vorgenommen, mit denen geprüft wird, ob die Kante vom Forwarder zu diesem Knoten zu einem Subgraphen gehört, der Auslieferungsgarantie bietet. Sollte dies der Fall sein, wird dieser Knoten gewählt, sonst wird der mit SC gefundene Knoten zum nächsten Hop.

ARROW [58] funktioniert nach einem anderen Schema als die bisher vorgestellten Protokolle. In jedem Paket sind Informationen zu den Koordinaten des Forwarders, der letzten beiden Hops, der Weiterleitungsrichtung und der Distanz

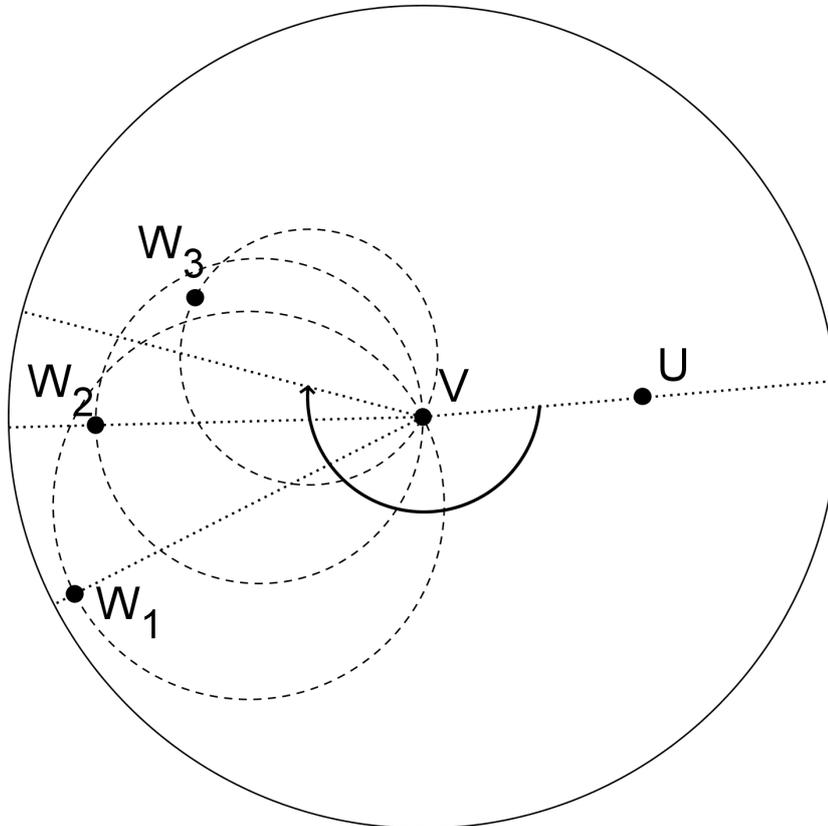


Abbildung 5.7: Veranschaulichung der Phasen von NB-FACE. Nachbarknoten W_i des Forwarders V antworten oder protestieren abhängig vom Winkel $\angle WVU$ mit letztem Forwarder U . Je kleiner der Winkel, desto früher die Antwort. Bildlich gesprochen wird von Forwarder V eine Halbgerade durch den letzten Forwarder U gezogen, die in Pfeilrichtung um V gedreht wird. Wie lange diese Drehung dauert, ist abhängig von einer Konstanten τ . In diesem Beispiel antwortet erst Knoten W_1 . Knoten W_2 protestiert, weil er im Gabrielkreis (gestrichelte Kreise) von V und W_1 liegt. W_2 wird durch den Protest als Gabriel-Graph-Nachbar erkannt. Jedoch liegt W_3 im Gabrielkreis von V und W_2 , protestiert aber nicht, weil τ zu klein gewählt wurde. W_2 wird somit fälschlicherweise für einen Gabriel-Graph-Nachbarn von V gehalten und als nächster Forwarder gewählt.

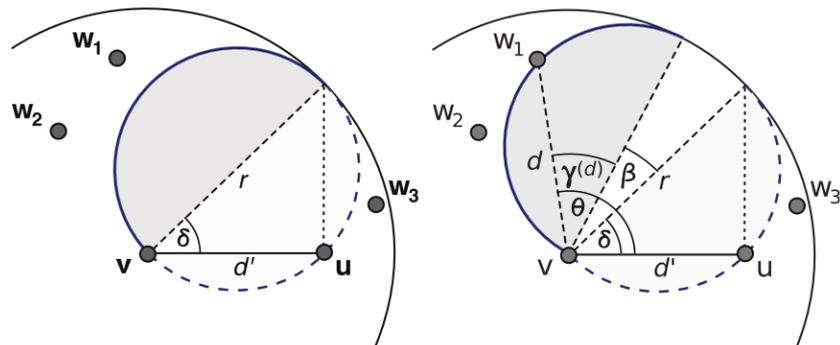


Abbildung 5.8: Sweep Circle. Beginnend von Forwarder v mit letztem Hop u und Startwinkel δ wird ein Kreisbogen gegen den Uhrzeigersinn um den Forwarder innerhalb seines Sendebereichs (äußerer Kreis) gedreht. Kandidatenknoten antworten in der Reihenfolge, in der sie von dem Bogen berührt werden. Bild entnommen aus [56].

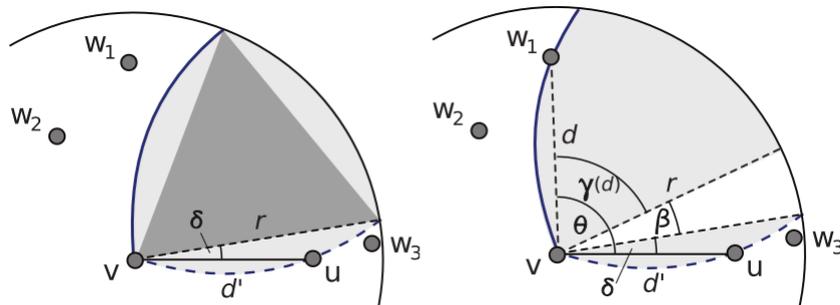


Abbildung 5.9: Twisting Triangle. Beginnend von Forwarder v mit letztem Hop u und Startwinkel δ wird ein Bogen eines Releaux Dreiecks gegen den Uhrzeigersinn um den Forwarder innerhalb seines Sendebereichs (Kreis) gedreht. Kandidatenknoten antworten in der Reihenfolge, in der sie von dem Bogen berührt werden. Bild entnommen aus [56].

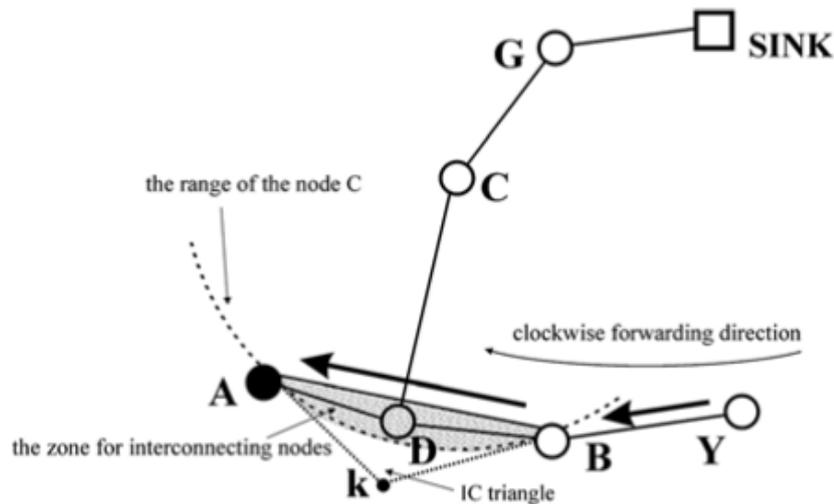


Abbildung 5.10: Weiterleitungsentscheidungen des ARROW-Protokolls. Entnommen aus [58].

des Zielknoten zum Knoten, bei dem das Recovery Routing angefangen hat, enthalten. Das Datenpaket wird immer in eine bestimmte Richtung weitergeleitet; im ersten Weiterleitungsschritt sucht der Forwarder v den Nachbarknoten u , sodass der *Azimuthwinkel* $\angle uvd$ mit Ziel d minimiert wird. Dieser Winkel überschreitet nie 180° ; in diesem Fall würde der Gegenwinkel geprüft. In den nachfolgenden Schritten wird immer in die Richtung weitergeleitet, in der u von v aus betrachtet lag; der nächste Forwarder u' ist der Knoten, der den Winkel $\angle u'uv$ mit aktuellem Forwarder u und letztem Hop v minimiert. Um Kantenschnitte zu vermeiden, wird zunächst ein Knoten D im Dreieck, das vom Kandidatenknoten A , dem Forwarder B und einem Punkt k (s. Abb. 5.10) aufgespannt wird, gesucht. Der Punkt k wird so gewählt, dass $\angle ABk$ und $\angle BAk = 30^\circ$ und $\angle AkB = 120^\circ$ entsprechen. Falls sich in diesem Dreieck mehr als ein Knoten befindet, wird der Knoten D_i mit dem größten Winkel $\angle AD_iB$ gewählt. Dieser Knoten sendet eine Nachricht, mit der er einen Knoten auf der anderen Seite der Strecke BA entdecken will. Falls ein solcher Knoten sich meldet, wird er der nächste Forwarder. Andernfalls wird die Nachricht zum Forwarder zurückgeleitet und die Nachricht wird einen weiteren Schritt entlang des Randes der Void-Area weitergeleitet.

ALBA-R [59] ist eine erweiterte Variante des Greedy Routing Protokolls ALBA [60]. Auch ALBA-R unterscheidet sich erheblich von den restlichen Protokollen, darum wird es nicht in der Tabelle 5.4 aufgeführt. In diesem Protokoll merken sich Knoten Zustände (Farben), die angeben, wie gut ein Knoten als neuer Forwarder geeignet ist. Im initialen Zustand haben alle Knoten den selben Zustand (gelb). Findet ein Knoten nach einer genügend hohen Anzahl von Weiterleitungsversuchen

keinen neuen Forwarder, erhöht er seinen Zustand (z.B. von gelb auf rot) und kann nicht mehr von Knoten mit niedrigeren Zuständen als Forwarder gewählt werden. Während Knoten im Initialzustand Greedy Routing betreiben, kann ein Knoten in einem höheren Zustand Datenpakete auch entgegen der Zielrichtung weiterleiten, um eine Void Area zu umgehen. Er leitet das Datenpaket dabei immer nur an Knoten mit gleich hohem oder niedrigerem Zustand weiter. Kann der Knoten weiterhin keinen passenden Kandidatenknoten finden, erhöht er seinen Zustand. Dieses Protokoll ist nicht explizit beaconless oder zählerbasiert, kann aber durchaus mit einer passenden Zählerberechnung so benutzt werden. Dieses Schema ist allerdings nicht zustandslos und somit nur auf Netzwerken mit statischen Knoten anwendbar. ALBA-R wird von ROME [61] für Unicast Routing – d.h. nicht nur als Recovery Methode – benutzt und für den Einsatz in Netzwerken mit mobilen Knoten optimiert. Es ändert dabei erheblich die Art und Weise, durch die den Knoten Zustände (Farben) zugeordnet werden.

Die untersuchten Verfahren werden in Tabelle 5.4 verglichen. Darin stellt die Spalte zur *k-Lokalität* dar, aus welcher k-Hop-Nachbarschaft ein Knoten Informationen zur Berechnung des nächsten Hops beziehen muss; die *Komplexität im schlechtesten Fall* bezieht sich auf die Nachrichtenkomplexität der Verfahren; in der Spalte *Metrik* wird beschrieben, nach welcher Methode in einem Verfahren die Zähler der Kandidatenknoten berechnet werden; *MANET* gibt an, ob das Verfahren explizit auch in Netzwerken mit mobilen Knoten verwendet werden kann; *Protestphase* und *UDG* veranschaulichen, welche Verfahren Protestphasen benutzen bzw. auf dem Unit Disk Graph Modell basieren.

Tabelle 5.4: Übersicht recovery Protokolle

Name	Ref.	garantierte Auslieferung	K-Lokalität	Komplexität im schlechtesten Fall	Metrik	MANET	Protestphase	UDG	Jahr
CR	[55]		1	$O(1)$	$\triangleleft dvw$	✓		✓	2003
NB-FACE	[2]	? ^a	1	$\Theta(n)$ ^a	$\triangleleft uvw$	✓	✓ ^a	?	2006
PSGR-Bypass	[43]		1	$O(1)$	$\triangleleft uvw$	✓		?	2008
AR	[18]	✓	1	$\Theta(n)$	$\triangleleft uvw$?	✓	✓	2010
ARROW	[58]	✓	2	$O(1)$	DW	?	?	✓	2013
RS-S	[57]	✓	2	$O(1)$	$\triangleleft uvw$			✓	2013
SC/TT	[56]	✓	1	$O(1)$	$\triangleleft uvw$			✓	2013

Legende

u: letzter Hop; v: Forwarder; w: Kandidatenknoten; d: Zielknoten; n: Anzahl Knoten im Sendebereich des Forwarders
 Metrik: D = Distanz, W = verschiedene Winkel

^aabhängig von einer gewählten Konstanten τ

Kapitel 6

Multicast Routing

Multicast Routing bezeichnet das Weiterleiten einer Nachricht von einem Quellknoten über andere Knoten zu einem oder mehreren Zielknoten (s. Abb 6.1). Beaconless Multicast Routing wird mit ähnlichen Methoden wie beaconless Unicast Routing benutzt, muss aber um Algorithmen ergänzt werden, die möglichst effektive Pfade innerhalb des Netzwerkes zu jedem Zielknoten finden. Um die totale Anzahl an Übertragungen innerhalb des Netzwerks zu reduzieren, sollten mehrere Nachrichten an verschiedene Knoten so lang wie möglich als ein Nachrichtenpaket verschickt werden. Erst, wenn eine bestimmte Bedingung erfüllt ist, z.B. einer der Zielknoten sehr nahe am Forwarder liegt, sollte die für diesen Knoten bestimmte Nachricht per Unicast an diesen Knoten weitergeleitet werden. In Abbildung 6.3 werden z.B. alle drei Nachrichten für die drei Zielknoten über zwei Hops als ein Paket verschickt. Erst, wenn ein Knoten nur noch einen Hop entfernt ist, wird ihm seine Nachricht zugesandt. Diesen Vorgang bezeichnen wir als „Abzweigung“. In dieser Abbildung erhalten wir so drei relativ kurze „Äste“, über die jede Nachricht einzeln per Unicast an ihren entsprechenden Zielknoten weitergeleitet wird. Den hier entstandenen Graphen der zurückgelegten Hops bezeichnen wir als „Baum“. In Abbildung 6.2 wird sehr früh abgezweigt, sodass wir in der Summe mehr Hops benötigen als in Abbildung 6.3, um alle Nachrichten auszuliefern. Die Suche nach möglichst effizienten Baumalgorithmen ist das Kernproblem im Gebiet der Multicast Protokolle.

Das BRUMA-Protokoll [62] basiert auf dem BOSS-Protokoll [41] für Unicast Routing. Anhand der Positionsinformationen der Zielknoten und des Quellknotens wird ein minimaler Spannbaum nach Kruskal berechnet, welchem die Anzahl aller Äste zur Weiterleitung einer Nachricht an alle Zielknoten entnommen werden kann. Das primäre Ziel dieses Protokolls ist laut den Autoren die Verringerung der benötigten Bandbreite durch Reduzierung zur Weiterleitung notwendiger Nachrichten. Die höchste Priorität bei der Auswahl eines Knotens als nächster Hop wird durch seine Nähe zu einem bestimmten Ziel bestimmt. Die Priorität eines

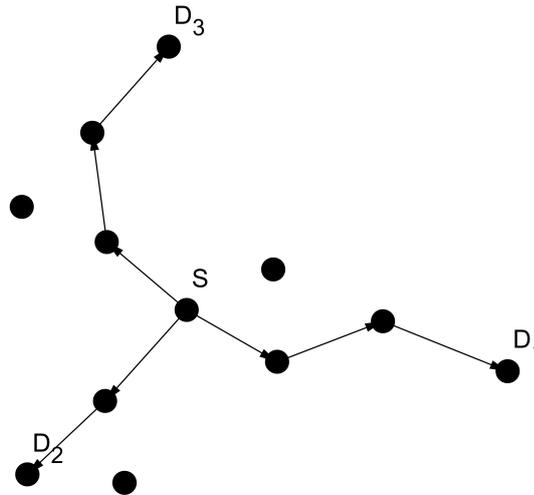


Abbildung 6.1: Knoten S möchte eine Nachricht an mehrere Knoten D_i senden.

Knotens, der sich sehr nah an einem Ziel befindet, ist höher als die Priorität eines anderen Knotens, der in einem Hop eine Nachricht an mehrere Zielknoten weiterleiten könnte.

Das GEM-Protokoll [63] wurde vom Steinerbaum inspiriert. Der Forwarder sammelt nach einer klassischen, zählerbasierten Fragerunde Informationen seiner Nachbarn ein. Die Antwort eines Knotens unterdrückt dabei nicht die Antwort eines noch wartenden Knotens, doch früher antwortende Knoten haben tendenziell bessere Weiterleitungspositionen. Der Forwarder bestimmt selbst, wie lange er auf Antworten wartet. Wie gut ein Knoten als nächster Hop geeignet ist, bzw. die Entscheidung, ob eine Nachricht in verschiedene Richtungen geschickt wird, hängt von einer „Average Progress“ Metrik ab (s. Abb. 6.1). Parameter dieser Metrik sind die Position eines Kandidatenknotens sowie die Anzahl und Positionen aller Zielknoten.

Die Autoren von DMPB [64] gehen nicht weiter auf das eigentliche beaconless Routing bzw. einen Zählermechanismus ein; das Protokoll lässt sich dementsprechend wohl mit jedem beliebigen Unicast Greedy Routing Algorithmus verwenden. Im Gegensatz zu anderen Multicast Routing Protokollen berechnet DMPB initial keinen Baum, sondern entscheidet jeweils ausgehend vom aktuellen Forwarder, ob eine Nachricht an mehrere Knoten gesendet werden muss. Dazu wird das Netzwerk vom Forwarder rechnerisch in vier Sektoren unterteilt und die durchschnittliche Position („Temporary Guiding Point“) aller Zielknoten ermittelt. Die Sektoren werden so berechnet, dass die zuvor berechnete Durchschnittsposition auf der Winkelhalbierenden eines Sektors liegt. Je nach Position jedes einzelnen Zielknotens wird die Nachricht zu einem Knoten des entsprechenden Sektors geschickt.

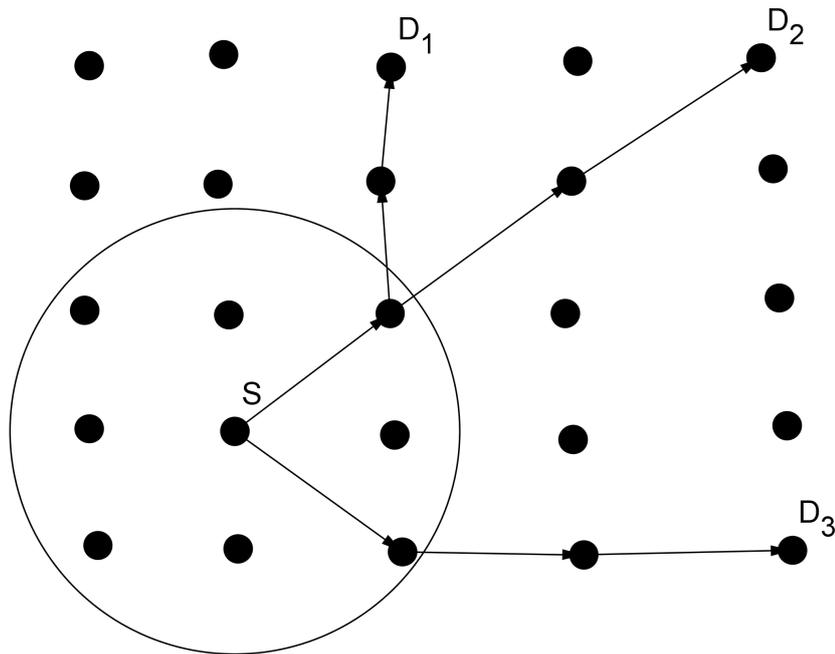


Abbildung 6.2: Schlechte Wahl von Abzweigungen. Quellknoten S möchte jeweils verschiedene Datenpakete an die Zielknoten D_1 , D_2 und D_3 senden. Die Nachricht für D_3 wird direkt über eine eigene Route geschickt, während die Nachrichten für D_1 und D_2 über den ersten Hop als ein gemeinsames Datenpaket übertragen werden. Erst im darauffolgenden Broadcast durch den nächsten Forwarder werden beide Pakete getrennt weitergeleitet. Es werden insgesamt sechs Broadcasts benötigt, um alle Zielknoten zu erreichen. Wir gehen in diesem Beispiel davon aus, dass jeder Broadcast die gleiche Energiemenge benötigt und jeder Knoten einen Sendebereich hat, der alle direkten Nachbarn umfasst, wie der Kreis um S es darstellt.

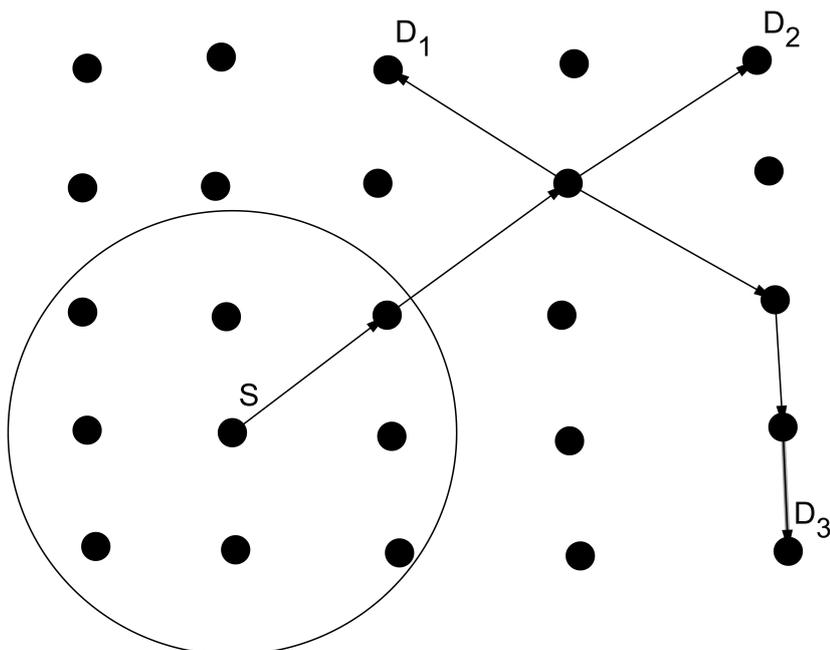


Abbildung 6.3: Bessere Wahl von Abzweigungen. Die drei Nachrichten für alle Zielknoten werden anders als in Abb. 6.2 hier über zwei Hops als ein Datenpaket weitergeleitet. Es werden insgesamt fünf Broadcasts benötigt, um alle Zielknoten zu erreichen.

Das Geocast Protokoll von Ko und Vaidya [65] geht im Bezug auf Nachrichtenaustausch zwischen Knoten nicht ins Detail, erfüllt aber die Kriterien eines beaconless Algorithmus. Geocast ist eine spezielle Art von Multicast; während in letztgenannten Protokollen meist explizite Ziele genannt werden, denen es Nachrichten auszuliefern gilt, wird im Geocast eine geographische Region bestimmt. Alle Knoten, die sich in dieser Region befinden, sind Zielknoten. Der oben genannte Artikel stellt zwei positionsbasierte Multicast Schemata vor, die potentiell von Knoten ausgeführt werden können, die keine Nachbarschaftsinformation haben. Einzige Voraussetzung ist das Vorhandensein eines Lokalisierungssystems wie GPS, mit dem Knoten ihre eigene Position abfragen können. Dies setzen fast alle anderen beaconless Protokolle ebenfalls voraus. Neben der Region für den Geocast wird in einem der Schemata auch ein Weiterleitungsbereich berechnet. Knoten, die die zu weiterleitende Nachricht empfangen, leiten diese nur weiter, wenn sie sich in diesem Bereich befinden (s. Abb. 6.4 und Abb. 6.5).

Die untersuchten Verfahren werden in Tabelle 6.1 verglichen. Darin stellt die Spalte zur *k-Lokalität* dar, aus welcher k-Hop-Nachbarschaft ein Knoten Informationen zur Berechnung des nächsten Hops beziehen muss; die *Pfadstrategie* gibt an, ob eine Nachricht über einen oder mehrere Weiterleitungswege gleichzeitig zum selben Ziel geleitet wird; in der Spalte *Metrik* wird beschrieben, nach welcher Methode in einem Verfahren die Zähler der Kandidatenknoten berechnet werden; *Weiterleitungsbereich* gibt die möglichen Formen eines Weiterleitungsbereiches für das jeweilige Verfahren an; *UDG* veranschaulicht, welche Verfahren auf dem Unit Disk Graph Modell basieren; die Spalte *Baumstruktur* steht für die einem Verfahren zugrundeliegende Baumstruktur, von der es abhängt, wann Nachrichten für verschiedene Ziele zusammen oder getrennt weitergeleitet werden.

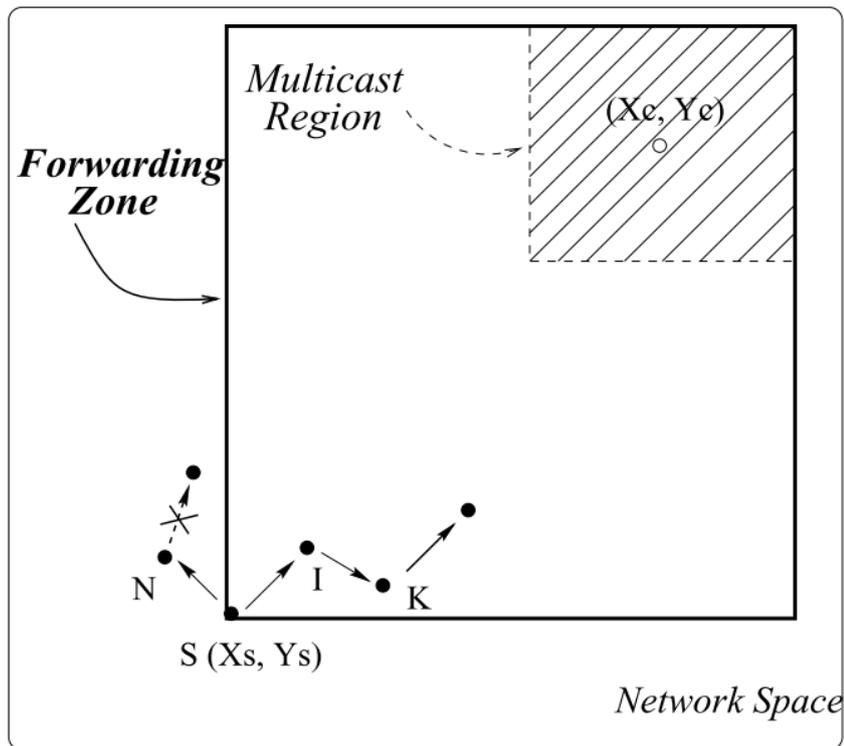


Abbildung 6.4: Geocast Variante 1. Die gestrichelte *Multicast Region* ist die Region, in der alle Knoten Zielknoten des Datenpakets sind. Alle Knoten die in der *Forwarding Zone* liegen, leiten ein erhaltenes Datenpaket über Broadcast weiter. Knoten außerhalb dieser Zone löschen das Datenpaket sofort. Die Koordinaten der *Forwarding Zone* werden vom Quellknoten berechnet und sind im Datenpaket enthalten. Bild entnommen aus [65].

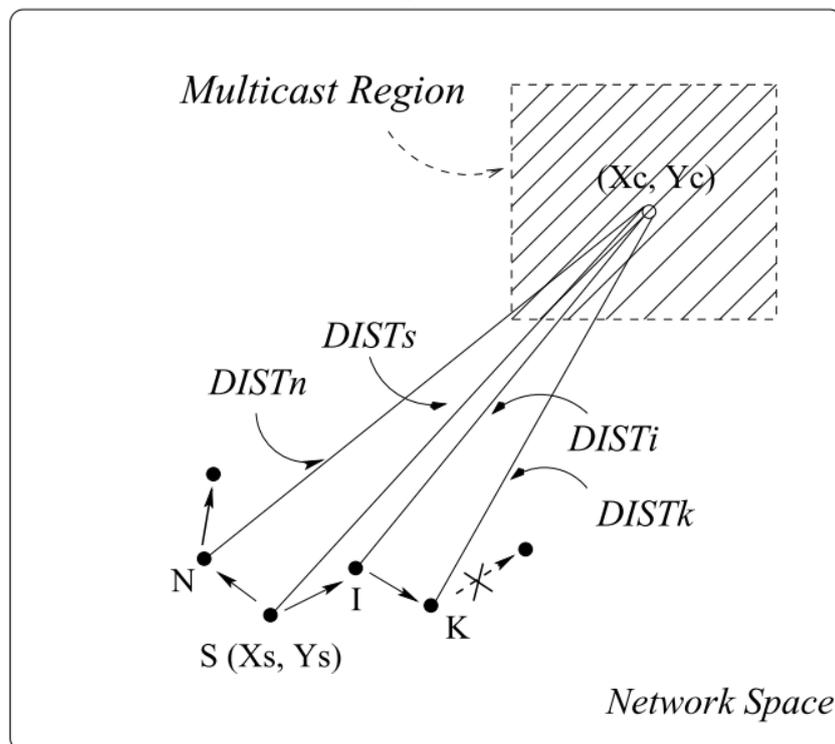


Abbildung 6.5: Geocast Variante 2. Ein Empfänger des Datenpakets entscheidet anhand seiner Distanz zum Mittelpunkt der *Multicast Region*, ob er die Nachricht per Broadcast weiterleiten soll. Die benötigten Daten zur *Multicast Region* sind im weiterzuleitenden Datenpaket enthalten. Bild entnommen aus [65].

Tabelle 6.1: Übersicht Multicast Protokolle

Name	Ref.	Schleifenfrei	k-Lokalität	Pfadstrategie	Metrik	Zustands- behaftet	Recovery	MAC-cross- layer	Weiterlei- tungsbereich	UDG	Baumstruktur	Jahr
Geocast	[65]	?	1	M	? ^a				F		e	1999
GEM	[63]	?	1	S	D			?	sb	✓	EST	2012
BRUMA	[62]	✓	1	S	D		e		G		K	2012
DMPB	[64]	✓	1	S	? ^a		i		S		e	2013

Legende

Pfadstrategie: S = Single Path, M = Multi Path

Metrik: D = Distanz

Recovery: e = eigene (undefiniert), i = implementationsabhängig

Forwarding Area: F = Forwarding Zone (beliebige Form; Knoten außerhalb der Zone leiten erhaltene Pakete nicht weiter), S = Sektoren,

sb = Sendebereich, G = Greedy Area

Baumstruktur: e = eigene (undefiniert), EST = euklidischer Steinerbaum, K = minimaler Spannb Baum nach Kruskal

^aauf die Metrik des Unicast Routings wird nicht eingegangen

Kapitel 7

VANET

„Vehicular ad hoc networks“, kurz VANETs, sind Netzwerke, deren Knoten auf Vehikeln, beispielsweise Automobilen, montiert sind. VANETs sind spezialisierte MANETs („Mobile ad hoc networks“) [66]. Beide Netzwerkarten haben viele Gemeinsamkeiten, VANETs haben aber zusätzliche Beschränkungen und Eigenschaften. Viele Protokolle ([67], [68], [69] und [70]) für solche Netzwerke gehen z.B. davon aus, dass jeder Knoten eine Karte aller Straßen kennt und so anhand seiner eigenen Position seinen Abstand zu Kreuzungen und anderen Fahrzeugen (Knoten) berechnen kann.

7.1 Routing

Routing in VANETs funktioniert in den meisten Fällen ähnlich zu Unicast Routing. Durch die Spezialisierung auf Vehikelknoten kommen weitere Einschränkungen dazu. In Städten beispielsweise bewegen sich Autos auf Straßen, also festgelegten Bahnen. Für den einfachsten Fall betrachten wir nur eine einfache, zweispurige Straße. Möchte ein Knoten eine Nachricht an einen anderen Knoten senden, muss die Nachricht entweder in die Richtung geschickt werden, in die sich das eigene Vehikel bewegt, oder entgegen dieser Richtung. Die Nachricht wird so lange von Vehikelknoten zu Vehikelknoten in eine der beiden möglichen Richtungen weitergeleitet, bis sie das Ziel erreicht. Ist das Straßennetz komplexer, kreuzen sich Straßen. In diesen Fällen muss die Nachricht in Richtung des Ziels bis zur nächsten Kreuzung weitergeleitet werden. An der Kreuzung muss entschieden werden, über welche Straße, d.h. in welche Richtung die Nachricht weitergeleitet werden soll.

Auch die Geschwindigkeit eines Vehikels spielt eine wichtige Rolle. Leitet ein Quellknoten eine Nachricht an einen Knoten weiter, dessen Vehikel sich in die selbe Richtung wie das Vehikel des Quellknotens bewegt, aber deutlich langsamer als dieses ist, überholt das Vehikel des Quellknotens das Vehikel des anderen Knotens

nach einer bestimmten Zeit. In diesem Fall ist es effizienter, zu warten, bis sich schnellere Vehikel in Reichweite befinden.

Die vorgestellten Verfahren funktionieren im Grunde wie ihre die entsprechenden Contention-Verfahren aus Kapitel 5.

7.1.1 Aggressive Contention

Die Verfahren aus RPBL [66] und CBFS¹ [69] versuchen, die Anzahl von Hops vom Quell- bis zum Zielknoten zu minimieren.

RPBL [66] setzt jeweils die nächste Kreuzung zwischen dem Forwarder und dem Zielknoten als temporäres Ziel fest. Die Nachricht wird in jedem Hop so nah wie möglich an das nächste temporäre Ziel gebracht. Es benutzt Dijkstras Algorithmus für kürzeste Pfade, um anhand eines aus einer digitalen Straßenkarte extrahierten Graphens den kürzesten Weg zum Ziel zu finden.

Die Grundidee des CBFS [69] Protokolls entspricht der von CBF [54] Protokoll (s. 5.1.3), das an Straßenszenarios angepasst wurde. Im Gegensatz zu CBF wurde in CBFS auf Unterdrückungsmechaniken verzichtet. Um mehrere gleichzeitige Weiterleitungen zu vermeiden, merken sich Knoten die Identifikationsnummern bereits weitergeleiteter Pakete und leiten Paketduplikate nicht weiter. CBFS verwendet keine digitale Straßenkarte.

7.1.2 Non-Aggressive Contention

Diese Verfahren funktionieren im Grunde wie Verfahren aus 5.1.2.

Die Verfahren BRAVE [67] und CBBPF [71] versuchen, die Anzahl von Hops vom Quell- bis zum Zielknoten zu minimieren.

BRAVE [67] merkt sich die nächsten beiden zu besuchenden Kreuzungen als temporäre Ziele. Für einen Straßenwechsel leitet der Forwarder die Nachricht an einen Knoten weiter, dessen Abstand zur übernächsten Kreuzung kleiner ist als der Abstand zwischen den beiden nächsten Kreuzungen (s. Abb. 7.1). Darüber hinaus verwendet BRAVE zur Weiterleitung von Nachrichten die selbe Vorgehensweise inklusive Zählerberechnung wie BOSS [41] (s. Kap. 5.1.2). Lediglich die Recovery-Strategie weicht von BOSS ab; BRAVE benutzt einen Store-Carry-Forward-Ansatz als Pseudo-Recovery Mechanismus. Dabei speichert der Forwarder das Datenpaket, bis sich ein Kandidatenknoten in Reichweite befindet (mehr Infos dazu s. Tab. A.42). Wie auch das RPBL benutzt BRAVE Dijkstras Algorithmus für kürzeste Pfade, um anhand eines aus einer digitalen Straßenkarte extrahierten Graphens den kürzesten Weg zum Ziel zu finden.

¹Das Protokoll wird nicht unter „CBFS“ im Artikel erwähnt; es wird dort immer auf CBF [54] verwiesen. Um Missverständnisse zu vermeiden, wird in dieser Arbeit die VANET-Variante von CBF um ein „S“ für „Street“ (Straße) ergänzt.

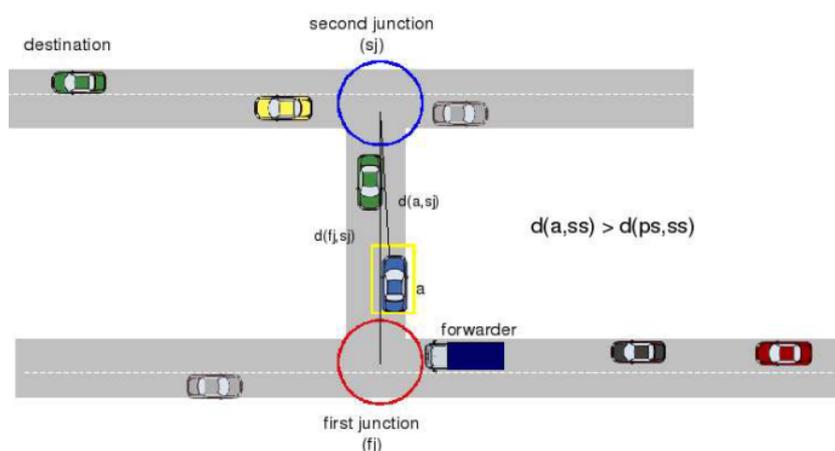


Abbildung 7.1: Der Forwarder unten im Bild möchte die Straße wechseln, auf der die Nachricht weitergeleitet wird. Er wählt als nächsten Hop a aus, da die Distanz von a zur übernächsten Kreuzung (*second junction*) kleiner ist als der Abstand zwischen den beiden Kreuzungen (*first junction* und *second junction*). Bild entnommen aus BRAVE [67].

S-BRAVE [72] ist ein Sicherheitsprotokoll, das auf BRAVE [67] basiert. Es schützt ein Netzwerk vor Angriffen, bei denen das Netzwerk mit Knoten infiltriert wird, welche vorgeben, Datenpakete weiterzuleiten, es aber nicht tun. Das Protokoll verwendet u.A. ein Beacon-basiertes Zertifikatsaustauschsystem und wird darum an dieser Stelle nur aus Gründen der Vollständigkeit erwähnt.

CBBPF [71] funktioniert wie eine vereinfachte Version von BRAVE, bzw. wie eine VANET-Version von BOSS [41]. Es bedient sich keiner Straßenkarte, sondern leitet die Nachricht wie in einem normalen DSN in jedem Schritt möglichst nah an das Ziel weiter. Es benutzt wie BRAVE einen Store-Carry-Forward Mechanismus, wenn es keinen Kandidatenknoten finden kann und verhält sich bei der Weiterleitung genau wie BOSS. Ein Unterschied zu BOSS ist lediglich, dass nicht direkt das Datenpaket bei der Forwarder-Suche gesendet wird, sondern ein kleines Kontrollpaket. CBBPF ist ein MAC-cross-layer-Protokoll.

IB [68] unterscheidet zwei Knotenzustände: zwischen Kreuzungen („between intersections“) und an Kreuzungen („at intersections“). Im ersten Fall berechnen Kandidatenknoten ihre Zähler abhängig von der relativen Richtung, in die sich ihr jeweiliges Vehikel bewegt, sowie der Stärke des empfangenen Signals, über welches die Weiterleitungsanfrage gesendet wurde (als Anhaltspunkt für die Qualität des Kanals zwischen beiden Knoten). Ein Knoten, der sich in die Richtung des Ziels bewegt und die Nachricht des Forwarders über ein Signal mit hoher Stärke empfangen hat, berechnet einen kurzen Zählerwert. Im zweiten Fall werden weitere Kriterien miteinbezogen, nämlich Informationen der digitalen Straßenkarte und die

eigene Distanz zum Ziel („Greediness-Factor“). Knoten mit kurzer Distanz zum Ziel werden bevorzugt.

Die untersuchten Verfahren werden in Tabelle 7.1 verglichen. Darin stellt die Spalte zur *Kommunikations-Technologie* dar, ob Knoten auf Vehikeln sich gegenseitig Nachrichten schicken, oder Knoten auf Vehikeln mit einer Infrastruktur, z.B. stationären Knoten an Straßenrändern kommunizieren; *Pfadberechnung* gibt an, ob und wie Nachrichtenwege auf Basis einer digitalen Straßenkarte berechnet werden; in der Spalte *Vehrkerswahrnehmung* wird angegeben, ob Knoten für eine Weiterleitungsentscheidung Informationen wie Verkehrsdichte einfließen lassen; *Anwendungsszenario* steht für das voraussichtliche Einsatzgebiet eines Verfahrens, meist entweder in Städten mit Kreuzungen oder in Autobahn-ähnlichen Szenarien ohne Kreuzungen; *Beachtete Kreuzungen* gibt an, ob und wie viele Kreuzungen als nächste Zielpunkte bei Weiterleitungsentscheidungen berücksichtigt werden; *UDG* veranschaulicht, welche Verfahren auf dem Unit Disk Graph Modell basieren.

7.2 Sicherheit

EBRM [70] enthält ein Sicherheitsprotokoll, das Knoten identifiziert, die Datenpakete nicht weiterleiten. Ein Knoten wartet nach Weiterleiten eines Datenpakets auf eine Kontrollnachricht des neuen Forwarders, die den Erhalt der Nachricht bestätigen soll. Wird diese Kontrollnachricht nicht vom letzten Forwarder empfangen, teilt er dies einer „road side unit“ (RSU) – eines Knotens am Straßenrand – mit. Übersteigt die Anzahl solcher durch eine RSU empfangenen Nachrichten einen bestimmten Schwellwert, teilt sie allen Knoten in ihrem Sendebereich mit, dass der gemeldete Knoten nicht als Forwarder gewählt werden sollte. EBRM beschreibt auch ein einfaches Weiterleitungsschema, welches jedoch ohne Beacons als Kernelement unseres Erachtens nach nicht umzusetzen ist. Wir beschränken uns in dieser Arbeit ausschließlich auf den Sicherheitsaspekt dieses Protokolls. Da es sich in dieser Form schlecht mit Weiterleitungsprotokollen vergleichen lässt, ist es nicht in der Tabelle 7.1 aufgeführt.

MEFAD [73] stellt kein vollständiges beaconless Verfahren vor. Der Artikel geht auf Angriffe durch Knoten ein, welche in zählerbasierten beaconless Verfahren bei Anfragen immer sehr früh antworten und dem Forwarder so eine sehr gute Position zur Weiterleitung des Datenpakets vortäuschen. Werden Informationen zu Knotenposition und -Geschwindigkeit zu Kontrollpaketen hinzugefügt, können Empfänger einer Antwortnachricht für einen Forwarder Berechnungen zur Verifikation der erhaltenen Daten durchführen. Beispielsweise muss der antwortende Knoten von seiner angeblichen Position aus trotzdem in Sendereichweite jedes

Tabelle 7.1: Übersicht VANET Protokolle

Name	Ref.	Digitale Straßenkarte	Kommunikations- Technologie	Fladberechnung	Fahrers- wahrnehmung	Anwendungs- szenario	Beachtete Kreuz- zungen	Recovery	UDG	Typ	Jahr
CBFS	[69]		V2V		H	H			✓	A	2004
BRAVE	[67]	✓	V2V	Dijkstra	U	U	2	SCF		N	2010
CBBPF	[71]		V2V	?	U	U	1	SCF		N	2013
IB	[68]	✓	V2V	?	U	U	1			N	2013
RPBL	[66]	✓	V2V	Dijkstra	U	U	1		✓	A	2013

Legende

Kommunikationstechnologie: V2V = Vehikel zu Vehikel; V2I : Vehikel zu Infrastruktur

Anwendungsszenario: U = urban (Stadtgebiet); H = highway (Autobahn)

Typ: A = Aggressive Contention; N = Non-Aggressive Contention

Recovery: SCF = store-carry-forward

Empfängers sein. Kann das laut der empfangenen Daten nicht der Fall sein, handelt es sich um einen Angriff und dieser wird dem Forwarder mitgeteilt. Grundsätzlich kann ein solches Sicherheitsschema von jedem VANET Weiterleitungsalgorithmus implementiert werden.

Kapitel 8

Fazit

Wir haben in dieser Arbeit eine Kategorisierung aller uns bekannten, lokalen beaconless Algorithmen auf Basis verfahrensähnlicher Gemeinsamkeiten vorgestellt. In den in Kapitel 3 genannten, einschlägigen Datenbanken wurde mit den dort genannten Suchbegriffen eine vollständige Literaturrecherche durchgeführt. Weitere potentiell relevante Artikel, die in den Literaturlisten der in der Datenbanksuche gefundenen Literatur referenziert wurden, wurden über die oben genannten Datenbanken hinaus einzeln untersucht. Wir können auf dieser Basis einen Anspruch auf Vollständigkeit über unseren Such- und Zeitraum erheben.

Es zeigte sich, dass es viel Redundanz in der Literatur gibt: viele Protokolle verfolgen sehr ähnliche Ziele und unterscheiden sich teilweise nur in wenigen kleinen Aspekten. Beaconless Verfahren sind sehr problemorientiert beschrieben und versuchen in den meisten Fällen eine von Beacon-behafteten Verfahren bereits gelöste Problemstellung durch einen beaconless Ansatz zu ersetzen.

Bis auf die in Kapitel 4 untersuchten Verfahren und der in Kapitel 7 angesprochenen Sicherheitsprotokolle dienen die untersuchten Verfahren ausschließlich der Datenkommunikation, d.h. der Weiterleitung einer Nachricht von einem Netzwerkknoten zu einem oder mehreren anderen Netzwerkknoten.

Insgesamt wurden in dieser Arbeit drei Topologiekontrollverfahren, 41 Unicast Routing-Verfahren (inkl. zweier nicht weiter erwähnenswerter Erweiterungen für Unicast Routing-Verfahren), vier Multicast Routing-Verfahren und fünf Routing-Verfahren für VANETs (inkl. zweier Sicherheitsprotokolle) untersucht.

Die untersuchten Unicast Routing-Verfahren stammen vermehrt aus den Jahren 2007 und 2010 bis 2011 (insgesamt 20 Protokolle); fünf Verfahren wurden im Jahr 2013 veröffentlicht, von denen drei aus dem Bereich des Recovery Routing kommen, was auf ein wachsendes Interesse an diesem Forschungsgebiet schließen lässt. Von den relevanten VANET-Protokollen wurden über die Hälfte im Jahr 2013 veröffentlicht, was auch auf ein steigendes Forschungsinteresse in diesem Teilgebiet der beaconless Verfahren hindeutet. Das Gleiche gilt für die untersuchten

Multicast Routing-Verfahren. Die drei Verfahren zu Topologiekontrolle sind zeitlich relativ weit gefächert mit einem Schwerpunkt auf die Jahre ab 2010. Während Unicast Routing-Verfahren bis auf den Recovery Routing-Bereich anscheinend bereits intensiv behandelt wurden und nur noch vereinzelt in den letzten Jahren Veröffentlichungen in diesem Gebiet stattfanden, wächst das Interesse an der Forschung aller anderen untersuchten Gebiete. Die Ausnahme bildet der Bereich der Topologiekontroll-Verfahren, welcher nur selten und zeitlich unsystematisch behandelt wird.

Abschließend bleibt zu sagen, dass diese Arbeit eine Übersicht aller lokalen beaconless Algorithmen darstellen und auf diese Weise Forschungslücken aufdecken soll, um die Forschung in diesem Gebiet voranzutreiben. Behandelte Probleme können schnell nachgeschlagen werden und offene Probleme können schneller als solche identifiziert werden.

Für eine tiefergehende Kategorisierung von beaconless Algorithmen bietet sich die Untersuchung des weiten Felds der beaconless Broadcast Algorithmen sowie das der zentralen beaconless Verfahren an. Die Datenbanksuche kann darüber hinaus mit weiteren Suchbegriffen ergänzt und auf weitere Datenbanken ausgeweitet werden.

Anhang A

Appendix

In diesem Kapitel finden sich Tabellen mit detaillierten Informationen zu den in dieser Arbeit untersuchten Verfahren.

A.1 Unicast Routing

A.1.1 Aggressive Contention

Tabelle A.1: BLR

Verfahren	Beacon-less routing algorithm (BLR) [23]
Form des Weiterleitungsbereiches	Sektor, Reuleaux Dreieck oder Kreis
Zählerberechnung	Implementationsabhängig: (1) $Max_{delay} * \left(\frac{r-p}{r}\right)$ (2) $Max_{delay} * \left(\frac{p}{r}\right)$ (3) $Max_{delay} * \left(\frac{e\sqrt{p^2+d^2}}{e}\right)$ p = Fortschritt = Abstand des Forwarders zur Projektion des Kandidatenknotens auf die Strecke zwischen Forwarder und Ziel r = Radius des Sendebereiches d = Abstand zur Linie Forwarder-Ziel
Recovery	1-hop beaconing und darauf aufbauende Berechnung eines planaren Subgraphen. Danach Unicast nach Rechte-Hand-Regel.
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.2: CBGR

Verfahren	Contention-based beaconless geographic routing (CBGR) [24]
Form des Weiterleitungsbereiches	60° Sektor vom Forwarder in Richtung des Ziels mit einem Radius r
Zählerberechnung	$D_{Max} * \left(1 - \frac{FP}{2r}\right)$ FP = Abstand des Forwarders zum in Richtung des Zielknotens liegenden Schnittpunkts zwischen dem Kreis des Sendebereichs des betrachteten Knoten und der Linie Forwarder-Zielknoten r = Radius des Sendebereiches
Recovery	One-hop beaconing und Unicast des Datenpakets an den dem Zielknoten nächsten Knoten aus der Nachbarschaft.
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.3: BGR

Verfahren	Blind geographic routing (BGR) [25]
Form des Weiterleitungsbereiches	Implementationsabhängig
Zählerberechnung	$Max_{Delay} * \left(1 - \frac{d-c}{r}\right)$ $d =$ Abstand Forwarder-Ziel $c =$ Abstand des Kandidatenknotens zum Ziel $r =$ Radius des Sendebereiches
Recovery	Verschieben des Weiterleitungsbereiches um 60° und Änderung der Methode zur Zählerberechnung: c wird auf den größten Abstand zwischen dem Zielknoten und jedem Knoten innerhalb des verschobenen Weiterleitungsbereiches gesetzt.
Besonderheiten	Avoidance of simultaneous forwarding (ASF): Hop count wird in den Header eines jeden Pakets geschrieben, um Problemen durch mehrere gleichzeitig weiterleitende Knoten vorzubeugen.
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.4: EABGR

Verfahren	Energy aware beaconless geographic routing in three dimensional WSNs (EABGR) [26]
Form des Weiterleitungsbereiches	60° Kegel mit dem Forwarder als Spitze und dem Schnittpunkt der Linie Forwarder-Zielknoten mit der Kegelebene als Mittelpunkt der Kegelebene
Zählerberechnung	$t = Max_{Delay} * \frac{\left(\frac{r-p}{r}\right)^2}{e}$ $r =$ Radius des Sendebereiches $p =$ Abstand des Forwarders zur Projektion des betrachteten Knotens auf die Linie Forwarder-Ziel $e =$ Energie des Knotens
Recovery	Forwarding Area Shift
MAC-cross-layer	nein
Tags	unicast, aggressive contention, 3D

Tabelle A.5: BMR

Verfahren	Beaconless multihop routing (BMR) [29]
Allg. Ablauf	Alle Knoten, die Nachrichten von der Basisstation, d.h. des Zielknotens, hören, sind potentielle Forwarder. Das Gebiet um einen Zielknoten wird in kreisförmige Zonen aufgeteilt und eine Nachricht wird von Zone zu Zone in Richtung Ziel weitergeleitet.
Zählerberechnung	$Timer(Scheduling) = \alpha * ZoneID + \beta * \lfloor \frac{IE}{CE} \rfloor$ $\alpha, \beta =$ Gewichtungen ZoneID = ZonenID eines Knotens; je kleiner diese ist, desto näher befindet sich der Knoten an der Basisstation IE = initial verfügbare Energie des Knotens CE = aktuell verfügbare Energie des Knotens
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.6: BLR-E

Verfahren	Beaconless Location-Based Routing with Signal Strength Assisted for Ad-Hoc Networks [32]
Form des Weiterleitungsbereiches	Sektor, Reuleaux Dreieck oder Kreis
Zählerberechnung	$Max_{delay} * \frac{r-p}{r} * f(s)$ $r =$ Radius des Sendebereiches $p =$ Fortschritt $d =$ Abstand zur Linie Forwarder-Ziel $f(x) =$ Funktion, die Knoten in Gebieten mit höherer Knotendichte einen kleineren Wert zuordnet als Knoten in Gebieten mit niedrigerer Knotendichte $s =$ Signalstärke in dBm
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.7: DAF

Verfahren	Distance-Aware Forwarding (DAF) [33]
Form des Weiterleitungsbereiches	Sektor, Reuleaux Dreieck oder Kreis; im Paper wird die Sektor-Form benutzt
Zählerberechnung	$d_A = \frac{r-y}{r} * d_M$ r = Radius des Sendebereiches y = Abstand des Kandidatenknotens zur Strecke zwischen Forwarder und Ziel d_M = maximale Verzögerung (Max Delay)
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.8: DBO

Verfahren	Directed Broadcast with Overhearing (DBO) [34]
Form des Weiterleitungsbereiches	-
Zählerberechnung	kein Zähler; Knoten bewerben sich für Sendekanal nach IEEE 802.15.4
Recovery	-
MAC-cross-layer	ja
Tags	unicast, aggressive contention

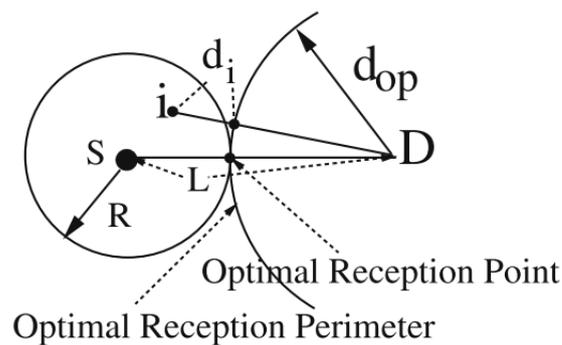
**Abbildung A.1:** Optimal Reception Perimeter in ABC. Bild entnommen aus [27].

Tabelle A.9: ABC

Verfahren	ABC [27]
Form des Weiterleitungsbereiches	Sektor (mit implementationsabhängigem Winkel)
Zählerberechnung	<p>Kleinerer Zähler bei niedrigerer Distanz zum Ziel.</p> $d_i = d(N_i, N_D) - d_{op}$ <p>d_i = Distanz des Knotens i zum „optimal reception perimeter“ (Siehe Abb. A.1) $d(N_i, N_D)$ = Distanz des Knotens i zum Ziel D d_{op} = $L - R$ = Distanz von Ziel D zum „optimal reception perimeter“ L = Distanz Forwarder - Ziel R = Radius des Sendebereiches</p> <p>Sendebereich wird in Relay Zones ($6 * 60^\circ$ Sektoren) aufgeteilt; Relay Zone in Richtung des Ziels wird in Streifen aufgeteilt, deren Breite von d_i abhängen; Knoten im selben Streifen haben etwa die gleiche Distanz zum Ziel; jedem Streifen werden bestimmte Zeitslots zum Senden von Antworten zugeordnet</p>
Recovery	Befindet sich der Empfänger einer Nachricht nicht in der ersten Relay Zone des Forwarders, wurde die Nachricht in eine andere Relay Zone (entspricht einem Forwarding Area Shift) weitergeleitet, also nicht in Richtung Ziel. In diesem Fall wird ein Zähler inkrementiert. In jedem anderen Fall wird der Zähler zurückgesetzt. Die weiteren Nachrichten werden mit zusätzlichen Informationen zu den Knoten am Rande des Void Gebietes erweitert. Die Nachricht wird um das Void Gebiet von Knoten zu Knoten herumgereicht, bis ein Knoten eine Weiterleitung des von ihm versendeten Paketes aus der ersten (greedy) Relay Zone hört.
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.10: RBRR

Verfahren	Receiver-based Beaconless Real-time Routing Protocol (RBRR) [35]
Form des Weiterleitungsbereiches	Reuleaux Dreieck
Zählerberechnung	<p>Der Weiterleitungsbereich wird, wie in BOSS, in Untergebiete aufgeteilt.</p> $T_{wait} = \left(Z_n * \frac{D_{MAX}}{N_{tz}} \right) + rand \left(\frac{D_{MAX}}{N_{tz}} \right)$ $Z_n = \lfloor N_{tz} * \left(1 - \frac{d(s,d)-d(n,d)}{r} \right) \rfloor$ <p>Z_n = Index der Zone, in der sich ein Knoten n befindet N_{tz} = Anzahl Sub-Time Zones $d(c,d)$ = euklidische Distanz der Knoten c und d s = Forwarder d = Ziel n = Kandidatenknoten D_{MAX} = maximale Verzögerungszeit rand(x) = Zufallswert zwischen 0 und x</p>
Speed	<p>Gewünschte Auslieferungsgeschwindigkeit:</p> $S_{setspeed} = \frac{D(source,sink)}{T_{setdeadline}}$ <p>$D(source, sink)$ = Distanz von Quell- zu Zielknoten $T_{setdeadline}$ = Zeitpunkt, zu dem das Datenpaket den Zielknoten erreicht haben soll</p> <p>Der speed-Wert eines beliebigen Knoten entspricht dem räumlichen Fortschritt zum nächsten Knoten geteilt durch die Verzögerung, die durch Weiterleiten des Pakets zum nächsten Knoten auftritt.</p>
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.11: IFSA

Verfahren	Improved forwarder selection approach [30]
Form des Weiterleitungsbereiches	Vier mögliche drei-dimensionale Formen: Kegel mit 60° Innenwinkel, Kegel mit 120° Innenwinkel (nicht empfohlen), parabolischer Kegel, elliptischer Kegel
Zählerberechnung	$t = MaxDelay * \frac{\left(\frac{Radius - XForward}{Radius}\right)^2 * angle}{ResidualEnergy}$ <p>Radius = Radius des Sendebereiches eines Knotens XForward = Distanz von Forwarder und Projektion des Kandidatenknotens auf die Strecke zwischen Forwarder und Ziel angle = Winkel der Strecke von Forwarder zu Ziel und der Strecke von Forwarder zu Kandidatenknoten ResidualEnergy = verbleibende Energie des Kandidatenknotens</p>
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.12: CNB-GEDIR

Verfahren	Cooperated no-beacon GEDIR (CNB-GEDIR) [28]
Form des Weiterleitungsbereiches	Greedy Area
Zählerberechnung	$T = \alpha(MM_d - (M_iM_d - d_{max}))$ <p>α = implementationsabhängig M_i = Position des Forwarders M_d = Position des Zielknotens M = betrachteter Kandidatenknoten d_{max} = maximaler Senderadius oder Senderadius, der nötig ist, um M_d zu erreichen und kleiner als der maximale Senderadius ist</p>
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

Tabelle A.13: MRR

Verfahren	Multipath Routeless Routing Protocol (MRR) [36]
Form des Weiterleitungsbereiches	Rechteck mit festgelegter Breite von Forwarder bis Zielknoten.
Zählerberechnung	$Max_{delay} \left(w_1 \frac{ RS ^\delta - r_{ij} ^\delta}{ RS ^\delta} + w_2 \frac{\alpha^\delta - \beta^\delta}{\alpha^\delta} + w_3 \frac{E_{init}^\delta - E_j^\delta}{E_{init}^\delta} \right)$ <p> RS = Empfänger Empfindlichkeit in dBm r_{ij} = Signalstärke in dBm, empfangen durch Knoten j von Knoten i α = Strecke zwischen dem Forwarder, der die Weiterleitung gestartet hat und dem Zielknoten β = Distanz von Knoten j zur Strecke α E_{init} = initial verfügbare Energie eines Knotens j δ = Konfigurationskonstante w_1, w_2, w_3 = Gewichtungen mit der Summe 1 </p>
Recovery	-
MAC-cross-layer	nein
Tags	unicast, aggressive contention

A.1.2 Non-aggressive Contention

Tabelle A.14: SIF

Verfahren	State-free implicit forwarding (SIF) [37]
Zählerberechnung	<p>Beispielhaft:</p> $C = [W_d * (1 - \frac{L}{T}) + W_e * (1 - \frac{R_e}{E}) + W_r * V] * M$ <p>C = Verzögerungswert L = Abstand Sender-Senke (Ziel) T = Sendereichweite des Senders R_e = verbleibende Energie E = maximale Energie V = Zufallswert zwischen 0 und 1 W_d, W_e, W_r = Gewichtungen für Distanz, Energie und Zufallswert $W_d + W_e + W_r = 1$ M = maximaler Verzögerungswert</p>
Recovery	Wird kein CTS erhalten, wird RTS erneut verschickt; falls das mehr als Threshold mal passiert, wird die Sendereichweite erhöht, bis ein Knoten gefunden wird; sonst wird in den inaktiven Modus gewechselt
MAC-cross-layer	ja
Tags	unicast, non-aggressive contention

Tabelle A.15: OGF

Verfahren	On-demand Geographic Forwarding (OGF) [38]
Zählerberechnung	<p>Beispielhaft:</p> $C = \left[W_d * \left(1 - \frac{L}{T} \right) + W_e * \left(1 - \frac{R_e}{E} \right) + W_r * V \right] * M$ $N_{slot} = \lfloor C_{abs} / t_{slot} \rfloor$ <p>C = Verzögerungswert L = Abstand Sender-Senke (Ziel) T = Sendereichweite des Senders R_e = verbleibende Energie E = maximale Energie V = Zufallswert zwischen 0 und 1 W_d, W_e, W_r = Gewichtungen für Distanz, Energie und Zufallswert $W_d + W_e + W_r = 1$ M = maximaler Verzögerungswert</p>
Recovery	<p>Partial Source Routing: Im Umkreis von zwei oder mehr Hops wird nach Knoten gesucht, die näher am Ziel sind. Passive Participation kann in Ausnahmefällen angewendet werden (z.B. wenn die Energiereserven eines Knotens nicht mehr für PSR reichen). In Passive Participation bewirbt sich ein Knoten, der keine geeigneten Nachbarn fürs Greedy Forwarding findet, nicht mehr als Kandidatenknoten.</p>
MAC-cross-layer	ja
Tags	unicast, non-aggressive contention

Tabelle A.16: GRoVar

Verfahren	GRoVar (Geographic Routing with Variable transmission range) [49]
Zählerberechnung	implementationsabhängig; in den Simulationen des Artikels: $(R - vt - ut)/R * T_{max}$ v = Forwarder t = Ziel u = Empfänger R = Senderadius von v
Recovery	Falls Greedy Forwarding (GF) fehlschlägt, wird die Sendereichweite erhöht und erneut GF angewandt; falls GF mit maximaler Sendereichweite fehlschlägt, wird Face Routing (FR; auf Gabriel Graph) mit initialer Sendereichweite ausgeführt; falls FR fehlschlägt, wird die Sendereichweite erhöht und erneut FR angewandt.
MAC-cross-layer	nein
Tags	unicast, non-aggressive contention

Tabelle A.17: IGF

Verfahren	Implicit Geographic Forwarding (IGF) [15]
Zählerberechnung	<p>CTS Response = SIFS + M * F</p> <p>SIFS = Short Inter Frame Spacing (From 802.11)</p> <p>M = maximale zusätzliche Wartezeit (es gilt: M+SIFS < DIFS)</p> $F = 1 - ((D_{Parameter} + E_{Parameter} + Rand) / (W_D + W_E + W_R))$ $D_{Parameter} = C_D * IDTD / R$ $E_{Parameter} = C_E * ER$ <p>Rand = Zufallszahl zwischen 0 und 1</p> <p>W_D, W_E, W_R: Gewichtungen für Distanz, Energie und Zufallszahl</p> <p>IDTD = Increase in Distance Toward the Destination (Zunahme der Entfernung in Richtung Ziel)</p> <p>R = Sendereichweite der Knoten</p> <p>ER = 1 - vom Empfänger verwendete Energie / maximale Energie eines Knotens</p>
Recovery	Forwarding Area Shift
MAC-cross-layer	ja
Tags	unicast, non-aggressive contention

Tabelle A.18: CBRR

Verfahren	Contention-based Beaconless Real-time Routing (CBRR) (OneHop Protokoll) [39]
Zähler	$P_i = (q_i + 1)(\alpha d_i / R_c + \beta E_i / E_t + \gamma(1 - q_i / Q_t) + \eta r_i)$ P_i = Verzögerungspriorität d_i = Fortschrittsabstand zur Senke R_c = Sendereichweite E_i = verbleibende Energie E_t = initial verfügbare Energie q_i = Anzahl Pakete in Warteschlange Q_t = gesamte Länge der Warteschlange r_i = Zufallswert zwischen 0 und 1 $\alpha + \beta + \gamma + \eta = 1$ (Gewichtungen) $T_{short} = SIFS + 0.5 * SIFS * (1 - P_j)$ $T_{long} = SIFS + SIFS * (1 - P_j)$ SIFS: Short Inter Frame Spacing (10 μ s in IEEE 802.11 standard)
Speed	Relay Speed (für einen beliebigen Knoten): $Speed_i^j = \frac{d(i,j)}{Delay_i^j}$ $d(i, j)$ = euklidische Distanz zwischen den Positionen der Knoten i und j $Delay_i^j$ = Verzögerung zwischen den Knoten i und j beim Senden eines Pakets von i nach j, gemessen am Zeitpunkt des Erhalts des Datenpakets zur Weiterleitung und des Zeitpunkts des Erhalts der ACK-Nachricht zur Bestätigung der Weiterleitung durch den nächsten Hop Required Speed (muss mindestens von jedem Knoten erreicht werden): $Speed_i^D = \frac{d(i,D)}{Deadline(D) - T_{now}}$ D = Position des Ziels $Deadline(D)$ = spätester Zeitpunkt, zu dem das Ziel das Paket erhalten haben muss T_{now} = aktueller Zeitpunkt
MAC	nein
Tags	unicast, non-aggressive contention

Tabelle A.19: GDBF

Verfahren	Guaranteed Delivery Beaconless Forwarding (GDBF) [17]
Zählerberechnung	Greedy Timer = $Max_{Timeout} * ((r + d - SD)/ SD)$ r = Senderadius S = Forwarder D = Ziel d = Abstand Knoten-Ziel
Recovery	andere Timerberechnung; Knoten nah am aktuellen Forwarder werden bevorzugt: Recovery Timer = $Max_{Timeout} * (u/r)$ u = Abstand Kandidatenknoten-Forwarder siehe auch Kap. 5.2
MAC-cross-layer	nein
Tags	unicast, non-aggressive contention

Tabelle A.20: BLGF

Verfahren	CoopGeo beaconless greedy forwarding (BLGF) [40]
Weiterleitungsbereich	Gesamter Sendebereich des Forwarders wird in zwei Gebiete unterteilt: Positive Progress Area (enthält alle Knoten, die näher am Ziel sind als der Forwarder) und Negative Progress Area (enthält alle anderen Knoten im Sendebereich)
Zähler	$T_{CBF} = (CSA * \frac{T_{Max}}{NSA}) + rand(\frac{T_{Max}}{NSA})$ $CSA_{PPA} = \lfloor NSA * \frac{r - (d_{S,D} - d_{F_i,D})}{2r} \rfloor$ (für Knoten in PPA) $CSA_{NPA} = \lfloor \left(\frac{\sqrt{n} * d_{S,F}}{r}\right)^2 \rfloor + \frac{NSA}{2}$ (für Knoten in NPA) NSA = Anzahl Untergebiete rand(x) = Zufallswert zwischen 0 und x r = Senderadius $(d_{S,D} - d_{F_i,D})$ = Fortschritt des Kandidatenknotens $n = \frac{NSA}{2}$
Recovery	CoopGeo beaconless recovery forwarding (BLRF): Beaconless Forwarder Planarization [18]; verwenden anderer Timerberechnung; auswählen des nächsten Hop per SELECT-ACK-Verfahren
MAC	ja
Tags	unicast, non-aggressive contention

Tabelle A.21: BOSS

Verfahren	Beacon-less On Demand Strategy for Geographic Routing in Wireless Sensor Networks (BOSS) [41]
Form des Weiterleitungsbereiches	Releaux Dreieck aus Sicht des Zielknotens. Das Dreieck wird zusätzlich in konzentrische Schichten (Untergebiete) unterteilt.
Zählerberechnung	$T = \left(CSA \times \frac{T_{Max}}{NSA} \right) + random \left(\frac{T_{Max}}{NSA} \right)$ $CSA = \lfloor (NSA - 1) \times \frac{r - P(n,d,c)}{2r} \rfloor$ $P(n, d, c) = dist(c, d) - dist(n, d)$ $NSA = \text{Anzahl Untergebiete}$ $dist(a, b) = \text{euklidische Distanz zwischen den Positionen a und b}$ $r = \text{Radius des Sendebereiches}$ $P(n, d, c) = \text{Fortschritt des Knotens n in Abhängigkeit zum Forwarder c und zum Zielknoten d}$
Recovery	Automatisches Sammeln der Positionsinformationen aller Nachbarknoten, falls der Weiterleitungsbereich leer ist; auf dieser Basis wird implementationsabhängig ein planarer Subgraph berechnet, auf dem ein implementationsabhängiges Routing ausgeführt wird, z.B. GFG.
MAC-cross-layer	Nein
Tags	unicast, non-aggressive contention

Tabelle A.22: EBGR

Verfahren	Energy-efficient Beaconless Geographic Routing (EBGR) [45]
Zähler	$\frac{\gamma * r_s(u)}{\sqrt{n}} * (2 \sum_{i=1}^m \sqrt{i} - \sqrt{m} - 1) + \gamma * (vf_u - \sqrt{\frac{m-1}{n}} r_s(u))$ <p> n = Anzahl Coronas γ = Verzögerung, um ein Paket über eine Unit-Einheit zu schicken u = betrachteter Knoten (wir suchen das next-hop Relay dieses Knotens) v = ein Knoten in der Relay Search Area von Knoten u (ein Kandidatenknoten) $r_s(u) \leq uf_u = d_0$ f_u = optimal next-hop relay position $m = \lfloor \left(\frac{\sqrt{n} * vf_u }{r_s(u)} \right)^2 \rfloor$ </p>
Recovery	Beaconless Recovery: Weiterleitung gegen den Uhrzeigersinn zu nächstem Gabriel Graph-Nachbarn
MAC	ja
Tags	unicast, non-aggressive contention

Tabelle A.23: EBGRES

Verfahren	Energy-efficient Beaconless Geographic Routing with Energy Supply (EBGRES) [46]
Form des Weiterleitungsbereiches	Kreis um die optimale Position des nächsten Hops. Der Kreis wird in konzentrische Koronas unterteilt.
Zählerberechnung	$\frac{\gamma * r_s}{\sqrt{n}} * (2 \sum_{i=1}^m \sqrt{i} - \sqrt{m} - 1) + \gamma * (v f_u - \sqrt{\frac{m-1}{n}} r_s(u))$ $m = \lfloor \left(\frac{\sqrt{n} v f_u }{r_s(u)} \right)^2 \rfloor + 1$ <p> γ = benötigte Zeit, um ein Paket über eine Unit-Einheit zu senden $r_s(u)$ = Radius des Kreises mit Mittelpunkt u, der die Relay Search Region um u darstellt f_u = ideale Position des nächsten Hops bei aktuellem Forwarder u v = Knoten aus der Relay Search Region von u u = ausführender Knoten </p>
Recovery	Automatisches Sammeln der Positionsinformationen aller Nachbarknoten, falls der Weiterleitungsbereich leer ist; auf dieser Basis wird implementationsabhängig ein planarer Subgraph berechnet, auf dem Face-Routing angewendet wird.
MAC	ja
Tags	unicast, non-aggressive contention

Tabelle A.24: EEGR

Verfahren	Energy-Efficient Geographic Routing (EEGR) [47]
Form des Weiterleitungsbereiches	„Relay Search Region“: Kreisförmiger Weiterleitungsbereich mit optimaler Weiterleitungsposition als Mittelpunkt. Der Radius des Bereiches darf höchstens dem Abstand vom Forwarder zur optimalen Weiterleitungsposition entsprechen.
Zählerberechnung	$\delta_{w \rightarrow p} = \gamma w f_p $ <p> $w f_p$ = Abstand vom Kandidatenknoten w zur optimalen Position f_p γ = empirisch festgelegte Konstante </p>
Recovery	-
MAC-cross-layer	nein
Tags	unicast, non-aggressive contention

Tabelle A.25: MEGAN

Verfahren	Mobility assisted Energy efficient Georouting in energy harvesting Actuator and sensor Networks (MEGAN) [48]
Zählerberechnung	$T_v = \frac{E_{max}}{E_v(t+k)}$ $E_v(t+k) = E_v(t) - E_{mov_v}(\ v - g_u\)$ $E_{max} = \text{maximale Energie eines jeden Knoten}$ $E_v(t) = \text{Energie des Knotens } v \text{ zum Zeitpunkt } t$ $k = \text{Zeiteinheiten, die der Kandidatenknoten benötigt, um sich zu } g_u \text{ zu bewegen}$ $E_{mov_v} = \text{Energie, die Knoten } v \text{ benötigt, um sich zur Position } g_u \text{ zu bewegen}$ $g_u = \text{ideale Position des nächsten Hops bei aktuellem Forwarder } u$ $v = \text{Kandidatenknoten}$
Recovery	-
MAC	nein
Tags	unicast, non-aggressive contention

Tabelle A.26: PF

Verfahren	Pizza-Forwarding (PF) [51]
Form des Weiterleitungsbereiches	Der kreisförmige Sendebereich um einen Knoten wird in sechs Sektoren aufgeteilt, welche jeweils unterschiedliche Weiterleitungsprioritäten besitzen.
Zählerberechnung	<p>Für Sektor 1 und 2: $delay = \left(\frac{1}{p} + \frac{\Theta}{\frac{\pi}{4}}\right) * T_{max} + Rand()$ $0 \leq \Theta \leq \frac{\pi}{4}$</p> <p>Für Sektor 3 und 4: $delay = \left(\frac{d-p}{d} + \frac{\Theta}{2*\pi}\right) * T_{max} + Rand()$ $\frac{\pi}{4} \leq \Theta \leq \frac{\pi}{2}$ Θ = Winkel zwischen den Strecken Forwarder-Kandidatenknoten und Forwarder-Zielknoten p = Fortschritt: Distanz von Forwarder zur Projektion des Kandidatenknotens auf die Strecke zwischen Forwarder und Ziel $Rand()$ = Zufallswert</p>
Recovery	<p>Knoten in der Recovery Area (Sektor 3' und 4') benutzen andere Delays. Entweder $delay = \left[1.25 + \left(\frac{\Theta}{2*\pi} - \frac{1}{4}\right) + \frac{N-SDA}{N}\right] * T_{max} + Rand()$ oder $delay = \left(1.25 + \frac{1}{\Delta d}\right) * T_{max}$ $\Delta d = D_{PH-Node} * \sin(\pi - \Theta)$ $D_{PH-Node}$ = Distanz Forwarder-Betrachteter Knoten SDA = Same Distance Area (implementationsabhängig) N = Recovery Area wird in N Subsets von Nachbarn unterteilt (implementationsabhängig) Nur die erste Delay-Funktion wurde in Simulationen verwendet.</p>
MAC	nein
Tags	unicast, non-aggressive contention

Tabelle A.27: PFMAC

Verfahren	Pizza- Forwarding Medium Access Control (PFMAC) [52]
Beschreibung	MAC-cross-layer-Variante des PF-Protokolls [51].
Tags	unicast, non-aggressive contention

Tabelle A.28: BGM

Verfahren	Beacon-less Geographic Multipath routing protocol (BGM) [53]
Form des Weiterleitungsbereiches	Der kreisförmige Sendebereich um einen Knoten wird mit Kurven in vier Untergebiete (Knotensets) Noc, Nic, Nof und Nif aufgeteilt.
Zählerberechnung	$DFD_e = Maxdelay \left(\omega_1 * \frac{a}{3} + \omega_2 * \frac{ce}{sd} + \omega_3 * rand \right)$ $DFD_g = Maxdelay \left(\omega_1 * \frac{a}{3} + \omega_2 * \frac{cd}{r+sd} + \omega_3 * rand \right)$ $\omega_i =$ Gewichtungen $\omega_1 + \omega_2 + \omega_3 = 1$ $a \in 1, 2, 3 =$ abhängig vom Szenario und Knotenset, in dem sich der Knoten befindet $cd =$ Distanz vom betrachteten Knoten zum Zielknoten $ce =$ Distanz vom betrachteten Knoten zur „boundary curve“ $sd =$ Distanz vom Quellknoten zum Zielknoten $r =$ Radius des Sendebereiches $rand =$ Zufallswert zwischen 0 und 1 Knoten, die zum Noc-Knotenset gehören, benutzen im „Boundary Forwarding Mode“ DFD_e zur Timerberechnung. Sonst wird DFD_g benutzt.
Recovery	-
MAC	nein
Tags	unicast, non-aggressive contention

Tabelle A.29: GeRaF

Verfahren	Geographic Random Forwarding (GeRaF) [42]
Form des Weiterleitungsbereiches	Der Weiterleitungsbereich umfasst den gesamten Teil des Sendebereichs des Forwarders, der sich näher am Ziel befindet als der Forwarder selbst. Dieser Bereich wird in Zonen unterteilt, die abhängig von ihrer Nähe zum Ziel sortiert werden.
Zählerberechnung	Jeder Zone wird ein implementationsabhängiger Zeitabschnitt zugeordnet, in dem Knoten der Zone dem Forwarder antworten dürfen. Der Forwarder kontrolliert mit verschiedenen Nachrichtentypen, ob Zonen übersprungen werden sollen oder ob Knoten einer Zone mehrmals antworten sollen.
Recovery	-
MAC	ja
Tags	unicast, non-aggressive contention

Tabelle A.30: MACRO

Verfahren	MAC/Router protocol (MACRO) [50]
Form des Weiterleitungsbereiches	Gesamter Sendebereich des Forwarders
Zählerberechnung	$G_{R',R''} = \frac{d(R',D) - d(R'',D)}{P}$ $G_{R',R''}$: Fortschritt zu Zielknoten D pro Einheit verwendeter Sendestärke R' = Forwarder R'' = Kandidatenknoten P = von R' verwendete Sendestärke, um das Paket an R'' senden zu können $d(A, B)$ = Distanz zwischen Knoten A und Knoten B
Recovery	-
MAC-cross-layer	ja
Tags	unicast, non-aggressive contention

Tabelle A.31: PSGR

Verfahren	Priority-based Stateless Geo-routing (PSGR) [43]
Zähler	(ATI ist implementationsabhängig) $AckT_i = \begin{cases} AckT_{i-1} + ATI_i, & 1 \leq i \leq Z \\ 0, & i = 0 \end{cases}$ $AckT_i$ = Acknowledgement Delay eines Knotens in Forwarding Zone i (FZ_i) ATI_i = Acknowledgement Timer Interval
Recovery	Stateless Rebroadcast oder Stateless Bypass
MAC	nein
Tags	unicast, non-aggressive contention

A.1.3 Hybrid Contention

Tabelle A.32: CBF

Verfahren	Contention-based forwarding scheme (CBF) [54]
Form des Weiterleitungsbereiches	Abhängig vom verwendeten Unterdrückungsschema. Im „Area-based Suppression“-Unterdrückungsschema kann wahlweise Total (alle Nachbarknoten), Release Dreieck oder Kreis als Form benutzt werden. In allen anderen Schemata wird Total verwendet.
Hybrid contention	Das „Active Suppression“-Unterdrückungsschema benutzt non-aggressive contention, die anderen Unterdrückungsschemata benutzen aggressive contention.
Zählerberechnung	$t(P) = T(1 - P)$ $P(f, z, n) = \max \left\{ 0, \frac{\text{dist}(f,z) - \text{dist}(n,z)}{r_{\text{radio}}} \right\}$ T = maximales forwarding delay P = Fortschritt f = Position des Forwarders z = Position des Ziels n = Position des betrachteten Knotens dist = euklidische Distanz zwischen zwei Positionen r = Radius des nominalen Sendebereiches
Recovery	Nur im „Area-based Suppression“-Unterdrückungsschema. Zusammen mit dem gewählten Unterdrückungsbereich (Weiterleitungsbereich) wird der Fortschrittsbereich eines Knotens in drei Bereiche aufgeteilt. Wird kein Knoten im Unterdrückungsbereich gefunden, werden nacheinander die anderen beiden Bereiche nach Knoten abgesucht.
MAC-cross-layer	nein
Tags	unicast, hybrid contention

Tabelle A.33: HCGR

Verfahren	Hybrid Contention-Based Geographic Routing (HCGR) [21]
Form des Weiterleitungsbereiches	Der Weiterleitungsbereich ist in zwei Regionen unterteilt. Aggressive Area: 60° Sektor vom Forwarder in Richtung des Ziels mit einem Radius r Non-aggressive Area: Bereich im Sendebereich des betrachteten Knoten, der nur Positionen enthält, die sich näher am Ziel befinden als der betrachtete Knoten
Hybrid contention	„Aggressive Contention“ unterdrückt „Non-aggressive Contention“. Das Datenpaket wird immer statt einem Kontrollpaket abgeschickt. Knoten in der Aggressive Area leiten das Datenpaket nach Ablauf des Zählers sofort weiter. Ist die Aggressive Area leer und der Zähler eines Knotens in der Non-aggressive Area läuft ab, wird ein RESPONSE/SELECTION/DATA-Schema (also Non-aggressive Contention) benutzt.
Zählerberechnung	Greedy-Zähler: $gct = \left(\frac{\theta + \frac{r-p}{r}}{360} \right) \cdot T_{max}$ θ = Winkel betrachteter Knoten - Forwarder - Ziel r = Radius des Sendebereiches (identisch für alle Knoten) p = Distanz Forwarder - Zielknoten abzüglich der Distanz betrachteter Knoten - Zielknoten T_{max} = maximaler Delay-Wert Recovery-Zähler: $rct = \left(\frac{\sigma}{360} \right) \cdot T_{max}$ σ = Winkel zwischen der Strecke CI und der Strecke C - Kandidatenknoten (s. Abb. A.2)
Recovery	Verwendung anderer Timerberechnung
MAC	nein
Tags	unicast, hybrid contention

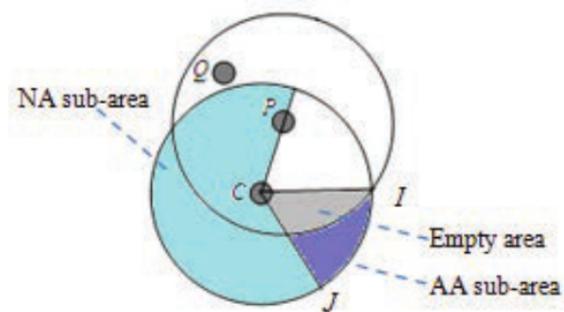


Abbildung A.2: Verzögerungsbereiche im Recovery Routing von HCGR [21]. C ist Forwarder, P ist der letzte Hop und Q ist der vorletzte Hop. I ist der Schnittpunkt der Kreise mit Radius r (uniformer Senderadius) um die Knoten P und C . Bild entnommen aus [21].

A.1.4 Recovery Routing

Tabelle A.34: CR

Verfahren	Clockwise Relaying (CR) [55]
Zählerberechnung	$Add_{delay} = Max_{delay} \frac{\alpha - 90^\circ}{360^\circ}$
Tags	recovery

Tabelle A.35: NB-FACE

Verfahren	No-Beacon FACE (NB-FACE) [2]
Zählerberechnung	$TW_i = \alpha(2\pi - \Theta)$ falls im Uhrzeigersinn weitergeleitet wird, $TW_i = \alpha\Theta$ sonst. $TW_{i+1} = 2\pi\Theta$ falls $M_{i+1} = M_{i-1}$ TW_i = Timeout (Zählerwert) des Knotens M_i M_{i-1} = vorletzter Hop M_i = Forwarder M_{i+1} = betrachteter Knoten α = keine Angabe Θ = Winkel $M_{i-1}M_iM_{i+1}$
Tags	recovery

Tabelle A.36: BFP/AR

Verfahren 1	Beaconless Forwarder Planarization (BFP) [18]
Verfahren 2	Angular Relaying [18]
Zählerberechnung 1	$t(d) = \frac{d}{r} * t_{max}$ d = Distanz dieses Knotens zum Forwarder r = Radius des Sendebereichs t_{max} = maximale Verzögerungszeit
Zählerberechnung 2	$t(\Theta) = \frac{\Theta}{2\pi} * t_{max}$ Θ = Winkel (je nach gewählter Regel für das Face-Routing wird er anders berechnet)
Tags	recovery

A.2 Multicast Routing

Tabelle A.37: GEM

Verfahren	Geographic Multicast (GEM) [63]
Baumberechnung	Euklidischer Steinerbaum
Average Progress Metric	$ap(v', v'', V^{(D)}) = cp(v', v'', V^{(D)})/ V^{(D)} $ $cp(v', v'', V^{(D)}) = cd(v', V^{(D)}) - cd(v'', V^{(D)})$ $cd(v, V^{(D)}) = \sum_{w \in V^{(D)}} d(v, w)$ $v, v', v'', w = \text{Knoten}$ $V^{(D)} = \text{Menge aller Zielknoten}$ $ M = \text{Kardinalität der Menge M}$ $d(v, w) = \text{Distanz der Position eines Knotens v zur Position eines Knotens w}$
Recovery	-
Tags	multicast routing

Tabelle A.38: DMPB

Verfahren	Distributed multicast protocol based on beaconless routing (DMPB) [64]
Baumberechnung	Jeder Forwarder teilt Netzwerk in vier Sektoren mit sich selbst als Zentrum ein und leitet jeweils ein Paket in den Sektor weiter, in dem sich das Ziel des Pakets befindet.
Zählerberechnung	implementationsabhängig
Recovery	implementationsabhängig
Tags	multicast routing

Tabelle A.39: BRUMA

Verfahren	BRUMA [62]
Baumberechnung	Minimaler Spannbaum nach Kruskal auf Basis der Positionsinformationen der Zielknoten und des Quellknotens.
Zählerberechnung	$t_f = f_i * T_{Branch} + (CSA_f * \frac{T_{Branch}}{NSA}) + random(\frac{T_{Branch}}{NSA})$ $t_f = \text{Delay eines Knoten zur Kandidatur als nächster Hop für Knoten f}$ $T_{Branch} = \frac{T_{Greedy}}{ F }$ $T_{Greedy} = \text{Zeit in Sekunden, die ein Forwarder auf Response-Nachrichten wartet}$ $F = \text{Menge aller FLR (First Level Receiver)}$ $\text{First Level Receiver} = \text{ein Zielknoten, welcher der erste nächste Hop im Spannbaum eines Forwarders ist}$ $CSA_f = \lfloor NSA * \frac{R - P_g(n, f, s)}{R} \rfloor$ $CSA_f = \text{Nummer der Common Sub Area für Knoten f, in der sich ein Knoten befindet}$ $NSA = \text{Anzahl aller Common Sub Areas}$ $R = \text{maximaler Sendebereich eines jeden Knotens}$ $P_g(n, f, s) = dist(s, f) - dist(n, f)$ $n = \text{betrachteter Knoten (ein Nachbarknoten von s)}$ $f = \text{ein bestimmter FLR}$ $s = \text{Forwarder}$ $dist(a, b) = \text{euklidische Distanz zwischen den Positionen der Knoten a und b}$ $random(x) = \text{Zufallswert zwischen 0 und x}$
Recovery	<p>Knoten näher am Forwarder antworten zuerst und unterdrücken entferntere Knoten. Auf Basis der gesammelten Positionsdaten wird ein planarer Graph nach GFG [3] erstellt.</p> $t_p = T_{Greedy} + (CSA_P * \frac{T_{Branch}}{NSA} + random(\frac{T_{Branch}}{NSA}))$ $CSA_P = \lfloor NSA * \frac{P_p(n, s)}{R} \rfloor$ $P_p(n, s) = dist(s, n)$
Tags	multicast routing

A.3 VANET

A.3.1 Aggressive Contention

Tabelle A.40: RPBL

Verfahren	Routing Protocol with BeaconLess (RPBL) [66]
Zählerberechnung	$BT = \frac{R-c}{R}$ $BT = \frac{c}{R}$ <p>c = Distanz vom Kandidatenknoten zu seiner Projektion auf die Strecke zwischen Forwarder und temporärem Ziel R = Kommunikationsreichweite</p> <p>Die erste Gleichung wird benutzt, wenn der Abstand vom Kandidatenknoten zum temporären Ziel größer als R ist; sonst wird die zweite Gleichung benutzt.</p>
Recovery	- (löscht das Paket)
Tags	VANET, aggressive contention

Tabelle A.41: CBFS

Verfahren	Contention-based forwarding for street scenarios (CBFS) [69]
Zählerberechnung	$t = \begin{cases} \tau \left(1 - \left(\frac{p_i}{p_{max}} \right) \right) & 0 \leq p_i \leq p_{max} \\ \infty & sonst \end{cases}$ <p>τ = Maximale Verzögerungszeit p_i = Abstand des Forwarders zum Ziel abzüglich des Abstandes des Kandidatenknotens zum Ziel p_{max} = Radius des Sendebereiches</p>
Recovery	-
Tags	VANET, aggressive contention

A.3.2 Non-aggressive Contention

Tabelle A.42: BRAVE

Verfahren	Beacon-less routing algorithm for vehicular environments (BRAVE) [67]
Zählerberechnung	s. BOSS in Tab. A.21
Recovery	Store-carry-forward: Der Knoten wartet, bis sich andere Knoten mit besserem Fortschritt als er selbst in Reichweite befinden. Für diese Recovery-Strategie müssen allerdings Beacon implementiert werden, welche aber nicht für sonstige Weiterleitungsentscheidungen benutzt werden.
Tags	VANET, non-aggressive contention

Tabelle A.43: CBBPF

Verfahren	Contention-based Beaconless Packet Forwarding (CBBPF) [71]
Zählerberechnung	s. BOSS in Tab. A.21
Recovery	Store-carry-forward: Der Knoten wartet, bis sich andere Knoten mit besserem Fortschritt als er selbst in Reichweite befinden. Für diese Recovery-Strategie müssen allerdings Beacon implementiert werden, welche aber nicht für sonstige Weiterleitungsentscheidungen benutzt werden.
Tags	VANET, non-aggressive contention

Tabelle A.44: IB

Verfahren	Intelligent Beaconless geographical forwarding protocol (IB) [68]
Zählerberechnung	<p>An einer Kreuzung:</p> $f(GF_i, D_i, P_i) = X * GF_i^{\beta_1} * D_i^{\beta_2} * P_i^{\beta_3} + Y_{max}$ <p>X = die variablenabhängige Gewichtung der begrenzenden Bedingung $GF_i = \frac{D_c}{D_i}$ β_i = Gewichtung D_c = Abstand des Forwarders zum Zielknoten D_i = Abstand des Kandidatenknotens n_i zum Zielknoten P_i = Stärke des Signals, über das Kandidatenknoten n_i eine RTS-Nachricht des Forwarders empfangen hat</p> <p>Zwischen zwei Kreuzungen:</p> $g(P_i, D_i) = A * P_i^{\gamma_1} * D_i^{\gamma_2} + B_{max}$ $A = \frac{-B_{max}}{P_i^{\gamma_1} * D_i^{\gamma_2}}$ <p>P_i = Stärke des Signals, über das Kandidatenknoten n_i eine RTS-Nachricht des Forwarders empfangen hat D_i = Bewegungsrichtung des Kandidatenknotens n_i γ_1, γ_2 = Gewichtungen für P und D B_{max} = maximale Verzögerungszeit nach Erhalt des RTS Fra- me</p>
Recovery	-
Tags	VANET, non-aggressive contention

Anhang B

Inhalte der CD-ROM

Der nachfolgenden Liste kann die Dateistruktur der beiliegenden CD-ROM entnommen werden:

- VivasEstevao2014.pdf – Die Arbeit im PDF-Format.
- Arbeit – Enthält alle in der Arbeit verwendeten Dateien.
 - abstract.tex
 - agasthesis.cls
 - alphadin.bst
 - appendix.tex
 - biblio.bib
 - bibtexDB.pl
 - Einleitung.tex
 - fazit.tex
 - getBibtex.pl
 - Grundlagen.tex
 - Inhalte-der-CD-Rom.tex
 - Makefile
 - makePDF.bat – Erzeugt die Arbeit im PDF-Format.
 - math.sty
 - Methoden-und-Werkzeuge.tex
 - Multicast-Routing.tex
 - thesis.bbl

- thesis.tex
- titlepage.tex
- Topologiekontrolle.tex
- Unicast-Routing.tex
- VANET.tex
- images – Enthält alle in der Arbeit verwendeten Bilddateien.
 - * ABC.PNG
 - * AggCon-Winkel.PNG
 - * ARROW.PNG
 - * FaceRouting.PNG
 - * FaceRouting-Neu-ausNBFACE.PNG
 - * ForwardingAreaImage.PNG
 - * ForwardingAreaImage2 (aus dem Cao Xie Paper).PNG
 - * Geocast1.PNG
 - * Geocast2.PNG
 - * GraphGG.png
 - * GraphPDT.png
 - * HCGR-Recovery.PNG
 - * MRR.png
 - * MulticastRouting.PNG
 - * MulticastRouting-BadRoute.PNG
 - * MulticastRouting-GoodRoute.PNG
 - * NB-FACE.png
 - * OptimalRelayPosition.PNG
 - * Recovery-CR.PNG
 - * Recovery-PSGR.png
 - * Recovery-SC.PNG
 - * Recovery-TT.PNG
 - * RecoveryVoidArea.png
 - * Topologiekontrolle.png
 - * UnicastRouting-GeographischesRouting.PNG
 - * UnicastRouting-Nachbarschaft.PNG
 - * Uni-Logo.eps
 - * Uni-Logo.pdf

- * VANET-BRAVE.PNG
- * yEDsmall.png
- * geogebra – Enthält *.ggb-Dateien, welche, in Bilddateien konvertiert, in der Arbeit verwendet wurden.
 - AggCon-Winkel.ggb
 - GeographischesRouting.ggb
 - Multicast.ggb
 - Multicast-2.ggb
 - Nachbarschaft.ggb
 - NB-Face.ggb
 - Recover-CR.ggb
 - Recovery.ggb

Literaturverzeichnis

- [1] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks*. John Wiley and Sons, 2010.
- [2] M. Narasawa, M. Ono, and H. Higaki, “Nb-face: No-beacon face ad-hoc routing protocol for reduction of location acquisition overhead,” in *7th International Conference on Mobile Data Management (MDM’06)*, 2006, pp. 102–106.
- [3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” in *In Proceedings of 3rd international workshop on discrete algorithms and methods for mobile computing and communications*, 1999, pp. 48–55.
- [4] M. Heissenbüttel, T. Braun, M. Walchli, and T. Bernoulli, “Optimized stateless broadcasting in wireless multi-hop networks,” in *In Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM), 2006*, 2006, pp. 1–12.
- [5] B. Garbinato, A. Holzer, and F. Vessaz, “Six-shot broadcast: A context-aware algorithm for efficient message diffusion in manets,” in *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE*, 2008, pp. 625–638.
- [6] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen, “A power-aware broadcasting algorithm,” in *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2006, pp. 1–5.
- [7] Q. Huang, Y. Bai, and L. Chen, “Efficient lightweight broadcasting protocols for multi-hop ad hoc networks,” in *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2006, pp. 1–5.
- [8] M.-T. Sun, W. Feng, and T.-H. Lai, “Location aided broadcast in wireless ad hoc networks,” in *GLOBECOM’01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, 2001, pp. 2842–2846.

- [9] M. Gosnell, R. Albarelli, M. Cheng, and B. McMillin, “Energy balanced broadcasting through delayed intelligence,” in *International Conference on Information Technology: Coding and Computing (ITCC’05) - Volume II*, vol. 2, 2005, pp. 627–632.
- [10] F. García-Nocetti, F. J. Ovalle-Martínez, J. Solano-González, and I. Stojmenovic, “The impact of delay in dominating set and neighbor elimination based broadcasting in ad hoc networks,” in *Proceedings of the 6th International Conference on Ad-hoc, Mobile and Wireless Networks*, 2007, pp. 115–128.
- [11] F. J. Ovalle-Martínez, A. Nayak, I. Stojmenovic, J. Carle, and D. Simplot-Ryl, *Area Based Beaconless Reliable Broadcasting in Sensor Networks*, 2006, pp. 140–151.
- [12] A. Durrezi, V. Paruchuri, S. Iyengar, and R. Kannan, “Optimized broadcast protocol for sensor networks,” pp. 1013–1024, 2005.
- [13] S. Panichpapiboon and G. Ferrari, “Irresponsible forwarding,” in *2008 8th International Conference on ITS Telecommunications*, 2008, pp. 311–316.
- [14] A. Khan, “Beacon-less broadcast protocol for vehicular ad hoc networks,” in *2013 19th Asia-Pacific Conference on Communications (APCC)*, 2013, pp. 153–154.
- [15] B. Blum, T. He, S. Son, and J. Stankovic, “Igf: A state-free robust communication protocol for wireless sensor networks,” Department of Computer Science, University of Virginia, USA, Tech. Rep., 2003.
- [16] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick, “On the spanning ratio of gabriel graphs and beta-skeletons,” pp. 412–427, 2006.
- [17] M. Chawla, N. Goel, K. Kalaichelvan, A. Nayak, and I. Stojmenovic, *Beaconless Position Based Routing with Guaranteed Delivery for Wireless Ad-Hoc and Sensor Networks*. Springer US, 2006, vol. 212, pp. 61–70.
- [18] S. Rührup, H. Kalosha, A. Nayak, and I. Stojmenovic, “Message-efficient beaconless georouting with guaranteed delivery in wireless sensor, ad hoc, and actuator networks,” *IEEE/ACM Transactions on Networking*, vol. 18, pp. 95–108, 2010.
- [19] M. Benter, F. Neumann, and H. Frey, “Reactive planar spanner construction in wireless ad hoc and sensor networks,” 2013, pp. 2193–2201.

- [20] K. Rachuri, F. A. Antony, and C. S. R. Murthy, "Coverage based expanding ring search for dense wireless sensor networks," in *15th International Conference, Bangalore, India, December 17-20, 2008. Proceedings*, 2008, pp. 245–256.
- [21] T. L. Dinh, D. T. Nguyen, and H. Thuan, "Hybrid contention-based geographic routing in wireless sensor networks," in *In Proceedings of the Second Symposium on Information and Communication Technology (SoICT)*, 2011, pp. 86–91.
- [22] D. Chen, J. Deng, and P. K. Varshney, "Selection of a forwarding area for contention-based geographic forwarding in wireless multi-hop networks," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3111–3122, 2007.
- [23] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, "Blr: beacon-less routing algorithm for mobile ad hoc networks," *Computer Communications*, vol. 27, no. 11, pp. 1076–1086, 2004.
- [24] X. Shi and K. Liu, "A contention-based beaconless geographic routing protocol for mobile ad hoc networks," in *In Proceedings of Third International Conference on Communications and Networking in China (ChinaCom)*, 2008, pp. 840–843.
- [25] M. Witt and V. Turau, "Bgr: blind geographic routing for sensor networks," *Third International Workshop on Intelligent Solutions in Embedded Systems*, pp. 51–61, 2005.
- [26] M. Jain, M. K. Mishra, and M. M. Gore, "Energy aware beaconless geographical routing in three dimensional wireless sensor networks," in *In Proceedings of First International Conference on Advanced Computing*, 2009, pp. 122–128.
- [27] T. Lee, C. Qiao, M. Demirbas, and J. Xu, "Abc: A simple geographic forwarding scheme capable of bypassing routing holes in sensor networks," *Ad Hoc Networks*, vol. 8, pp. 361–377, 2010.
- [28] M. Watanabe and H. Higaki, "No-beacon gedir: Location-based ad-hoc routing with less communication overhead," in *Fourth International Conference on Information Technology*, 2007, pp. 48–55.
- [29] R. Nawaz, S. A. Hussain, S. A. Abid, and J. Shafi, "Beaconless multihop routing protocol for wireless sensor networks," pp. 721–725, 2011.
- [30] M. K. Mishra and M. M. Gore, "An improved forwarder selection approach for energy aware geographic routing in three dimensional wireless sensor networks," pp. 166–171, 2011.

- [31] Y. Cao and S. Xie, "A position based beaconless routing algorithm for mobile ad hoc networks," in *In Proceedings of International Conference on Communications, Circuits and Systems (ICCCAS)*, vol. 1, 2005, pp. 303–307.
- [32] G. Chen, T. Sato, and K. Itoh, "Beaconless location-based routing with signal strength assisted for ad-hoc networks," in *In Proceedings of IEEE 66th Vehicular Technology Conference (VTC)*, 2007, pp. 86–90.
- [33] Y. Inoue, K. Endo, and Y. Takahashi, "A distance-aware forwarding protocol for beaconless communication in mobile ad hoc networks and its performance," 2010, pp. 1–8.
- [34] R. Jurdak, A. G. Ruzzelli, G. M. P. O'hare, and R. Higgs, "Directed broadcast with overhearing for sensor networks," *ACM Transactions on Sensor Networks*, pp. 1–35, 2009.
- [35] Y. Yim, H. Park, J. Lee, S. Oh, and S.-H. Kim, "Distributed forwarder selection for beaconless real-time routing in wireless sensor networks," pp. 1–5, 2013.
- [36] M. A. Otaibi, H. Soliman, and J. Zheng, "A multipath routeless routing protocol with an efficient location update mechanism," *Int. J. Internet Protocol Technology*, vol. 6, pp. 75–82, 2011.
- [37] P. K. Varshney, "A state-free data delivery protocol for multihop wireless sensor networks," pp. 1818–1823, 2005.
- [38] D. Chen and P. K. Varshney, "On-demand geographic forwarding for data delivery in wireless sensor networks," *Computer Communications*, vol. 30, pp. 2954–2967, 2007.
- [39] C. Huang and G. Wang, "Contention-based beaconless real-time routing protocol for wireless sensor networks," *Wireless Sensor Network*, pp. 528–537, 2010.
- [40] T. Aguilar, S.-J. Syue, V. Gauthier, H. Afifi, and C.-L. Wang, "Coopgeo: A beaconless geographic cross-layer protocol for cooperative wireless ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 2554–2565, 2011.
- [41] J. A. Sanchez, R. Marin-Perez, and P. M. Ruiz, "Boss: Beacon-less on demand strategy for geographic routing in wireless sensor networks," in *In Proceedings of 4th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2007, pp. 1–10.

- [42] M. Zorzi, S. Member, and R. R. Rao, “Geographic random forwarding (geraf) for ad hoc and sensor networks: Energy and latency performance,” vol. 2, pp. 349–365, 2003.
- [43] Y. Xu, W.-C. Lee, J. Xu, and G. Mitchell, “Energy-aware and time-critical geo-routing in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 4, pp. 315–346, 2008.
- [44] M. Zorzi, “A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks,” vol. 6, 2004, pp. 3481–3485.
- [45] H. Zhang and H. Shen, “Energy-efficient beaconless geographic routing in wireless sensor networks,” *IEEE Transactions on Parallel Distributed Systems*, pp. 881–896, 2010.
- [46] O. Jumira, R. Wolhuter, and S. Zeadally, “Energy-efficient beaconless geographic routing in energy harvested wireless sensor networks,” *Concurrency and Computation: Practice and Experience*, vol. 25, pp. 58–84, 2012.
- [47] H. Zhang and H. Shen, “Eegr: Energy-efficient geographic routing in wireless sensor networks,” in *2007 International Conference on Parallel Processing (ICPP 2007)*, 2007.
- [48] N. Mitton, E. Natalizio, and R. Wolhuter, *Beacon-Less Mobility Assisted Energy Efficient Georouting in Energy Harvesting Actuator and Sensor Networks*, 2013, pp. 281–292.
- [49] S. Das, A. Nayak, S. Rührup, and I. Stojmenovic, “Semi-beaconless power and cost efficient georouting with guaranteed delivery using variable transmission radii for wireless sensor networks,” in *In Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2007, pp. 1–6.
- [50] L. Galluccio, A. Leonardi, G. Morabito, and S. Palazzo, “A mac/routing cross-layer approach to geographic forwarding in wireless sensor networks,” pp. 872–884, 2007.
- [51] I. Amadou and F. Valois, “Pizza forwarding: A beaconless routing protocol designed for realistic radio assumptions,” in *In Proceedings of Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, 2010, pp. 495–500.
- [52] I. Amadou, G. Chelius, and F. Valois, “Energy-efficient beacon-less protocol for wsn,” in *In Proceedings of IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2011, pp. 990–994.

- [53] P. Dong, H. Qian, X. Wei, S. Lan, and C. Pu, “A beacon-less geographic multipath routing protocol for ad hoc networks,” *Mobile Networks and Applications*, vol. 18, pp. 500–512, 2012.
- [54] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, “Contention-based forwarding for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.
- [55] M. Heissenbüttel, “A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks,” in *Proceedings of ASWN '03*, 2003, pp. 197–210.
- [56] S. Rührup and I. Stojmenovic, “Optimizing communication overhead while reducing path length in beaconless georouting with guaranteed delivery for wireless sensor networks,” pp. 2440–2453, 2013.
- [57] F. Neumann and H. Frey, “Path properties and improvements of sweep circle traversals,” in *In Proceedings of 9th IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MSN), 2013*, 2013, pp. 1–9.
- [58] P. Kulakowski, E. Egea-Lopez, J. Garcia-Haro, and L. Orozco-Barbosa, “Arrow: Azimuth-range routing for large-scale wireless sensor networks,” p. 93, 2013.
- [59] P. Casari, M. Nati, C. Petrioli, and M. Zorzi, “Efficient non-planar routing around dead ends in sparse topologies using random forwarding,” pp. 3122–3129, 2007.
- [60] —, “Alba: an adaptive load-balanced algorithm for geographic forwarding in wireless sensor networks,” in *IEEE MILCOM*, Washington, DC, Oct 2006.
- [61] S. Basagni, M. Nati, C. Petrioli, and R. Petrocchia, “Rome: Routing over mobile elements in wsns,” in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, 2009, pp. 1–7.
- [62] J. A. Sanchez, R. Marin-Perez, and P. M. Ruiz, “Beacon-less geographic multicast routing in a real-world wireless sensor network testbed,” *Wireless Networks*, vol. 18, pp. 565–578, 2012.
- [63] L. Galluccio, G. Morabito, and S. Palazzo, “Geographic multicast (gem) for dense wireless networks: Protocol design and performance analysis,” *IEEE/ACM Transactions on Networking*, vol. 21, pp. 1332–1346, 2012.
- [64] H. Park, J. Lee, S. Oh, Y. Yim, and S.-H. Kim, “Distributed multicast protocol based on beaconless routing for wireless sensor networks,” in *IEEE International Conference on Consumer Electronics (ICCE)*, 2013, pp. 522–523.

- [65] Y.-B. Ko and N. Vaidya, "Geocasting in mobile ad hoc networks: location-based multicast algorithms," in *Proceedings WMCSA '99. Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 101–110.
- [66] Y. Sasaki, W.-C. Lee, T. Hara, and S. Nishio, "On alleviating beacon overhead in routing protocols for urban vanets," in *2013 IEEE 14th International Conference on Mobile Data Management*, 2013, pp. 66–76.
- [67] P. M. Ruiz, V. Cabrera, J. A. Martinez, and F. J. Ros, "Brave: Beacon-less routing algorithm for vehicular environments," in *In Proceedings of 7th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2010, pp. 709–714.
- [68] K. Z. Ghafoor, K. A. Bakar, J. Lloret, R. H. Khokhar, and K. C. Lee, "Intelligent beaconless geographical forwarding for urban vehicular environments," *Wireless Networks*, vol. 19, pp. 345–362, 2013.
- [69] H. Füßler, H. Hartenstein, M. Mauve, W. Effelsberg, and J. Widmer, "Contention-based forwarding for street scenarios," *1st International Workshop in Intelligent Transportation (WIT 2004)*, 2004.
- [70] B. Wilson and K. G. Preetha, "Enhanced beaconless routing mechanism in vanet," in *In Proceedings of International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2013, pp. 1076–1081.
- [71] M. Asgari, M. Ismail, and R. Alsaqour, "Reliable contention-based beaconless packet forwarding algorithm for vanet streets," pp. 1011–1017, 2013.
- [72] J. A. Martinez, D. Viguera, F. J. Ros, and P. M. Ruiz, "Evaluation of the use of guard nodes for securing the routing in vanets," *Journal of Communications and Networks*, vol. 15, pp. 122–131, 2013.
- [73] S. A. Chhoeun, K. S. N. Ayutaya, C. Charnsripinyo, K. Chamnongthai, and P. Kumhom, "Enhanced beaconless routing mechanism in vanet," 2009, pp. 453–457.