

# Routing und Adressierung in mobilen multi-hop Ad-hoc-Netzen

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der  
Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung  
des akademischen Grades eines Doktors der Naturwissenschaften  
genehmigte Dissertation

Vorgelegt von  
Diplom-Informatiker  
Mesut Güneş  
aus Gaziantep/Türkei

Berichter  
Universitätsprofessor Dr. Otto Spaniol  
Universitätsprofessor Dr. Petri Mähönen

Tag der mündlichen Prüfung  
19. Januar 2004

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.



---

# Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit am Lehrstuhl für Informatik IV der RWTH Aachen. Mein Dank gilt allen, die zum Entstehen dieses Werkes beigetragen haben.

Bei Herrn Prof. Otto Spaniol möchte ich mich für die Freiheit bei der Verfolgung meiner Forschungsinteressen bedanken.

Herrn Prof. Petri Mähönen gilt mein Dank für die Übernahme des Koreferats.

Bei meinem Bürokollegen Imed Bouazizi möchte ich mich für die wertvollen Gespräche bedanken. Kai Jakobs möchte ich für die Durchsicht einiger meiner Paper bedanken. Meine Diplomanden waren für einige der Implementierungsarbeiten verantwortlich, hier sei ihnen nochmals gedankt. Meinen langjährigen Freunden und Ex-WG-Mitbewohnern Aydemir Kaplangiray und Osman Öner Ünsal danke ich für die Durchsicht des gesamten Manuskripts und die wertvollen Hinweise.

Zuletzt, aber am wichtigsten, gilt mein Dank meiner Familie. Zuerst meinen Eltern, die immer zu mir gestanden haben. Meine Frau Züleyha und meine Tochter Bera Asime Zehra haben mich immer getröstet, mir Kraft und Erheiterung geschenkt und waren sehr oft mit mir geduldig, dafür danke ich ihnen vom ganzen Herzen.

Aachen, im Januar 2004

Mesut Güneş



---

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	vii
<b>Abkürzungsverzeichnis</b>	xiii
<b>1 Einführung</b>	<b>1</b>
1.1 Problemstellung und Zielsetzung der Arbeit . . . . .	2
1.2 Aufbau der Arbeit . . . . .	3
<b>2 Ad-hoc-Netze</b>	<b>5</b>
2.1 Was ist ein Ad-hoc-Netz? . . . . .	5
2.1.1 Eigenschaften von Ad-hoc-Netzen . . . . .	7
2.1.2 Die Geschichte von Ad-hoc-Netzen . . . . .	9
2.1.3 Klassifikation von Ad-hoc-Netzen . . . . .	10
2.1.4 Anwendungsgebiete von Ad-hoc-Netzen . . . . .	14
2.2 Mobilfunknetze, WLANs und MANETs . . . . .	16
2.2.1 Zellulare Mobilfunknetze . . . . .	16
2.2.2 WLAN . . . . .	17
2.3 Netzwerkarchitekturen und Ad-hoc-Netze . . . . .	18
2.3.1 Das TCP/IP-Referenzmodell . . . . .	19
2.3.2 Kommunikation in Ad-hoc-Netzen . . . . .	21
2.4 Technologien für mobile Ad-hoc-Netze . . . . .	24
2.4.1 IEEE 802.11 und Verwandte . . . . .	24
2.4.2 Bluetooth und IEEE 802.15 . . . . .	31
2.5 Fazit . . . . .	38

---

<b>3 Methodik</b>	<b>39</b>
3.1 Das Simulationstool . . . . .	39
3.1.1 Der Protokollstack . . . . .	40
3.2 Simulationsparameter und ihr Einfluss . . . . .	42
3.2.1 Das Ausbreitungsmodell . . . . .	42
3.2.2 Das Mobilitätsmodell . . . . .	43
3.3 Simulationsumgebung . . . . .	48
3.3.1 Grundeinstellungen . . . . .	48
3.3.2 Bewegungsszenarien . . . . .	50
3.3.3 Kommunikationsmuster . . . . .	51
<b>4 Adressierung in Ad-hoc-Netzen</b>	<b>55</b>
4.1 Grundlagen zur Adresskonfiguration . . . . .	56
4.2 Adressierung in lokalen Netzen . . . . .	59
4.2.1 Dynamic Host Configuration Protocol . . . . .	59
4.2.2 Zero Configuration Networking . . . . .	61
4.2.3 Mobile-IP . . . . .	61
4.3 Ansätze zur Adressierung von Ad-hoc-Netzen . . . . .	62
4.3.1 Autokonfiguration für Ad-hoc-Netze . . . . .	63
4.3.2 MANETconf . . . . .	64
4.3.3 Buddy basierte Adressierung . . . . .	65
4.3.4 Prophet Address Allocation . . . . .	66
4.3.5 Klassifikation der Verfahren . . . . .	66
4.4 Die Agentenbasierte Adressierung . . . . .	67
4.4.1 Zustandsgraph eines Knotens . . . . .	68
4.4.2 Der Adressierungsagent . . . . .	69
4.4.3 Robustheit des Verfahrens . . . . .	71
4.4.4 Generierung von Adressen . . . . .	72
4.4.5 Einfluss von Timern und Parametern . . . . .	72
4.5 Adressierungsszenarien . . . . .	74
4.5.1 Vollständige Adressierung eines Netzes . . . . .	74
4.5.2 Vereinigung von mehreren Netzen . . . . .	75

---

4.5.3	Aufteilung eines Netzes in mehrere Netze . . . . .	76
4.6	Ergebnisse . . . . .	77
4.6.1	Vollständige Adressierung eines Netzes . . . . .	77
4.6.2	Vereinigung von zwei Netzen zu einem Netz . . . . .	82
4.6.3	Aufteilung eines Netzes in mehrere Netze . . . . .	85
4.7	Fazit . . . . .	88
<b>5</b>	<b>Routing in Ad-hoc-Netzen</b>	<b>91</b>
5.1	Routingalgorithmen für Ad-hoc-Netze . . . . .	92
5.1.1	Anforderungen an Routingalgorithmen . . . . .	92
5.1.2	Klassifikation von Routingalgorithmen . . . . .	94
5.1.3	Destination-Sequenced Distance-Vector Routing . . . . .	95
5.1.4	Ad hoc On-Demand Distance Vector Routing . . . . .	96
5.1.5	Dynamic Source Routing . . . . .	99
5.1.6	Andere Routingalgorithmen für MANETs . . . . .	101
5.2	Schwarmintelligenz . . . . .	103
5.2.1	Insektenchwärme in der Natur . . . . .	104
5.2.2	Selbstorganisation . . . . .	105
5.2.3	Kommunikation in einem Schwarm . . . . .	106
5.3	Ameisenalgorithmen . . . . .	107
5.3.1	Verhalten von Ameisen bei der Futtersuche . . . . .	107
5.3.2	Ant System . . . . .	110
5.3.3	Ant Colony System . . . . .	111
5.3.4	Max-Min AS . . . . .	112
5.3.5	AS-Rank . . . . .	113
5.3.6	Adaption an andere Fragestellungen . . . . .	113
5.4	Der Ameisenroutingalgorithmus . . . . .	115
5.4.1	Adaption des Ameisenalgorithmus . . . . .	115
5.4.2	ARA im Detail . . . . .	117
5.4.3	Module von ARA . . . . .	126
5.4.4	Die Eigenschaften von ARA . . . . .	132
5.5	Ergebnisse . . . . .	133

5.5.1	Bewertungskriterien . . . . .	133
5.5.2	Vergleich verschiedener Varianten von ARA . . . . .	135
5.5.3	Übertragung von Strömen mit konstanter Datenrate .	141
5.5.4	Übertragung von Echtzeitdaten . . . . .	148
5.5.5	Übertragung von TCP-Strömen . . . . .	153
5.6	Verwandte Ansätze . . . . .	156
5.6.1	ABC Routing . . . . .	156
5.6.2	AntNET . . . . .	158
5.6.3	Verfahren von White . . . . .	159
5.6.4	GPSAL . . . . .	160
5.6.5	Mobile Agenten . . . . .	160
5.6.6	Diskussion der unterschiedlichen Ansätzen . . . . .	161
5.7	Fazit . . . . .	162
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>163</b>
<b>Literaturverzeichnis</b>		<b>167</b>
<b>Lebenslauf</b>		<b>177</b>

---

# Abbildungsverzeichnis

2.1 Ein Ad-hoc-Netz . . . . .	7
2.2 Single-hop Ad-hoc-Netz . . . . .	11
2.3 Mobiles multi-hop Ad-hoc-Netz . . . . .	12
2.4 Hierarchisches Ad-hoc-Netz . . . . .	13
2.5 Vereinfachte Systemarchitektur von zellularen Mobilfunknetzen. . . . .	16
2.6 Anbindung von mobilen Teilnehmern an ein LAN über ein WLAN. . . . .	17
2.7 Zellulare Mobilfunknetze, WLAN und Ad-hoc-Netze im Vergleich. . . . .	18
2.8 Kommunikation nach dem ISO/OSI-Referenzmodell. . . . .	19
2.9 ISO/OSI-Modell vs. TCP/IP-Modell . . . . .	20
2.10 Architektur für das Mobile Internet. . . . .	22
2.11 Horizontale vs. vertikale Kommunikation. . . . .	23
2.12 Architektur von IEEE 802.11 . . . . .	25
2.13 Protokollstack von IEEE 802.11 . . . . .	26
2.14 Hidden- und Exposed-Terminals. . . . .	29
2.15 Schematischer Ablauf des Medienzugriffs mit CS-MA/CA und der Erweiterung von RTS/CTS. . . . .	29
2.16 Übersicht der IEEE 802.11x Standards für drahtlose Netze. . . . .	30
2.17 Protokollstack von Bluetooth . . . . .	32
2.18 Protokollstack von IEEE 802.15.1 und Bluetooth . . . . .	33
2.19 Bluetooth Scatternetz . . . . .	33
2.20 Standards unter IEEE 802.15 . . . . .	37

---

2.21 Referenzmodell für Ad-hoc-Netze . . . . .	38
3.1 Protokollstapel eines Knotens im Simulationswerkzeug ns-2. . . . .	40
3.2 Signalausbreitungsmodelle . . . . .	43
3.3 Bewegung eines Knotens gemäß der eindimensionalen RWM bei der ein Punkt vom Knoten besucht wird. . . . .	46
3.4 Verteilungsfunktion von 1D-RWM . . . . .	47
3.5 Verteilungsfunktion von 2D-RWM . . . . .	47
3.6 Knotenverteilung beim Mobilitätsmodell RWM mit unterschiedlichen Pausenzeiten. . . . .	49
3.7 Verbindungsabbrüche und Pfadwechsel in Bewegungsszenarien	50
4.1 Die Konfiguration von Netzwerkknoten mit DHCP. . . . .	59
4.2 Prinzip von Mobile-IP. . . . .	62
4.3 Buddy basierte Adressierung . . . . .	65
4.4 Zustandsgraph bei der Agentenbasierten Adressierung . . . . .	69
4.5 Konfiguration mit der Agentenbasierten Adressierung. . . . .	70
4.6 Konstruktion einer IPv6-Site-Local-Adresse aus den MAC-Adressen vom Adressierungsagenten und anfragenden Knoten. . . . .	73
4.7 Direkte Adressierung . . . . .	74
4.8 Adressierung über mehrere Hops . . . . .	75
4.9 Vereinigung von Ad-hoc-Netzen. . . . .	76
4.10 Aufspaltung eines Ad-hoc-Netzes in mehrere kleine Netze. . . . .	76
4.11 Multi-hop Adressierung mit Autokonfiguration . . . . .	78
4.12 Vollständige Adressierung eines Ad-hoc-Netzes mit Autokonfiguration mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	79
4.13 Multi-hop Adressierung mit Agentenbasierter Adressierung . . . . .	80
4.14 Vollständige Adressierung eines Ad-hoc-Netzes mit der Agentenbasierten Adressierung mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	81
4.15 Vereinigung mit Autokonfiguration . . . . .	83

4.16 Vereinigung zweier Ad-hoc-Netze zu einem neuen Ad-hoc-Netz mit Autokonfiguration mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	84
4.17 Vereinigung mit Agentenbasierter Adressierung . . . . .	85
4.18 Vereinigung zweier Ad-hoc-Netze zu einem neuen Ad-hoc-Netz mit Agentenbasierter Adressierung mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	86
5.1 Klassifikation von Routingalgorithmen. . . . .	94
5.2 Pfadsuche von AODV . . . . .	98
5.3 Pfadsuche von DSR . . . . .	100
5.4 Zonenaufteilung beim Fisheye State Routing . . . . .	102
5.5 Multipoint-Relay von OLSR . . . . .	103
5.6 Futtersuchverhalten von Ameisen . . . . .	108
5.7 Futtersuchverhalten von Ameisen bei mehreren Wegen mit und ohne Futter. . . . .	109
5.8 Konvergenzeigenschaft von Ameisenalgorithmen . . . . .	114
5.9 Pfadfindung von ARA . . . . .	119
5.10 Pfadpflege von ARA . . . . .	120
5.11 Fehlerbehandlung erste Variante . . . . .	123
5.12 Fehlerbehandlung zweite Variante . . . . .	125
5.13 Verflüchtigung der Pheromonwerte. . . . .	127
5.14 Kontrollnachrichten von IEEE 802.11. . . . .	131
5.15 Leistung von ANT und ARA als Funktion der Pausenzeit bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s. . . . .	136
5.16 Leistung von ARA mit statistischer und maximaler Wegwahl als Funktion der Pausenzeit bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s. . . . .	137
5.17 Leistung von ARA mit statistischer und maximaler Wegwahl als Funktion der Pausenzeit bei 10 CBR-Duplex-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s. . . . .	138

---

5.18	Zustellrate von ARA mit Sendepuffer bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s. . . . .	139
5.19	Leistung von ANT und ARA als Funktion der Pausenzeit bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s. . . . .	140
5.20	Routingoptimalität von ANT und ARA. . . . .	141
5.21	Zustellrate von ANT und ARA mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	142
5.22	Zustellrate von AODV, DSR und ARA als Funktion der Pausenzeit bei 10 CBR-Verbindungen. . . . .	143
5.23	Zustellrate von AODV, DSR und ARA mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	144
5.24	Zustellrate von AODV, DSR und ARA als Funktion der Anzahl paralleler Verbindungen bei einer Pausenzeit von 300 Sekunden. . . . .	145
5.25	Pfadoptimalität von AODV, DSR und ARA bei 10 CBR-Verbindungen. . . . .	146
5.26	Routingaufwand von AODV, DSR und ARA als Funktion der Pausenzeit und der Anzahl paralleler Verbindungen. . . . .	147
5.27	Zustellrate von AODV, DSR und ARA als Funktion der Anzahl paralleler Audio-Duplex-Verbindungen mit 13 kBit/s in beide Richtungen. . . . .	149
5.28	Zustellrate von AODV, DSR und ARA bei Audioübertragung als Funktion der Anzahl paralleler Duplex-Verbindungen mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	150
5.29	Ende-zu-Ende Verzögerung von AODV, DSR und ARA als Funktion der Anzahl paralleler Audio-Duplex-Verbindungen mit 13 kBit/s in beide Richtungen. . . . .	151
5.30	Ende-zu-Ende Verzögerung von AODV, DSR und ARA bei Audioübertragung als Funktion der Anzahl paralleler Duplex-Verbindungen mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	152
5.31	Jitter von AODV, DSR und ARA als Funktion der Anzahl paralleler Audio-Duplex-Verbindungen mit 13 kBit/s in beide Richtungen. . . . .	154

---

5.32 Jitter von AODV, DSR und ARA bei Audioübertragung als Funktion der Anzahl paralleler Duplex-Verbindungen mit $\alpha = 0,05$ -Konfidenzintervall. . . . .	155
5.33 Durchsatz von AODV, DSR und ARA bei 5 TCP-Verbindungen. . . . .	156
5.34 Durchsatz von AODV, DSR und ARA mit $\alpha = 0,05$ -Konfidenzintervall bei 5 TCP-Verbindungen. . . . .	157



---

# Abkürzungsverzeichnis

<b>AA</b>	Adressierungsagent
<b>ABC</b>	Ant-Based Control
<b>ACK</b>	Acknowledgement
<b>ACL</b>	Asynchronous Connectionless Link
<b>ACS</b>	Ant Colony System
<b>AC</b>	Address Confirm Packet
<b>AL</b>	Address List
<b>AODV</b>	Ad hoc On-Demand Distance Vector Routing
<b>AP</b>	Access Point
<b>ARA</b>	Ameisenroutingalgorithmus, Ant Routing Algorithm
<b>ARP</b>	Address Resolution Protocol
<b>AR</b>	Address Request Packet
<b>AS</b>	Ant System
<b>BANT</b>	Backward Ant
<b>BSC</b>	Base Station Controller
<b>BSS</b>	Basic Service Set

---

<b>BS</b>	Base Station
<b>CBRP</b>	Cluster Based Routing Protocol
<b>CBR</b>	Constant Bit Rate
<b>CSGR</b>	Clusterhead Gateway Switch Routing Protocol
<b>CSMA/CA</b>	Carrier Sense Multiple Access with Collision Avoidance
<b>CSMA</b>	Carrier Sense Multiple Access
<b>CTS</b>	Clear To Send
<b>DDR</b>	Distributed Dynamic Routing Algorithm
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DNS</b>	Domain Name System
<b>DSDV</b>	Destination-Sequenced Distance-Vector Routing
<b>DSR</b>	Dynamic Source Routing
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>ESS</b>	Extended Service Set
<b>FANT</b>	Forward Ant
<b>FHSS</b>	Frequency Hopping Spread Spectrum
<b>FSR</b>	Fisheye State Routing
<b>FTP</b>	File Transfer Protocol
<b>GAP</b>	Generic Access Profile
<b>GloMo</b>	Global Mobile Information Systems
<b>GPRS</b>	General Packet Radio Service

<b>GPS</b>	GPS/Ant-Like Routing Algorithm
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile Communications
<b>HIPERLAN</b>	High Performance Radio Local Area Network
<b>HLR</b>	Home Location Register
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IETF</b>	Internet Engineering Task Force
<b>IP</b>	Internet Protocol
<b>IrDA</b>	Infrared Data Association
<b>ISM</b>	Industrial Scientifical Medical
<b>ISO</b>	International Standardization Organization
<b>L2CAP</b>	Logical Link Control Adaptation Protocol
<b>LAN</b>	Local Area Network
<b>LAR</b>	Location-Aided Routing Protocol
<b>LMP</b>	Link Management Protocol
<b>MAC</b>	Media Access Control
<b>MANET WG</b>	Mobile Ad-hoc Networking Working Group
<b>MANET</b>	Mobile Ad-hoc Network
<b>MMAS</b>	Max-Min Ant System
<b>MSC</b>	Mobile Services Switching Center

---

<b>MTU</b>	Maximum Transmission Unit
<b>NAV</b>	Network Allocation Vector
<b>NFS</b>	Network File System
<b>NIC</b>	Network Interface Card
<b>NTDR</b>	Near-term Digital Radio
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>OLSR</b>	Optimized Link State Routing Protocol
<b>OSI</b>	Open Systems Interconnection
<b>PAN</b>	Personal Area Network
<b>PCM</b>	Pulse Code Modulation
<b>PDA</b>	Personal Digital Assistant
<b>PLCP</b>	Physical Layer Convergence Protocol
<b>PMD</b>	Physical Medium Dependent
<b>PRNET</b>	Packet Radio Network
<b>QCIF</b>	Quarter Common Intermediate Format
<b>QoS</b>	Quality of Service
<b>RARP</b>	Reverse Address Resolution Protocol
<b>RREP</b>	Route-Reply-Packet
<b>RREQ</b>	Route-Request-Packet
<b>RTP</b>	Real Time Transport Protocol
<b>RTS</b>	Ready To Send

<b>RWM</b>	Random Waypoint Mobility Model
<b>SCO</b>	Synchronous Connection Oriented Link
<b>SOHO</b>	Small-Office/Home-Office
<b>TCP</b>	Transmission Control Protocol
<b>TORA</b>	Temporally-Ordered Routing Algorithm Routing Protocol
<b>TSP</b>	Traveling Salesman Problem
<b>TTL</b>	Time To Live
<b>UDP</b>	User Datagram Protocol
<b>VLR</b>	Visitor Location Register
<b>VP</b>	Verify Packet
<b>WEP</b>	Wired Equivalent Privacy
<b>WLAN</b>	Wireless Local Area Network
<b>WPAN</b>	Wireless Personal Area Network
<b>WRP</b>	Wireless Routing Protocol
<b>ZHLS</b>	Zone-Based Hierarchical Link State Routing
<b>ZRP</b>	Zone Routing Protocol



---

## KAPITEL 1

---

### Einführung

Heutzutage ist die moderne Kommunikation aus unserer Gesellschaft nicht mehr wegzudenken. Zwei technologische Entwicklungen haben hierzu besonders beigetragen. Entwicklung 1: Das World-Wide-Web hat das Internet in den Mittelpunkt der Kommunikation gerückt. Sowohl privat als auch geschäftlich findet die Kommunikation hauptsächlich über das Internet statt. Das Internet erlaubt jedoch nicht nur das Versenden von elektronischen Briefen, sondern es ist auch die Plattform zum Auffinden von Information, Unterhaltungsmedium und teilweise auch die Basis für Geschäftsbeziehungen geworden. Entwicklung 2: Der Erfolg der GSM-Mobilfunktechnik erlaubt den Zugriff auf Informationen fast von überall und jederzeit.

Eine dritte Entwicklung, die diese beiden ergänzt, ist die Verfügbarkeit von leistungsfähigen mobilen Computern. Sei es ein mobiles Telefon, ein PDA, ein Smartphone, das beides integriert, oder ein TabletPC, der den Komfort eines PDAs mit der Leistung eines Arbeitsplatzrechners vereinigt – sie sind ständige Begleiter des modernen Benutzers und verfügen über eine oder mehrere drahtlose Kommunikationstechniken.

Die Kombination dieser drei Entwicklungen wird unter dem Begriff *pervasive* oder *ubiquitous Computing* verstanden, d.h. der Zugriff auf Informationen von jedem Ort und zu jeder Zeit.

In Unternehmen werden Arbeitsplatzrechner benutzt, die über ein lokales Netzwerk verbunden sind. Entsprechend sind die Büros ausgelegt. Ist es jedoch erforderlich ein Netzwerk schnell an einem Ort zu installieren, an dem das Verlegen von Leitungen nicht möglich, erlaubt oder erwünscht ist, sind drahtlose Netze die einzige Lösung.

Sowohl für den Datenaustausch zwischen unterschiedlichen Geräten für das pervasive Computing, als auch für die drahtlose Kommunikation in lokalen Netzen, die ein größeres Areal abdecken und viele Benutzer bedienen sollen,

sind besondere Kommunikationstechniken erforderlich. Ad-hoc-Netze, die im Rahmen dieser Arbeit behandelt werden, bieten sich für diese und andere Szenarien als ein Mittel der Kommunikation an.

Dieses Kapitel ist im Weiteren wie folgt aufgebaut. In Abschnitt 1.1 wird die Problemstellung, die dieser Arbeit zu Grunde liegt, und die Zielrichtung beschrieben. Abschnitt 1.2 gibt einen Überblick über die Arbeit.

## 1.1 Problemstellung und Zielsetzung der Arbeit

Ein Ad-hoc-Netzwerk besteht aus einer Menge von Knoten, die über Funk miteinander kommunizieren und dafür keinerlei Infrastruktur benötigen. Zwei Knoten, die sich in ihrer gegenseitigen Reichweite befinden, können direkt miteinander kommunizieren. Knoten, die voneinander entfernt sind, benötigen die Hilfe von weiteren Knoten, die sich zwischen ihnen befinden. Ein Ad-hoc-Netzwerk ist flexibel, sowohl hinsichtlich der Selbstkonfiguration als auch bezüglich der Anpassung an die Netzwerkstruktur, d.h. an die vorhandenen Netzwerkteilnehmer und die Netzwerktopologie.

In dieser Arbeit werden grundlegende Verfahren betrachtet, die für die Entwicklung und Realisierung von zukünftigen Ad-hoc-Netzen erforderlich sind. Obwohl sich Ad-hoc-Netze von klassischen leitungsgebundenen Netzen in nur wenigen Punkten unterscheiden, besitzen sie von Haus aus bestimmte Eigenschaften, wodurch sie sehr schwer handhabbar sind. Vor allem zwei Aspekte von Ad-hoc-Netzen erschweren ihre effiziente Realisierung. Der erste Aspekt ist das verwendete Übertragungsmedium, nämlich die Luftschnittstelle, die, verglichen mit anderen Medien, schlechte Eigenschaften für die Kommunikation besitzt. Der zweite Aspekt ist die sich ändernde Netzwerktopologie, welche durch die Knotenmobilität verursacht wird. Diese beiden Problemursachen haben Auswirkungen auf alle Schichten des Kommunikationsprotokolls.

Im Rahmen dieser Arbeit werden zwei aktuelle Fragestellungen von Ad-hoc-Netzen behandelt. Die erste Fragestellung betrifft das Routing und die zweite Fragestellung die Autokonfiguration in Ad-hoc-Netzen.

### Routing

Das Routing wird in Ad-hoc-Netzen durch die Knotenmobilität erschwert, da im ungünstigsten Fall die Netzwerktopologie sich ständig ändert. Um die Kommunikation effizient gestalten zu können, müssen die Datenpakete auf einem günstigsten (bzw. kürzesten) Pfad zwischen dem Quell- und Zielknoten übertragen werden. Dies erfordert von den eingesetzten Routingalgorithmen

eine hohe Adoptionsfähigkeit an die Netzwerktopologie. Dabei ist darauf zu achten, dass in einem Ad-hoc-Netz, wegen des Fehlens einer Infrastruktur, die Funktionalität von den teilnehmenden Knoten erbracht werden muss. In dieser Arbeit wird ein neuartiger Routingalgorithmus auf der Basis von Ameisenalgorithmen, die ein Teilgebiet der Schwarmintelligenz sind, vorgestellt und mit bekannten Routingalgorithmen für Ad-hoc-Netze verglichen.

## Adresskonfiguration

Die zweite Frage, mit der sich diese Arbeit beschäftigt, ist die automatische Konfiguration von Ad-hoc-Netzen. Da in Zusammenhang mit Ad-hoc-Netzen implizit auch über Zero-Konfiguration-Netzwerke gesprochen wird, spielt diese Fragestellung eine wichtige Rolle. Es wird also angenommen, dass ein Ad-hoc-Netz sich automatisch selbstkonfiguriert, ohne dass Benutzereingriffe erforderlich sind. Viele Untersuchungen im Bereich der Ad-hoc-Netze gehen davon aus, dass alle Knoten in einem Ad-hoc-Netz eine eindeutige Adresse haben. Bevor jedoch z.B. ein Routingalgorithmus einen Pfad zwischen Quell- und Zielknoten finden kann, müssen die Knoten auf der Netzwerkschicht identifiziert werden. Diese Arbeit stellt einen verteilten Algorithmus für die automatische Konfiguration von Ad-hoc-Netzen und das zugehörige Protokoll vor und bewertet seine Leistung.

## 1.2 Aufbau der Arbeit

Die Arbeit ist wie folgt aufgebaut.

- Kapitel 2 führt Ad-hoc-Netze im Allgemeinen ein, arbeitet ihre Eigenschaften aus und versucht, die unterschiedlichen Arten von Ad-hoc-Netzen, die in der Literatur behandelt werden, zu klassifizieren. Zur Verdeutlichung der Flexibilität von Ad-hoc-Netzen und der dadurch bedingten Schwierigkeiten wird ein Vergleich zu anderen mobilen Netzwerken gezogen und die Problematik der Netzwerkarchitektur von Ad-hoc-Netzen diskutiert. In diesem Teil der Arbeit werden auch zwei wichtige Technologien, die für die Realisierung von Ad-hoc-Netzen in Frage kommen, vorgestellt. Am Ende des Kapitels wird die Art von Ad-hoc-Netzen beschrieben, die im Rahmen dieser Arbeit betrachtet wird.
- Kapitel 3 stellt die für die Leistungsbewertung benutzte Methodik vor. Insbesondere werden das verwendete Simulationswerkzeug und die benutzten Simulationsparameter diskutiert.

- In Kapitel 4 wird die Konfiguration von Netzen allgemein und die automatische Adresskonfiguration von mobilen multi-hop Ad-hoc-Netzen im Besonderen betrachtet. In diesem Teil der Arbeit wird versucht, einen Überblick über mögliche Konfigurationsarten von Knoten in einem Netz zu geben. Weiterhin werden Argumente geliefert, warum existierende Konfigurationsverfahren sich nicht für mobile multi-hop Ad-hoc-Netze eignen. Schließlich wird die Agentenbasierte Adressierung von mobilen multi-hop Ad-hoc-Netzen vorgestellt und die Leistung bewertet.
- Kapitel 5 behandelt das Routing in mobilen multi-hop Ad-hoc-Netzen. Nach einer Klassifizierung von existierenden Routingalgorithmen und der Besprechung einiger besonders interessanter Ansätze von Routingalgorithmen für Ad-hoc-Netze, erfolgt eine Einführung in das Gebiet der Schwarmintelligenz und der Ameisenalgorithmen. Anschließend wird der Ameisenroutingalgorithmus für mobile multi-hop Ad-hoc-Netze vorgestellt und mit zwei bekannten Routingverfahren verglichen.
- Kapitel 6 schließt mit einer Zusammenfassung der vorgestellten Erkenntnisse und einem Ausblick auf offene Themen die Arbeit ab.

---

## KAPITEL 2

---

# Ad-hoc-Netze

Durch Ad-hoc-Netze werden existierende Anwendungen, die über leitungsgebundene Netze realisiert sind, auch in Umgebungen möglich, in denen keine Infrastruktur vorhanden oder der Aufbau einer Netzwerkinfrastruktur nicht möglich ist. Zu diesen Anwendungen gehören der Aufbau von Personal Area Networks (PAN) genauso wie der Aufbau von Heimnetzen und Netzwerken für Freizeitanwendungen wie Spiele im Gelände. Es wird gewünscht, dass auf diesen Netzen vorhandene Protokolle und Anwendungen in einer gewohnten Weise und Performance laufen. Durch ihre Flexibilität und der erwarteten einfachen Handhabung werden Ad-hoc-Netze sowohl die Kommunikation zwischen Mensch–Maschine als auch Maschine–Maschine erheblich verbessern.

In diesem Kapitel werden die Grundlagen für die später behandelten Themen erarbeitet. Das Kapitel ist im Weiteren wie folgt aufgebaut. In Abschnitt 2.1 werden Ad-hoc-Netze im Allgemeinen charakterisiert, die Entwicklungs geschichte kurz angerissen und drei unterschiedliche Klassifizierungen vorgestellt. Abschnitt 2.2 stellt zwei weitere prominente Netztypen für die mobile Kommunikation vor und stellt mobile Ad-hoc-Netze diesen gegenüber. An schließend wird in Abschnitt 2.3 auf die Architektur von Netzen eingegangen und grundsätzliche Unterschiede zwischen mobilen Ad-hoc-Netzen und typischen Festnetzen diskutiert. Abschnitt 2.4 stellt zwei wichtige Technologien für Ad-hoc-Netze im Detail vor. Das Kapitel endet in Abschnitt 2.5 mit einer Zusammenfassung und einer Beschreibung der im Rahmen dieser Arbeit betrachteten Ad-hoc-Netze.

### 2.1 Was ist ein Ad-hoc-Netz?

Der Begriff *ad-hoc* kommt aus dem lateinischen und bezeichnet einen plötzlichen Zustandswechsel für einen bestimmten Zweck [Dud00]. Der Begriff

*Ad-hoc-Netz* bezeichnet ein drahtloses Netzwerk, das ohne das Vorhandensein von fester Infrastruktur und ohne großen Konfigurationsaufwand aufgebaut werden kann.

Im Allgemeinen besteht ein Ad-hoc-Netz aus einer Menge von Knoten, die ohne vorinstallierte Infrastruktur miteinander über Funk kommunizieren. Da per definitionem keine spezielle Infrastruktur existiert, muss der Kommunikationsverkehr zwischen Quell- und Zielknoten, die sich nicht in ihrer gegenseitigen Reichweite befinden, über mehrere Zwischenknoten weitergeleitet werden. Diese Art von Ad-hoc-Netzen werden auch multi-hop Ad-hoc-Netze genannt. Die Knoten in einem multi-hop Ad-hoc-Netz müssen deshalb, neben ihrer eigentlichen Funktion, auch Dienste der nicht vorhandenen Infrastruktur erbringen.

In Abbildung 2.1 ist ein multi-hop Ad-hoc-Netz dargestellt. Anhand dieser Abbildung sollen die grundlegenden Eigenschaften von Ad-hoc-Netzen diskutiert und Probleme aufgezeigt werden, mit denen ein Ad-hoc-Netz umgehen muss. Das Ad-hoc-Netz besteht aus drei Laptops, einem Notebook, einem Palmtop, einem Smartphone und einer Kamera. Alle Knoten sind mit entsprechenden Funktransceivern ausgerüstet und es existiert keine Infrastruktur. Laptop-1 und Laptop-2 können direkt miteinander kommunizieren, Laptop-3 kann direkt mit dem Smartphone kommunizieren, der wiederum kann Laptop-1 und den Palmtop erreichen, und die Kamera kann nur über das Notebook erreicht werden. Die einzelnen Teilnehmer im Netz sind technisch unterschiedlich ausgelegt. Es ist zu erwarten, dass die Laptops und das Notebook leistungsstärkere Prozessoren und einen großen Bildschirm im Vergleich zum Palmtop und dem Smartphone haben werden.

Wenn nun der Palmtop den Videostrom von der Kamera empfangen möchte, muss er vorher herausfinden, dass es so eine Kamera existiert und diesen Dienst anbietet. Danach muss der Palmtop die Kamera ansprechen. Hier stellt sich die Frage, ob die Kamera jedem Anfragenden den Dienst erbringt oder vorher eine Authentifizierung voraussetzt. Wenn eine Authentifizierung erforderlich ist, stellt sich die Frage, wie sich die Geräte gegenseitig authentifizieren können. Wenn diese Frage nicht besteht oder gelöst ist, muss der Videostrom von der Kamera an den Palmtop übertragen werden. Hierzu muss ein Pfad zwischen den beiden aufgebaut und während der Kommunikation aufrechterhalten werden. Die Kamera wird den Videodatenstrom aus technischen Gründen nur in einem bestimmten Format liefern, z.B. QCIF und in Farbe. Dies kann zu folgenden Schwierigkeiten führen: Der Palmtop hat keinen Farbbildschirm und benötigt daher die Farbinformationen gar nicht, wodurch eine unnötige Belastung des Netzes entsteht. Um die Netzwerkbelastung zu reduzieren könnte der Palmtop versuchen, einen Teilnehmer im Netz zu finden, der auf dem Pfad liegt und die Daten vor dem Weiterleiten konvertiert. Zu all diesen Fragen kommt die Schwierigkeit der Mobilität der

einzelnen Knoten im Netzwerk hinzugefügt. Während der Übertragung könnte sich der Palmtop nach rechts oben bewegen (Abbildung 2.1(b)). Somit müsste das Netzwerk auf diese Topologieänderung reagieren, um den Videostrom weiterhin an den korrekten Empfänger zu liefern.

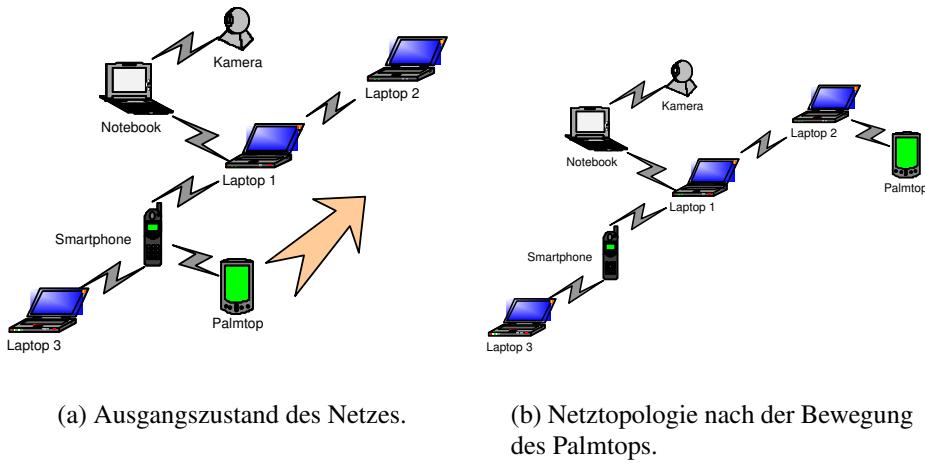


Abbildung 2.1: Ein Ad-hoc-Netz.

### 2.1.1 Eigenschaften von Ad-hoc-Netzen

Wie aus dem Beispiel deutlich wurde, besitzen Ad-hoc-Netze eine Vielzahl von Eigenschaften, die zu unterschiedlichen Schwierigkeiten führen [CMC99, FL01], die berücksichtigt werden müssen.

- **Keine Infrastruktur**

Da Ad-hoc-Netze ohne Infrastruktur arbeiten, müssen die Knoten neben ihrer eigentlichen Funktion auch die Funktionen der fehlenden Infrastruktur übernehmen.

Eine Konsequenz dieser Eigenschaft kann es sein, dass durch die zusätzlichen Dienste die Knoten erbringen müssen, ihre beschränkten Ressourcen möglicherweise nicht mehr ausreichen, um ihre eigentliche Funktionalität aufrecht zu halten.

- **Knotenmobilität**

Durch die Knotenmobilität ist die Netzwerktopologie ständigen Änderungen unterworfen. Dies erfordert, dass die eingesetzten Verfahren sich an die verändernde Topologie anpassen. Zusätzlich kommt die Schwierigkeit hinzu, dass das Verhalten der Knoten nicht vorhersagbar

ist. Zu beliebigen Zeiten können neue Knoten zum Netz hinzukommen und existierende Knoten temporär oder für immer aus dem Netz ausscheiden.

- **Keine ständig erreichbaren Knoten**

Eine direkte Konsequenz der vorherigen Eigenschaften ist, dass in einem Ad-hoc-Netz keine Knoten existieren können, die immer erreichbar sind. Deshalb können Lösungsansätze, die in zellularen Mobilfunknetzen und drahtlosen LANs funktionieren, auf mobile Ad-hoc-Netze nicht direkt übertragen werden. Beispielsweise gestaltet sich das Auffinden von Diensten in einem Ad-hoc-Netz viel schwieriger als bei zellulären Mobilfunknetzen, da kein zentrales Dienstverzeichnis existiert, das immer erreichbar ist.

- **Kurzlebigkeit**

Im Vergleich zu Festnetzverbindungen werden mobile Ad-hoc-Netze nur für kurze Zeit aufgebaut. Gleichzeitig ist zu erwarten, dass die in einem Ad-hoc-Netz benutzten Geräte sehr schnell betriebsbereit sein werden, z.B. ist ein Palmtop oder ein Smartphone auf Knopfdruck betriebsbereit. Deshalb werden die Benutzer längere Aufbau- und Konfigurationszeiten des Netzes nicht akzeptieren. Aus diesem Grund ist es erforderlich, dass der Aufbau und die Konfiguration des Netzes schnell und möglichst ohne Eingriff vom Benutzer erfolgt.

- **Heterogene Geräte**

Die Geräte in einem Ad-hoc-Netz werden hinsichtlich ihrer Komponenten wie Prozessor, Speicher, Bildschirm und Energieversorgung sehr unterschiedlich sein. Die eingesetzten Verfahren sind daher gefordert diese Einschränkungen zu berücksichtigen.

- **Funkkommunikation**

Die Funkkommunikation unterliegt im Vergleich zur leitungsgebundenen Kommunikation höheren Bitfehlerraten und niedrigeren Übertragungsraten. Die Verbindungsqualität zwischen zwei Kommunikationsendpunkten wird von Verbindungsabbrüchen, die durch Knotenmobilität bedingt sind, weiter verschlechtert. Aus diesen Gründen sollte die zur Verfügung stehende Bandbreite möglichst optimal ausgenutzt werden.

- **Multi-hop-Kommunikation**

Das Vorhandensein von mehreren Funkverbindungen auf dem Pfad zwischen Quell- und Zielknoten vermindert die Gesamtverbindungsqualität. Die eingesetzten Routingalgorithmen sollten daher die kürzesten

bzw. günstigsten Pfade für die Übertragung der Daten finden, was durch die hohe Dynamik erschwert wird.

### 2.1.2 Die Geschichte von Ad-hoc-Netzen

Mobile Ad-hoc-Netze sind genauso alt wie das Internet [FL01, CMC99]. Parallel zur Entwicklung des Internets startete das US-Militär Forschungsarbeiten für ein paketvermitteltes Funknetzwerk. 1972 wurde das *Packet Radio Network (PRNET)* vom US-Militär ins Leben gerufen. In den 80er Jahren wurden im Projekt *Survivable Adaptive Radio Networks (SURAN)* die Arbeiten weitergeführt [RR02]. Das Ziel war, ein infrastruktur-loses paketvermitteltes Funknetz für mobile Endgeräte in feindlichen Umgebungen zu entwickeln. Als mobile Endgeräte wurden Panzer, Soldaten und Flugzeuge in Betracht gezogen [FL01]. Als Mediumzugriffsverfahren wurde im PRNET eine Kombination aus ALOHA und CSMA eingesetzt. Für das Routing wurde eine spezielle Variante des Distance-Vector-Routings eingesetzt. Im Nachfolgeprojekt SURAN wurden die Verfahren mit den gewonnenen Erfahrungen weiter verbessert. Da sich das Routing als großes Problem herausstellte, wurde ein neuer Routingalgorithmus auf der Basis des Link-State Routingalgorithmus eingesetzt.

Die Entwicklung von mobilen Ad-hoc-Netzen setzte sich in den 90er Jahren fort, jedoch kam der Antrieb jetzt aus dem zivilen Bereich. Leistungsfähige Laptops, die ohne ständige Anbindung an das Stromnetz eine gewisse Mobilität erlaubten, wurden populär und stärkten den Trend der drahtlosen Kommunikation. Zuerst wurde die drahtlose Kommunikation mit der Infrarottechnologie IrDA und anschließend über die Wireless-LAN Technologie populär. IrDA hatte den großen Nachteil, sehr anfällig gegen kleinste Störungen zu sein und erforderte auch einen direkten Sichtkontakt der kommunizierenden Geräte.

Parallel zu dieser Entwicklung wurde die Popularität von Kleinst-Computern, den so genannten Palmtops oder Palm-Sized-Computer, immer größer. Um die Kommunikation, z.B. den Datenabgleich zwischen Palmtop und Desktop, zu erleichtern, wurde, die für die Kurzstreckenkommunikation ausgelegte Technologie Bluetooth entwickelt, wodurch herkömmliche Kabel durch eine robuste Funktechnologie ersetzt werden sollte.

Das US-Militär re-initiierte zu dieser Zeit seine Aktivitäten im Bereich der Ad-hoc-Netze. Projekte wie das *Global Mobile Information Systems (GloMo)* und das *Near-term Digital Radio (NTDR)* wurden ins Leben gerufen [Sas99]. Das Ziel der Militärs war diesmal die Büro- und Multimediakommunikation in einer zu Ethernet ähnlichen Qualität für militärische Zwecke zu erreichen.

Die Bestrebungen im Bereich der mobilen Ad-hoc-Netze regte auch das In-

teresse der Internet-Community und der hierzu gehörigen *Internet Engineering Task Force (IETF)*, welches für die technologische Seite des Internets verantwortlich ist, an. Mitte der 90er Jahre wurde die Arbeitsgruppe *Mobile Ad-hoc Networking Working Group (MANET WG)* initiiert, die sich mit der Standardisierung von Protokollen für mobile Ad-hoc-Netze beschäftigt [Iet]. Die Arbeitsgruppe beschäftigt sich hauptsächlich mit der Entwicklung von optimierten Routingverfahren für Ad-hoc-Netze. Das Anliegen der MANET WG ist die Realisierung von mobilen Ad-hoc-Netzen auf der Basis von offenen Internet-Protokollen wie TCP und IP. Vor allem aber die Anbindung dieser Netze an das Internet regt das Interesse der Internet-Community an. Eine zweite Arbeitsgruppe innerhalb der IETF, die sich hauptsächlich mit der Skalierbarkeit von Ad-hoc-Netzen beschäftigt, wurde 2003 gegründet [Iet03].

### 2.1.3 Klassifikation von Ad-hoc-Netzen

Es gibt keine von der Literatur vorgegebene und allgemein anerkannte Klassifikation von Ad-hoc-Netzen. Jedoch lässt sich eine Klassifizierung auf der Grundlage der in der Literatur behandelten Netztypen durchführen. Im Folgenden werden Ad-hoc-Netze nach drei unterschiedlichen Aspekten klassifiziert.

#### Klassifikation nach der Kommunikation

Die Klassifikation nach der Kommunikationsart basiert auf der Unterscheidung, wie zwei Knoten in einem Ad-hoc-Netz miteinander kommunizieren.

- **Single-hop Ad-hoc-Netze**

Bei dieser einfachsten Art von Ad-hoc-Netzen befinden sich alle Knoten in ihrer gegenseitigen Reichweite, d.h. die einzelnen Knoten können direkt miteinander kommunizieren, ohne dass der Kommunikationsverkehr über andere Knoten weitergeleitet werden muss (siehe Abbildung 2.2). Diese Art von Netzen könnte man auch als Plug-and-Play-Netze bezeichnen, da es hier hauptsächlich um den einfachen und schnellen Aufbau von temporären Verbindungen geht.

Bei dieser Klasse spielt die Mobilität keine Rolle. Die einzelnen Knoten müssen nicht statisch sein, sie müssen jedoch innerhalb der Reichweite aller Knoten bleiben, d.h. das gesamte Netz könnte sich als Gruppe bewegen, was an den Kommunikationsbeziehungen nichts ändern würde.

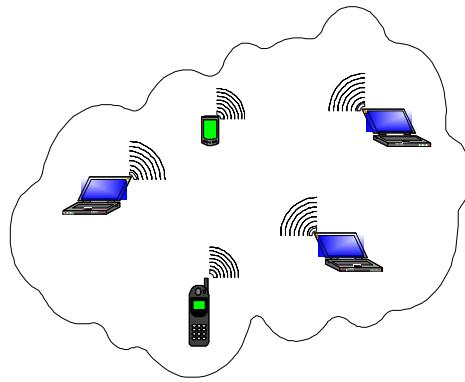


Abbildung 2.2: Ein single-hop Ad-hoc-Netz, in dem alle Knoten sich gegenseitig direkt erreichen können und daher keine Weiterleitung des Kommunikationsverkehrs über Zwischenknoten nötig ist.

- **Mobile multi-hop Ad-hoc-Netze**

Diese Klasse ist die in der Literatur am meisten untersuchte Art von Ad-hoc-Netzen. Sie unterscheidet sich von der ersten Klasse darin, dass nun der Kommunikationsverkehr zwischen einem Quell- und Zielknoten über Zwischenknoten weitergeleitet werden muss. Bei dieser Klasse wird auch angenommen, dass die Knoten mobil sind.

Die grundsätzliche Schwierigkeit der Netze dieser Klasse ist die Knotenmobilität, wodurch die Netztopologie ständigen Veränderungen unterworfen ist. Die Abbildung 2.3 zeigt ein mobiles multi-hop Ad-hoc-Netz. Die Hauptprobleme in Netzen dieser Klasse sind die Folgenden:

- Das eingesetzte Routingverfahren muss adaptiv sein und sich an die schnelle Topologieänderung anpassen. Vor allem alte Informationen in Routingtabellen können hier zu einer schlechten Leistung bis hin zum Abbruch der Kommunikation führen.
- Es ist eine offene Frage, wie die Knoten eines solchen Netzes automatisch konfiguriert werden können. Insbesondere die Tatsache, dass es keine festen Knoten gibt, die wichtige Dienste übernehmen könnten, erfordert die automatische Konfiguration verteilte, adaptive Verfahren.

- **Mobile multi-hop multimedia Ad-hoc-Netze (3M-Netze)**

Diese Klasse ist eine Erweiterung der zweiten Klasse mit dem Unterschied, dass hier der Kommunikationsverkehr hauptsächlich aus Echtzeitdaten wie Audio und Video besteht. Die Netze dieser Klasse wer-

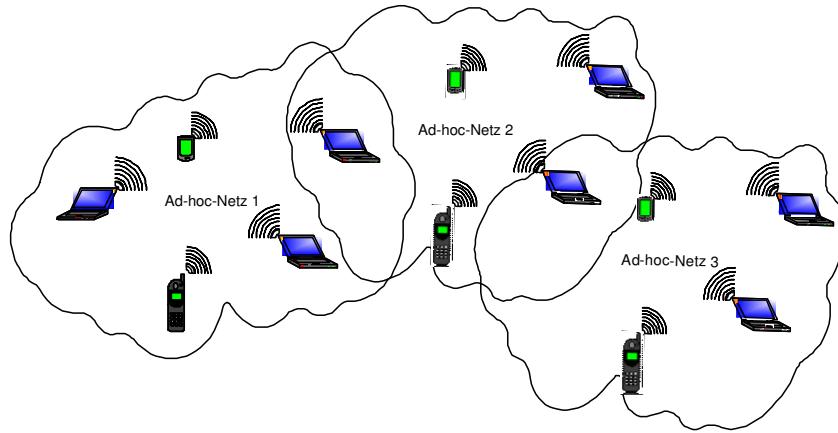


Abbildung 2.3: Ein mobiles multi-hop Ad-hoc-Netz, bei dem sich nicht alle Knoten direkt erreichen können. Die Kommunikation zweier entfernter Knoten wird durch die Weiterleitung über Zwischenknoten gewährleistet.

den auch als 3M-Netze [BG02, BG03] bezeichnet. Bei dieser Klasse spielen Qualitätsanforderungen an das Ad-hoc-Netz die Hauptrolle, da Echtzeitdaten andere Anforderungen an das Kommunikationsnetz haben.

### Klassifikation nach der Netztopologie

Eine andere Klassifikation von Ad-hoc-Netzen lässt sich auf der Grundlage der Netztopologie durchführen. Die einzelnen Knoten in einem Ad-hoc-Netz werden in unterschiedliche Typen mit speziellen Aufgaben aufgeteilt. Die in der Literatur behandelten Ad-hoc-Netze lassen sich in diesem Fall zwei Klassen zuordnen.

- **Flache Ad-hoc-Netze**

Es findet keine Unterscheidung zwischen den einzelnen Knoten statt, alle Knoten sind gleichwertig und können alle Aufgaben im Ad-hoc-Netz übernehmen.

- **Hierarchische Ad-hoc-Netze**

In diesem Fall besteht ein Ad-hoc-Netz aus mehreren Clustern, die ein Mininetzwerk darstellen und miteinander verknüpft sind (siehe Abbildung 2.4). Die Knoten in hierarchischen Ad-hoc-Netzen lassen sich in zwei Arten unterscheiden:

- **Master-Knoten:** verwalten die Cluster und sind für das Weiterleiten der Daten an andere Cluster verantwortlich.
- **Normale-Knoten:** Kommunizieren innerhalb des Clusters direkt miteinander und mit Knoten in anderen Clustern mit Hilfe des Master-Knotens. Normale-Knoten werden auch als Slave-Knoten bezeichnet.

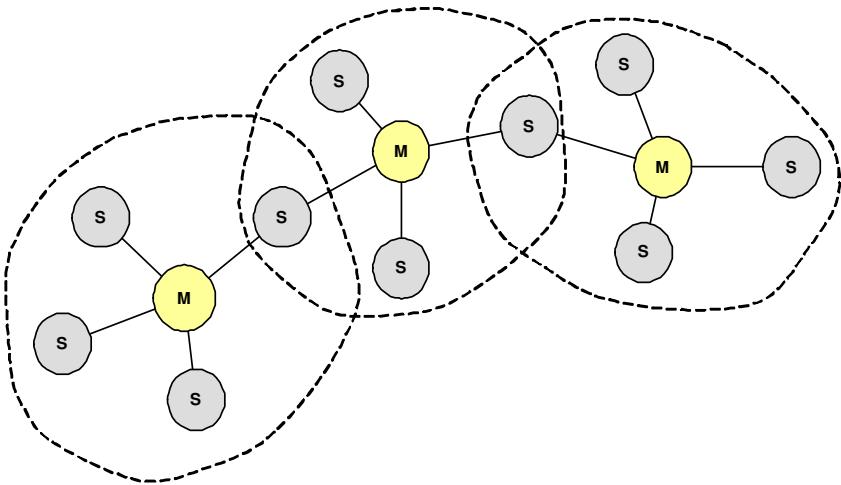


Abbildung 2.4: Hierarchisches Ad-hoc-Netzwerk bestehend aus Master-Knoten (M) und Normalen-Knoten (S).

Man nimmt an, dass der Großteil der Kommunikation innerhalb der Cluster selbst und nur ein Bruchteil zwischen unterschiedlichen Clustern stattfindet. Bei der Kommunikation innerhalb eines Clusters ist keine Weiterleitung des Kommunikationsverkehrs notwendig. Für die Vermittlung einer Verbindung zwischen Knoten in unterschiedlichen Clustern ist der Master-Knoten verantwortlich.

### Klassifikation nach Knotenausstattung

Eine weitere Klassifikation von Ad-hoc-Netzen lässt sich auf der Grundlage der Hardwareausstattung der Knoten durchführen [Toh02]. Die Ausstattung der Knoten in einem Ad-hoc-Netz ist wichtig und kann sehr stark von der eigentlichen Anwendung abhängen.

- **Homogene Ad-hoc-Netze**

In homogenen Ad-hoc-Netzen besitzen alle Knoten die gleichen Eigenchaften hinsichtlich der Hardwareausstattung wie Prozessor, Speicher,

Bildschirm und Peripheriegeräte. Bekanntester Vertreter von homogenen Ad-hoc-Netzen sind drahtlose Sensornetze. In homogenen Ad-hoc-Netzen können Anwendungen von bestimmten Voraussetzungen ausgehen, beispielsweise wird die Lokalisierung von Knoten durch das Vorhandensein von GPS-Komponenten in jedem Knoten beträchtlich erleichtert.

- **Heterogene Ad-hoc-Netze**

In heterogenen Ad-hoc-Netzen unterscheiden sich die Knoten hinsichtlich der Hardwareausstattung. In Ad-hoc-Netzen dieser Klasse können nicht alle Knoten die gleichen Dienste erbringen, deshalb spielt das Anbieten und Auffinden von Diensten hier eine besonders wichtige Rolle.

#### 2.1.4 Anwendungsbereiche von Ad-hoc-Netzen

Im Allgemeinen sind Ad-hoc-Netze für alle Situationen in denen ein temporärer Kommunikationswunsch existiert bzw. der Aufbau von Netzinfrastruktur nicht möglich oder gewünscht ist geeignet. Bekannte Anwendungsbereiche für Ad-hoc-Netze sind:

- Heimnetze
- Gruppenbesprechungen
- Hallenszenario
- Fahrzeugkommunikation
- Aufbau von PANs (Personal Area Network) für die Kommunikation von mehreren tragbaren Geräten
- Katastrophensituationen
- Kriegssituationen

In den folgenden Unterabschnitten werden einige Szenarien diskutiert, die mehrere Anwendungen vereinen. Der Einsatz von Ad-hoc-Netzen für militärische Zwecke wird hier nicht betrachtet.

#### Katastrophenszenario

Ein in der Literatur sehr beliebtes Szenario ist der Einsatz von Ad-hoc-Netzen in Katastrophensituationen. Bei einer großen Naturkatastrophe, wie einem

Erdbeben, kann es passieren, dass die vorhandene Kommunikationsinfrastruktur nicht mehr funktioniert. Das kann daran liegen, dass die Vermittlungsstellen (bei zellularen Mobilfunknetzen die Basisstationen) nicht mehr funktionieren, oder das Kommunikationsmedium (Kabel, Glasfaser) teilweise oder vollständig beschädigt ist.

Die Rettungseinheiten werden mit entsprechenden Kommunikationsgeräten ausgerüstet, die ihnen die Kommunikation untereinander und mit der Einsatzzentrale ermöglichen.

## Hallenszenario

Menschen befinden sich oft in großen Gebäuden, die vereinfacht als Hallen angesehen werden können. Hierzu gehören Einkaufszentren, Messehallen, Konferenzräume, Schulen und Museen. Unterschiedliche Veranstaltungen erfordern, dass in diesen Räumen oft umgebaut werden muss.

In allen Anwendungsfällen dieses Szenarios bewegen sich die Benutzer jedoch nach unterschiedlichen Mustern. In einem Einkaufszentrum bewegen sich einzelne Personen mit unterschiedlicher Geschwindigkeit von einem Geschäft zum anderen. In einem Museum verweilt eine Gruppe von Besuchern eine bestimmte Zeit vor einem Exponat bevor sie zum nächsten Exponat geht. Die Teilnehmer einer Konferenz oder die Schüler in einer Schule verhalten sich eher bimodal. Es gibt kurze mobile Zeiten und lange statische Zeiten. Zwischen diesen Phasen ändert sich u.U. der Aufenthaltsort oder auch die Rolle der Person. Beispielsweise wird ein Konferenzteilnehmer längere Zeit eine Zuhörerrolle einnehmen. Seine Rolle ändert sich wenn seine Vortragsreihe kommt. Die Anforderungen des Benutzers an das Kommunikationsnetz werden in den beiden Rollen unterschiedlich sein.

## Small-Office/Home-Office

Das Szenario *Small-Office/Home-Office* (SOHO) beschreibt einen Haushalt oder ein kleines Büro mit mehreren Geräten, die vernetzt werden können. In dieser Umgebung ist der Aufwand einer verkabelten Vernetzung und insbesondere der Konfigurationsaufwand oft nicht akzeptabel. Technisch versieretes Personal oder eigenes Wissen ist oft nicht vorhanden. Der Einsatz eines Ad-hoc-Netzes, das sich selbst konfiguriert, ist hier wünschenswert.

## 2.2 Mobilfunknetze, WLANs und MANETs

In diesem Abschnitt werden zellulare Mobilfunknetze, drahtlose lokale Netze und mobile Ad-hoc-Netze miteinander verglichen. Dabei werden die ersten beiden Netzarten nur schematisch beschrieben.

### 2.2.1 Zellulare Mobilfunknetze

In Abbildung 2.5 wird der Aufbau und die Systemarchitektur von zellulären Mobilfunknetzen, angelehnt an GSM, vereinfacht dargestellt. Eine detaillierte Erläuterung der Systemarchitektur würde den Rahmen dieser Arbeit sprengen. Hier soll sie nur zur Verdeutlichung der benötigten Infrastruktur, die für die Realisierung der Mobilkommunikation erforderlich ist, dienen.

Zellulare Mobilfunknetze unterteilen die geographische Fläche in Zellen auf, die hauptsächlich als Sechsecke modelliert werden (siehe Abbildung 2.5(a)). Jede Zelle besitzt eine Basisstation (*Base Station, BS*), die für die Anbindung von Mobilstationen an das Mobilfunknetz verantwortlich ist. Kommunikationsverbindungen werden immer über die Basisstation vermittelt und aufgebaut, auch wenn sich die Gesprächspartner direkt nebeneinander befinden. Bewegt sich eine Mobilstation von seiner aktuellen Zelle in eine Nachbarzelle wird die Mobilstation an die Nachbarzelle übergeben. Dieser Vorgang wird als *Handover* bezeichnet, dabei bricht die Verbindung nicht ab.

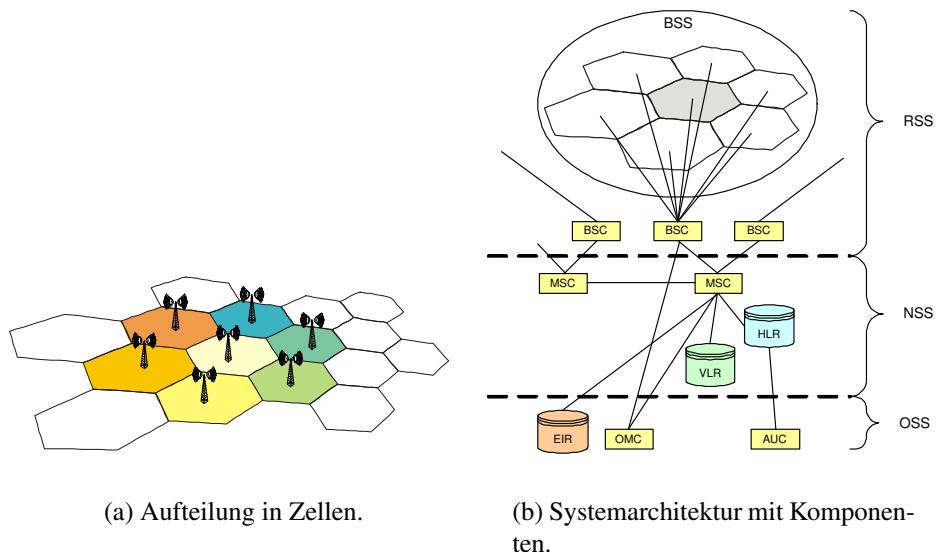


Abbildung 2.5: Vereinfachte Systemarchitektur von zellulären Mobilfunknetzen.

Die Infrastruktur in zellulären Mobilfunknetzen ist hierarchisch aufgebaut (siehe Abbildung 2.5(b)), d.h. die Basisstationen sind an Verwaltungsknoten, die so genannten *Base Station Controller* (BSC) und *Mobile Services Switching Center* (MSC), angebunden. Innerhalb von zellulären Mobilfunknetzen existieren eine Vielzahl von Datenbanken, z.B. das *Home Location Register* (HLR) und *Visitor Location Register* (VLR), die zur Verwaltung dienen. Somit existieren sehr viele Informationen über das Netz, wodurch die Realisierung von Funktionen erheblich erleichtert wird.

### 2.2.2 WLAN

Drahtlose lokale Netze (*Wireless Local Area Network, WLAN*) sind eine Erweiterung von drahtgebundenen lokalen Netzen, um mobile Teilnehmer in vorhandene lokale Netze anzubinden. Im Vergleich zu zellulären Mobilfunknetzen ist die Infrastruktur von WLANs einfach. Durch *Access Points* (AP) werden mobile Teilnehmer an das lokale Netz angebunden, sie erfüllen in etwa die Rolle der Basisstationen in zellulären Mobilfunknetzen (siehe Abbildung 2.6). Ein Access Point ist für einen bestimmten Bereich der durch seine Reichweite gegeben ist verantwortlich.

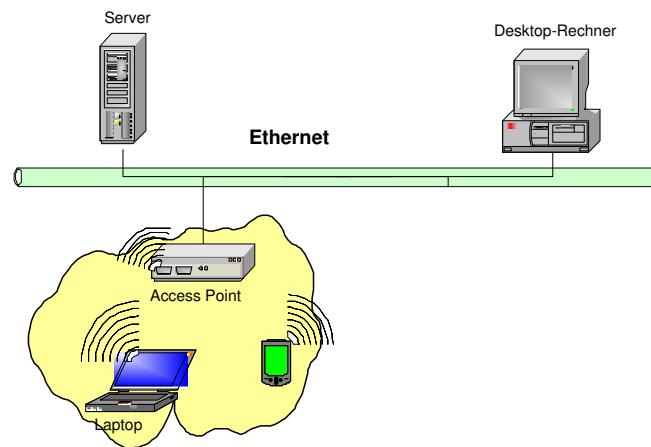


Abbildung 2.6: Anbindung von mobilen Teilnehmern an ein LAN über ein WLAN.

### Vergleich

Ein zusammenfassender Vergleich der Eigenschaften von zellulären Mobilfunknetzen, WLAN und Ad-hoc-Netzen ist in Abbildung 2.7 dargestellt. Der

Hauptunterschied zwischen den drei Netztypen liegt in der inhärenten Mobilität, die bei mobilen multi-hop Ad-hoc-Netzen am stärksten ausgeprägt ist.

	Mobilfunknetz	Wireless LAN	Ad-hoc-Netz
Netztopologie	Fixe Zellen mit fixen Basisstationen	Fixe Access Points	Keine Infrastruktur
Struktur	Statisches Backbone	Anbindung an LAN	Dynamische multi-hop Kommunikation
Umgebung	Gutes Wissen über die Umgebung und Teilnehmer	Gutes Wissen über die Umgebung und Teilnehmer	Unvollständiges Wissen über Umgebung und Teilnehmer
Vorarbeiten	Detaillierte Planung der Netzumgebung	Planung mit einfacher Re-Installation	Spontaner Aufbau ohne Planung

Abbildung 2.7: Zellulare Mobilfunknetze, WLAN und Ad-hoc-Netze im Vergleich.

## 2.3 Netzwerkarchitekturen und Ad-hoc-Netze

Bei der Diskussion von Kommunikationssystemen wird im Allgemeinen ihre Architektur betrachtet, dabei steht zunächst der Ablauf der Kommunikation zwischen zwei Kommunikationspartnern im Mittelpunkt.

Das bekannteste Beispiel für die Beschreibung der Kommunikation über ein Netzwerk ist das ISO/OSI-Referenzmodell [Tan96]. In Abbildung 2.8 ist die Kommunikation zwischen zwei Kommunikationspartnern nach dem ISO/OSI-Referenzmodell dargestellt. Um die Komplexität der Kommunikation handhabbar zu gestalten, ist sie in Schichten oder Ebenen aufgeteilt, die aufeinander aufbauen. Jede Schicht erfüllt eine bestimmte Aufgabe und bietet ihre Dienste an. Dabei bietet eine Schicht ihre Dienste der Schicht über ihr an. Um ihre Dienste zu erbringen benutzt sie die Dienste der nächstniedrigeren Schicht.

Daten werden von einem Benutzer – besser gesagt von der Anwendung des Benutzers – erzeugt, und an den Kommunikationspartner versendet. Für die Kommunikationspartner sieht es so aus, als ob sie direkt miteinander Nachrichten austauschen würden. Dies gilt sogar für alle Schichten, die logisch betrachtet miteinander die gleiche Sprache sprechen. In der Realität interagieren jedoch die Schichten nur vertikal. Die Anwendung übergibt die Daten an die Schicht unter ihr und diese an die nächstniedrigere Schicht. Dabei

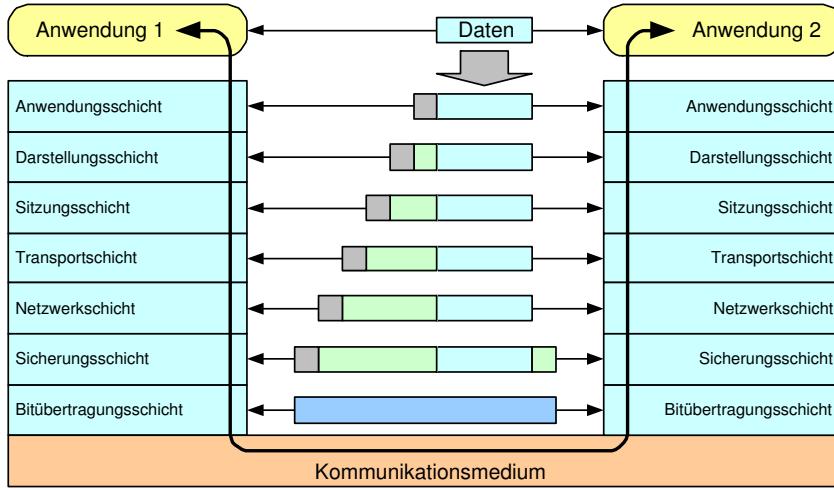


Abbildung 2.8: Kommunikation nach dem ISO/OSI-Referenzmodell.

fügt jede Schicht zu den eigentlichen Benutzerdaten spezielle Informationen hinzu. Auf der untersten Schicht, der so genannten Bitübertragungsschicht, werden die Daten über das zur Verfügung stehende Medium zum Kommunikationspartner übertragen. Beim Kommunikationspartner erfolgt der Transport der Daten von unten nach oben. Dabei werden die eingefügten Informationen auf den entsprechenden Schichten wieder entfernt. Schließlich erreichen die Daten die Anwendung des Kommunikationspartners.

Das ISO/OSI-Referenzmodell hat keine praktische Bedeutung, es wird jedoch für die Beschreibung unterschiedlichster Netzwerkarchitekturen benutzt und besitzt deshalb eine theoretische Bedeutung.

### 2.3.1 Das TCP/IP-Referenzmodell

Das TCP/IP-Referenzmodell stellt die Netzwerkarchitektur des Internets dar. Die Abbildung 2.9 stellt das ISO/OSI-Referenzmodell dem TCP/IP-Referenzmodell gegenüber [Tan96]. Im Vergleich zum ISO/OSI-Referenzmodell fehlen im TCP/IP-Referenzmodell zwei Schichten, nämlich die Sitzungsschicht und die Darstellungsschicht. Die Aufgaben der fehlenden Schichten werden auf der Anwendungsschicht von den eigentlichen Anwendungen erfüllt.

Im Folgenden werden die Funktionen der einzelnen Schichten des TCP/IP-Referenzmodells kurz beschrieben, die Aufgaben sind wie folgt:

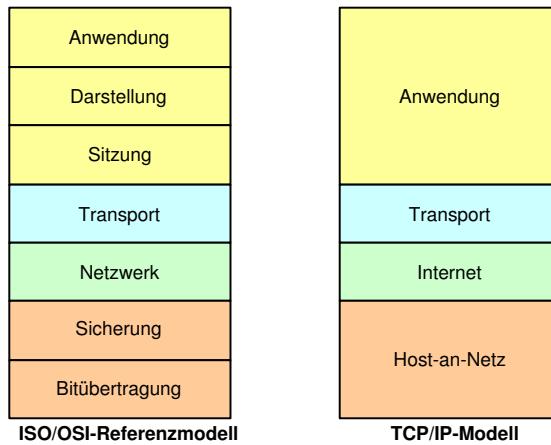


Abbildung 2.9: Das ISO/OSI-Referenzmodell und das TCP/IP-Referenzmodell.

- **Host-an-Netz-Schicht**

Diese Schicht ist im TCP/IP-Referenzmodell nicht genau definiert. Es wird nur gefordert, dass ein Host über ein Protokoll am Netz angeschlossen und in der Lage sein muss, Pakete, die von der Internet-Schicht übergeben werden, die so genannten IP-Datagramme, zu übertragen. Das benutzte Protokoll ist nicht vorgegeben und variiert deshalb von Netz zu Netz. Weit verbreitete Host-an-Netz Techniken sind IEEE 802.3 (Ethernet) für verdrahtete lokale Netze und IEEE 802.11 (WLAN) für drahtlose lokale Netze.

- **Internet-Schicht**

Auf der Internet-Schicht befindet sich das *Internet Protocol (IP)*, das einen verbindungslosen und unzuverlässigen Übertragungsdienst für IP-Datagramme anbietet. Auf dieser Schicht befinden sich auch die wichtigen Dienste für die eindeutige Adressierung von Hosts und Netzen, und das Routing, welches für die Pfadfindung zwischen den Kommunikationspartnern verantwortlich ist.

- **Transportschicht**

Auf der Transportschicht stehen zwei Transportprotokolle zur Auswahl. Das *Transmission Control Protocol (TCP)* bietet einen verbindungsorientierten und zuverlässigen Ende-zu-Ende-Übertragungsdienst für Datenströme an. Es ist das meist benutzte Protokoll im Internet. Das zweite Protokoll auf der Transportschicht ist das *User Datagram Protocol (UDP)*, welches einen verbindungslosen und unzuverlässigen Über-

tragungsdienst für Datagramme anbietet. Die Funktionalität von UDP ist dem von IP sehr ähnlich.

- **Anwendungsschicht**

Auf der Anwendungsschicht sind alle anderen Dienste und Protokolle angesiedelt, die nicht auf den anderen Schichten vorhanden sind. Sie basieren entweder auf TCP oder UDP. Zu den Protokollen, die das verbindungsorientierte TCP verwenden gehören u.a. FTP und HTTP. Beispiele für Protokolle, die auf UDP aufsetzen, sind NFS (Network File System) und RTP (Real Time Transport Protocol).

### 2.3.2 Kommunikation in Ad-hoc-Netzen

Die Kommunikation in mobilen multi-hop Ad-hoc-Netzen gestaltet sich im Vergleich zu Festnetzen viel komplexer. Die Realisierung und der Entwurf von mobilen multi-hop Ad-hoc-Netzen betrifft alle Schichten [RR02] des ISO/OSI-Referenzmodells. So muss die Bitübertragungsschicht mit sich ständig verändernden Verbindungseigenschaften kämpfen. Die Mediumzugriffskontrolle, eine Teilschicht der Sicherungsschicht, muss die Anzahl von Paketkollisionen zu minimieren versuchen und gleichzeitig alle Knoten fair behandeln. Probleme wie Hidden-Stations und Exposed-Terminals (siehe Seite 29) müssen ebenfalls auf dieser Schicht berücksichtigt werden. Die Netzwerkschicht muss Informationen über das Netzwerk sammeln und die Berechnung von günstigen Pfaden erlauben, was durch die Knotenmobilität erschwert wird. Die Anbindung von Ad-hoc-Netzen an vorhandene Netze muss auch auf dieser Schicht realisiert werden. Anwendungen gehen davon aus, dass eine Verbindung zwischen den Kommunikationspartnern wie ein Tunnel funktioniert. Die Behandlung von Fehlern, deren Ursache in der Knotenmobilität oder der schlechten Verbindungseigenschaften liegen werden nicht berücksichtigt.

### Architektur für das mobile Internet

Eine weitergehende Frage, die mit der Netzwerkarchitektur und dem Entwurf von Protokollen behaftet ist, ist die, wie mobile multi-hop Ad-hoc-Netze in die existierende Kommunikationsinfrastruktur eingebettet werden können. Die IETF ist an einem Modell interessiert, das die nahtlose Anbindung von mobilen multi-hop Ad-hoc-Netzen an das Internet erlaubt. Dabei sollen vorhandene offene Internetprotokolle zum Einsatz kommen. In Abbildung 2.10 ist ein Vorschlag für das zukünftige Internet mit der Integration von mobilen Ad-hoc-Netzen dargestellt [CMC99]. Das Modell besteht aus drei Ebenen, wobei eine Ebene das vorhandene Internet und die anderen beiden Ebenen

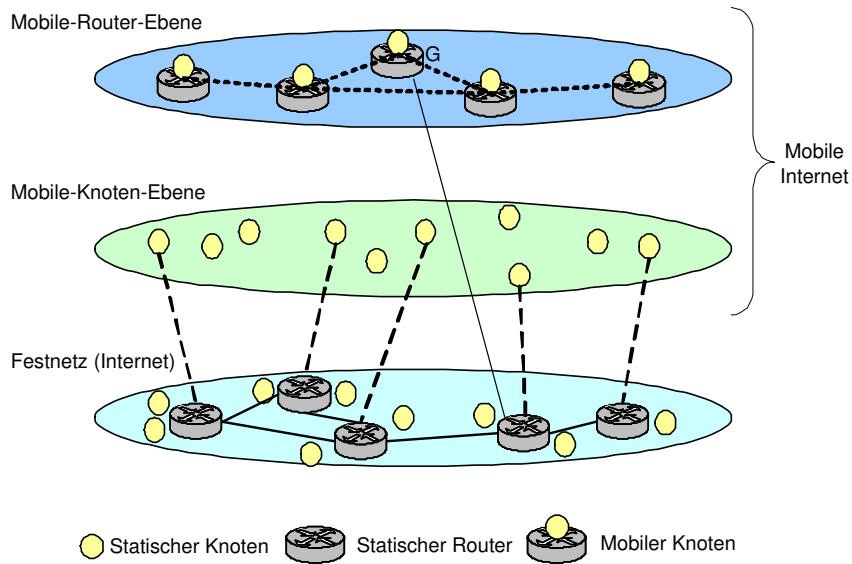


Abbildung 2.10: Architektur für das Mobile Internet.

das zukünftige *Mobile Internet* darstellen. Die Aufgaben der beiden neuen Schichten sind wie folgt:

- **Mobile-Knoten-Ebene**

Auf der Mobile-Knoten-Ebene befinden sich mobile Knoten, die temporär an einen festen Router angebunden sind und dadurch Zugang zum Internet haben. Diese Knoten sind typischerweise nur einen Link von einem festen Router entfernt. Beispielsweise könnte ein Knoten über eine WLAN-Verbindung oder über GSM (GPRS) an das Internet angebunden sein. Die Besonderheit dieser Ebene ist, dass für das Routing das Festnetz verantwortlich ist.

- **Mobile-Router-Ebene**

Die Mobile-Router-Ebene besteht aus Knoten, die die Funktion eines Hosts und eines Routers gleichzeitig erfüllen (*mobiler Knoten*). Die Knoten auf dieser Ebene bilden typischerweise autonome Netze, in denen die Quelle und Senke des Kommunikationsverkehrs enthalten sind. Jedoch können die autonomen Netze auch an das Internet angebunden sein, dann übernimmt einer der Knoten die Funktion des Gateways. Die IETF nimmt an, dass Mobile-IP Foreign-Agents diese Anbindung übernehmen könnten. Diese Möglichkeit ist in Abbildung 2.10 durch die Verbindung des mobilen Knotens G an das Internet dargestellt.

### Horizontale vs. vertikale Kommunikation

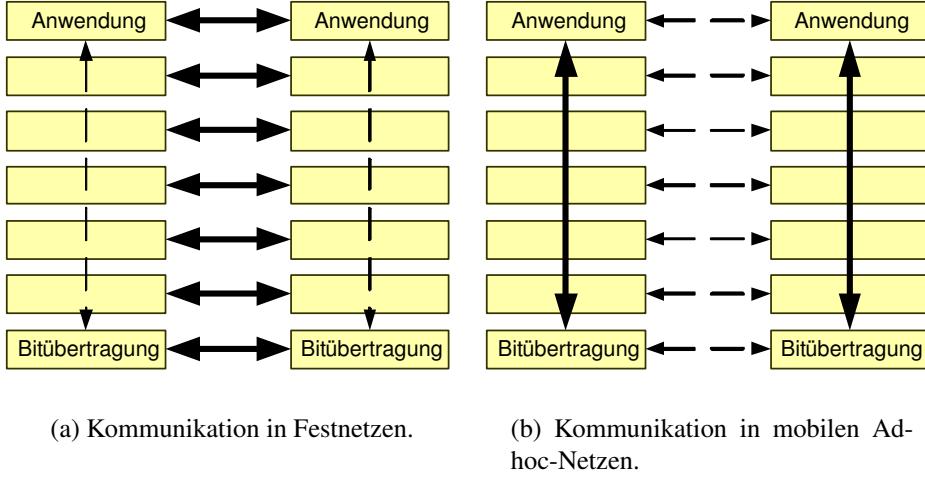


Abbildung 2.11: Horizontale vs. vertikale Kommunikation.

Sowohl das ISO/OSI-Referenzmodell als auch das TCP/IP-Referenzmodell basieren auf der strikten Trennung in *horizontale* und *vertikale* Kommunikation (siehe Abbildung 2.11(a)). Diese Aufteilung ist für Festnetze, in denen weder die Netzwerkinfrastruktur wie Router noch die Hosts mobil sind, vorteilhaft. Sie minimiert die Last der Router, was zu einer besseren Leistung führt, vereinfacht die benötigten Protokolle auf jeder Schicht und erlaubt den Austausch von Protokollen und Implementierungen, solange die Schnittstellen zwischen den einzelnen Schichten unverändert bleiben. Der Nachteil dieser Aufteilung ist die benötigte hohe Last, die aus Abbildung 2.8 klar wird. Die von den einzelnen Schichten hinzugefügten Informationen sind nicht unerheblich. Eine weitere Schwäche ist durch die Unabhängigkeit der einzelnen Schichten gegeben. Diese Anforderung beschränkt die Protokolle auf den einzelnen Schichten auf allgemeine Annahmen, sie sind nicht in der Lage auf wechselnde Besonderheiten einzugehen. Beispielsweise hat die Transportschicht keine Information, welches Verfahren auf der Mediumzugriffsschicht eingesetzt wird, obwohl das Verhalten und die Leistung des Dienstes auf der Transportschicht indirekt vom eingesetzten Mediumzugriffsverfahren beeinflusst wird [GV02, GHB03].

Die strikte Trennung in horizontale und vertikale Kommunikation ist in mobilen multi-hop Ad-hoc-Netzen nicht vorteilhaft [MC98]. Die höheren Schichten (Transportschicht, Anwendungsschicht) sind ohne Informationen aus den unteren Schichten nicht in der Lage ihre volle Leistung zu erbringen. Ein weiteres Problem ist durch die zur Verfügung stehenden Ressourcen gegeben. In [CMC99, MC98] wird deshalb gefordert die strikte Trennung in horizontale

und vertikale Kommunikation in mobilen multi-hop Ad-hoc-Netzen aufzuweichen und die vertikale Kommunikation im Gegensatz zur horizontalen zu stärken (siehe Abbildung 2.11(b)). Hierdurch wären vorhandene Protokolle in der Lage auch in diesen Netzen eine akzeptable Leistung zu erbringen. Tatsächlich wird in einigen Untersuchungen und Ansätzen davon ausgegangen, dass man in den höheren Schichten Zugriff auf Informationen aus den unteren Schichten hat. Der Zugriff auf die Informationen ist entweder explizit durch zusätzliche Schnittstellen realisiert, oder sie wird per Piggybacking mit den eigentlichen Daten von der unteren Schicht zu der höheren Schicht weitergeleitet.

## 2.4 Technologien für mobile Ad-hoc-Netze

Unterschiedliche drahtlose Technologien bieten sich für die Realisierung von Ad-hoc-Netzen an. Im folgenden Abschnitt wird die inzwischen weit verbreitete Technik für drahtlose lokale Netze vorgestellt. Anschließend wird Bluetooth beschrieben, das zuerst als einfacher Ersatz für Kabel entworfen wurde, jedoch auch das Potenzial für die Realisierung von Ad-hoc-Netzen besitzt.

### 2.4.1 IEEE 802.11 und Verwandte

Die IEEE 802.11 [Iee97] spezifiziert einen Standard für drahtlose lokale Netze. Ähnlich zu anderen IEEE 802.x Standards, z.B. IEEE 802.3 (Ethernet) und IEEE 802.5 (Token Ring), wird die Mediumzugriffsschicht (MAC Layer) und die Bitübertragungsschicht (Physical Layer) definiert.

Die erste Version des Standards spezifizierte Datenraten von 1 Mbit/s und 2 Mbit/s. Der Standard wurde 1999 erweitert und die Datenrate auf 11 Mbit/s angehoben, diese Erweiterung ist im Standard IEEE 802.11b [Iee99] beschrieben. Die Reichweiten der einzelnen Knoten variieren zwischen 30 m in Gebäuden und 300 m im Freien. Die praktische Reichweite hängt jedoch sehr stark von der Beschaffenheit und dem Einfluss der Umgebung ab. Die IEEE 802.11b Version des Standards ist inzwischen sehr weit verbreitet.

Im Folgenden wird die Architektur von IEEE 802.11 detailliert besprochen, da sowohl die Ergebnisse dieser Arbeit als auch die meisten anderen Forschungsarbeiten im Bereich der Ad-hoc-Netze auf Simulationen mit Knoten gemäß dieser Spezifikation beruhen.

## Architektur

Im Standard werden zwei unterschiedliche Betriebsmodi beschrieben. Der infrastrukturbasierte Betrieb und der Betrieb im Ad-hoc-Modus. Im ersten Fall besteht das Netzwerk aus mobilen Endgeräten, die über so genannte *Access Points* an ein lokales Netzwerk angebunden werden. Im Ad-hoc-Modus wird für die Kommunikation kein Access Point benötigt. In diesem Fall können die Knoten direkt miteinander kommunizieren.

In Abbildung 2.12 sind die Komponenten eines drahtlosen Netzes gemäß IEEE 802.11 mit allen Komponenten dargestellt. Im Beispiel besteht das Netzwerk aus zwei so genannten *Basic Service Sets* (BSS1 und BSS2), die den Abdeckungsbereich eines drahtlosen Netzwerks kennzeichnen. In einem BSS existiert ein Access Point der die mobilen Stationen an das lokale Netzwerk anbindet; im Beispiel wird ein Ethernet-LAN angenommen. Die beiden BSS sind über das so genannte *Distribution System* miteinander verbunden und bilden einen *Extended Service Set*. Dadurch sind mobile Knoten in unterschiedlichen BSS in der Lage miteinander zu kommunizieren.

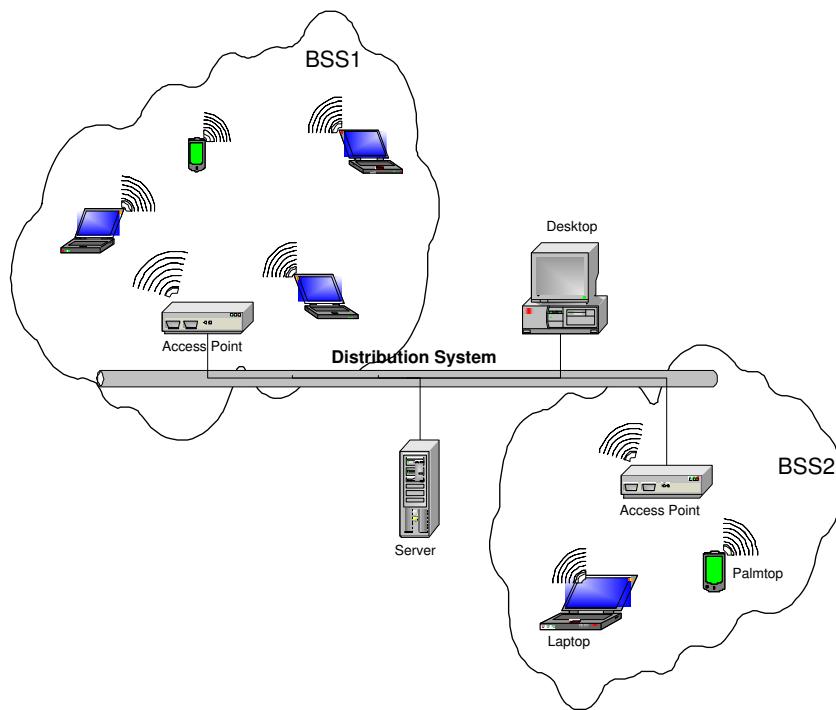


Abbildung 2.12: Architektur von IEEE 802.11. Dargestellt sind zwei Basic Service Sets mit mehreren mobilen Knoten. Die beiden BSS sind über das Distribution System verbunden und bilden einen Extended Service Set.

Die Access Points spielen eine zentrale Rolle, da sie wie im Beispiel angenommen die Pakete von IEEE 802.11 nach IEEE 802.3 übersetzen müssen. Der zugehörige Protokollstack ist in Abbildung 2.13 dargestellt.

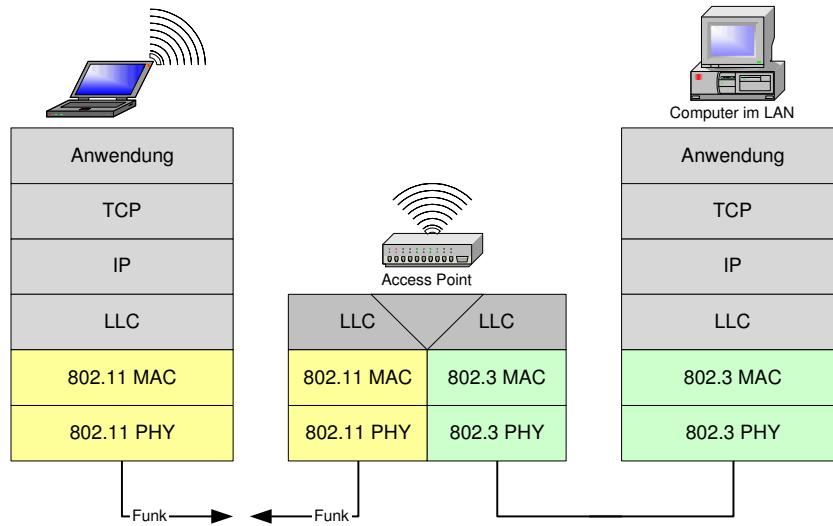


Abbildung 2.13: Der Protokollstack von IEEE 802.11 und die Anbindung an ein LAN.

Wenn mehrere Access Points existieren können die mobilen Stationen zwischen den Access Points wechseln, sodass auch das Roaming zwischen mehreren BSS möglich ist. Ähnlich zum *Handover* von zellulären Mobilfunknetzen meldet sich eine Station dabei in seiner aktuellen BSS ab und meldet sich in der neuen BSS an.

Im Ad-hoc-Modus bildet eine Menge von mobilen Stationen einen BSS. Die mobilen Stationen im gleichen BSS können direkt miteinander kommunizieren. Für diesen Betrieb ist aber weder ein Routing noch der Austausch von Topologieinformationen vorgesehen. Aus diesem Grund können mobile Stationen in unterschiedlichen BSS im Ad-hoc-Modus nicht miteinander kommunizieren.

Die Spezifikation definiert neben diesen eigentlichen Funktionen auch noch Managementfunktionen. Einerseits wird die Zuordnung von Stationen an Access Points geregelt, wodurch das Roaming von Stationen möglich wird, andererseits werden Funktionen für Authentifizierung, Verschlüsselung, Power-Management und Synchronisation mit einem Access Point definiert. Neben diesen beiden Unterschicht-Managern existiert ein Management für die Station, das für Funktionen in höheren Schichten verantwortlich ist. Dazu gehören die Überbrückung zwischen den drahtlosen und drahtgebundenen Teil-

netzen und die Kommunikation zwischen Access Points über das *Distribution System*.

### Bitübertragungsschicht

Die Bitübertragungsschicht ist in zwei Unterschichten aufgeteilt: das *Physical-Layer-Convergence-Protocol (PLCP)* und die *Physical-Medium-Dependent (PMD)* Unterschicht. Die PLCP-Unterschicht ist für die Bereitstellung eines Signals für die Kontrolle des Mediums und eines von der eigentlichen Übertragung unabhängigen Dienstzugangspunktes für den Zugriff auf die Bitübertragung verantwortlich. Die PMD-Unterschicht ist für die Modulation und Demodulation der eigentlichen Signale zuständig.

Für die Bitübertragungsschicht sind drei unterschiedliche Verfahren vorgesehen. Eins davon basiert auf Infrarot, die beiden anderen auf Funk.

Die infrarot-basierte Übertragung verwendet Frequenzen mit Wellenlängen um 850-950 nm, welche lizenfrei nutzbar ist. Da Punkt-zu-Mehrpunkt-Kommunikation erforderlich ist – damit ein Access Point mehrere Stationen bedienen kann – ist kein Sichtkontakt nötig. Der Abstand zwischen den einzelnen Knoten darf maximal 10 m betragen.

Für die Funkübertragung definiert der Standard zwei Verfahren. Das erste ist das Frequenzsprungverfahren (Frequency Hopping Spread Spectrum, FHSS). Mit diesem Verfahren ist die Koexistenz mehrerer Netze, die unterschiedliche Sprungsequenzen und somit auch unterschiedliche Frequenzen verwenden, möglich. Das zweite Verfahren ist die Bandspreizung (Direct Sequence Spread Spectrum, DSSS). Hierbei wird die Spreizung durch unterschiedliche Codes und nicht durch unterschiedliche Frequenzen erreicht. Das in IEEE 802.11 eingesetzte Verfahren ist unter dem Namen *Barker-Code* bekannt. Dessen Hauptmerkmale sind die Unempfindlichkeit gegen Mehrwegeausbreitung und Robustheit gegen Interferenzen.

Um einen weltweiten Betrieb zu gewährleisten arbeiten beide Funkübertragungsverfahren im freien ISM-Band mit 2,4 GHz.

### MAC-Schicht

Zu den Aufgaben der MAC-Schicht gehört die Mediumzugriffskontrolle, das Roaming, die Authentifizierung und Power-Management. Die Spezifikation definiert zwei unterschiedliche Arten der Datenübertragung, die asynchrone und die synchrone (zeitgebundene) Datenübertragung, wobei letztere Variante nur optional ist. Im infrastrukturbasierten Modus sind beide Übertragungsarten möglich. Im Ad-hoc-Betrieb wird nur die asynchrone Datenübertragung

möglich. Hier fehlt die zentrale Einheit für die Synchronisation. Weiterhin ist die Übertragung von Multicast- und Broadcast-Paketen möglich. Der angebotene Dienst arbeitet nach dem Best-Effort-Prinzip, d.h. es werden keinerlei Garantien für die Übertragungszeiten gegeben.

Die MAC-Schicht von IEEE 802.11 kennt drei unterschiedliche Zugriffsverfahren auf das Medium.

- **Zugriff auf der Basis von CSMA/CA**

Das *Carrier Sense Multiple Access with Collision Avoidance* ist ein Verfahren mit zufälligem Zugriff auf das Medium, Medium-Überprüfung und Kollisionsvermeidung. Beim CSMA/CA überprüft ein Knoten vor dem Zugriff das Medium. Wenn es frei sein sollte fängt der Knoten mit der Übertragung von Daten an. Ansonsten muss er eine zufällige Zeit warten. Das Intervall aus dem die Wartezeit gewählt wird, wird durch den *Backoff-Timer* festgelegt. Wenn der Knoten danach wiederum nicht senden konnte muss er erneut eine zufällige Zeit warten. Hierbei erhöht sich das Intervall für die Wartezeiten exponentiell. Deshalb heißt dieses Verfahren *Exponential-Backoff*.

- **Zugriff mit CSMA/CA und der Erweiterung von RTS/CTS**

Bei dieser Erweiterung geht es hauptsächlich um die Lösung des so genannten *Hidden-Terminal-Problems*, das für viele Paketkollisionen verantwortlich ist. Dieses Problem tritt auf wenn zwei Sender an den gleichen Empfänger Daten übertragen, wobei sie sich jedoch nicht in ihrer gegenseitigen Reichweite befinden. In Abbildung 2.14 a) entsteht eine Kollision beim Empfänger  $D$ , da die beiden Sender  $S_1$  und  $S_2$  sich außerhalb ihrer gegenseitigen Funkreichweite befinden. In Abbildung 2.14 b) ist das *Exposed-Terminal-Problem* dargestellt. Während  $S_1$  mit  $D_1$  kommuniziert darf  $S_2$  nicht mit  $D_2$  kommunizieren, obwohl  $D_1$  sich außerhalb der Funkreichweite von  $S_2$  befindet.

Beim CSMA/CA mit der Erweiterung von RTS/CTS schickt der Sender ein RTS-Paket (Ready To Send) an den Empfänger, der dies durch ein CTS-Paket (Clear To Send) bestätigt (siehe Abbildung 2.15). Dabei übermittelt der Sender im RTS-Paket die Adressen des Senders und des Empfängers und die Übertragungszeit der zu übertragenden Daten. Das CTS-Paket vom Empfänger enthält den Sender und ebenfalls die Übertragungszeit. Durch diesen Informationsaustausch sind sowohl die Nachbarn des Senders, erste Zeile in Abbildung 2.15, und die Nachbarn des Empfängers, letzte Zeile in Abbildung 2.15, über die bevorstehende Übertragung und der benötigten Übertragungszeit informiert. Dadurch sind alle Nachbarn verpflichtet das Ende der Übertragung zwischen

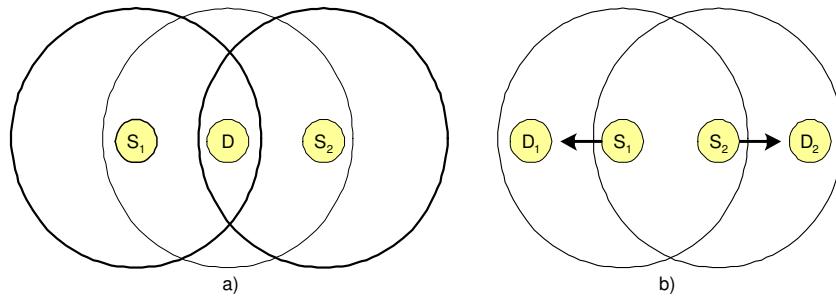


Abbildung 2.14: Hidden- und Exposed-Terminals.

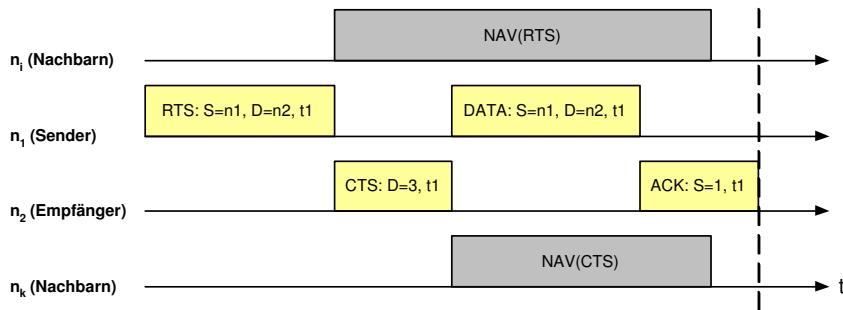


Abbildung 2.15: Schematischer Ablauf des Medienzugriffs mit CSMA/CA und der Erweiterung von RTS/CTS.

dem Sender und dem Empfänger abzuwarten. Am Ende der Übertragung erlöst der Empfänger seine Nachbarn durch das Aussenden eines ACK-Pakets, wodurch auch der Sender Gewissheit über den korrekten Empfang der Daten erlangt.

- **Zugriff mit Polling**

Beim Mediumzugriff mit Polling existiert immer ein Access Point, der den Zugriff steuert. Hierzu wird die Zeit in so genannte Super-Frames aufgeteilt, der aus einer wettbewerbsfreien Phase und einer Wettbewerbsphase besteht. In der Wettbewerbsphase können die beiden verteilten Verfahren verwendet werden. In der wettbewerbsfreien Phase fragt der Koordinator – üblicherweise der Access Point – die einzelnen mobilen Stationen ab und bedient sie nacheinander.

Es ist zu bemerken, dass der verteilte und der zentral koordinierte Zugriff auf das Medium gleichzeitig in einer Zelle verwendet werden kann. Dies ist

von besonderem Interesse, da hierdurch Knoten, die im Ad-hoc-Modus arbeiten, gleichzeitig auch an das infrastrukturbasierte Teil des Netzes angebunden werden können.

### Verwandte und Nachfolger von IEEE 802.11

Der große Erfolg von IEEE 802.11b führte zu weiteren Erweiterungen des IEEE 802.11 Standards, die noch höhere Datenraten ermöglichen. Die Tabelle in Abbildung 2.16 gibt einen Überblick über die einzelnen Standards der IEEE 802.11x Familie. Aus der Menge der Standards sollen zwei näher betrachtet werden.

Standard	Beschreibung
802.11	Standard für drahtlose Netze im 2,4 GHz Bereich mit 1-2 Mbit/s.
802.11a	Erweiterung für hohe Datenraten von 6-54 Mbit/s im 5 GHz Bereich.
802.11b	Erweiterung von IEEE 802.11 DSSS für hohe Datenraten von 5,5-11 Mbit/s im 2,4 GHz Bereich.
802.11c	Spezifikation für WLAN spezifisches Bridging auf MAC-Layer.
802.11e	Quality of Service (QoS)
802.11f	Protokoll für die Kommunikation zwischen Access Points.
802.11g	Erweiterung für hohe Datenraten von 802.11b bis 54 Mbit/s.
802.11h	Dynamische Kanalwahl und Kontrolle der Übertragungsleistung.
802.11i	Neues Sicherheitsverfahren zur Ersetzung des schwachen Wired Equivalent Privacy (WEP).

Abbildung 2.16: Übersicht der IEEE 802.11x Standards für drahtlose Netze.

IEEE 802.11g ist eine Erweiterung zum IEEE 802.11b Standard. Sie arbeitet im gleichen Frequenzband und soll eine höhere Datenrate anbieten. Sie wurde insbesondere für den Übergang von IEEE 802.11b zu IEEE 802.11a entworfen, um Unternehmen einen gewissen Investitionsschutz zu gewährleisten. Erwartete Datenraten liegen zwischen 20 und 54 Mbit/s.

Der Nachfolger von IEEE 802.11b soll IEEE 802.11a werden. Er bietet unterschiedliche Datenraten bis 54 Mbit/s. Dieser Standard arbeitet im Gegensatz zu den anderen Mitgliedern von IEEE 802.11x im 5 GHz Band. Es besteht daher keine Kompatibilität zwischen IEEE 802.11b und IEEE 802.11a.

Auf der Bitübertragungsschicht setzt 802.11a das so genannte OFDM (Or-

thogonal Frequency Division Multiplexing) ein. Hierbei werden gleichzeitig Daten auf mehreren Frequenzen übertragen. Insgesamt gibt es 52 Frequenzen, von denen 48 für Daten- und 4 für Steuernachrichten verwendet werden. Ein weiterer Vorteil von IEEE 802.11a ist es, dass die verwendete Technik OFDM zum europäischen HIPERLAN/2 Standard kompatibel ist. Auf der Mediumzugriffsschicht setzt IEEE 802.11a wie die anderen beiden Verwandten IEEE 802.11 und IEEE 802.11b CSMA/CA ein.

Für alle Mitglieder der IEEE 802.11 Familie gilt, dass die zu erwartende Netto-Datenrate etwa bei 45-55% der angegebenen Brutto-Datenrate liegt. Beispielsweise kann man bei IEEE 802.11b mit einer typischen Netto-Datenrate von 5–6 Mbit/s rechnen.

#### 2.4.2 Bluetooth und IEEE 802.15

Die Firma Ericsson führte 1994 eine Studie für eine Mehrzweck-Kommunikationstechnologie durch. Auf der Grundlage dieser Studie bildeten fünf große Unternehmen (Ericsson, Intel, IBM, Nokia, Toshiba) 1998 ein Industriekonsortium, dem in kürzester Zeit mehr als tausend Mitglieder angehörten.

Die eigentliche Idee hinter Bluetooth<sup>1</sup> war, Kabel durch eine Funkkommunikation zu ersetzen, also gewissermaßen eine Funkschnittstelle für Geräte wie PDAs, Tastaturen und Mäuse zu etablieren. Dies sollte die Benutzung solcher Geräte und insbesondere den Austausch von Daten zwischen PDA, Mobiltelefon und Desktoprechnern erleichtern. Technisch gesehen sollte Bluetooth der Nachfolger von IrDa werden. Dieser bietet eine ähnliche Funktion wie Bluetooth, hatte aber aufgrund der Benutzung von Infrarot und dem dadurch bedingten Sichtkontakt mit vielen Schwierigkeiten zu kämpfen.

Bluetooth ist ein Kurzstreckenkommunikationsstandard, der wie das IEEE 802.11 im lizenzenfreien 2,4 GHz ISM (*Industrial Scientific Medical*) Funkbereich arbeitet. Die Spezifikation deckt Entferungen von 10-100 m ab. Mit zunehmender Überbrückungsstrecke steigt die benötigte Sendeleistung. Der Haupteinsatz von Bluetooth ist für kurze Strecken bis 10 m gedacht. Bluetooth wurde speziell für den Einsatz in kleinen mobilen Geräten, die ihre Energie aus Batterien oder Akkus beziehen, entworfen. Deshalb sollten alle Komponenten auf einen Chip passen und sehr wenig Energie verbrauchen.

Im Gegensatz zu anderen Funktechniken definiert die Bluetooth-Spezifikation nicht nur die Bitübertragungsschicht und die Mediumzugriffsschicht, sondern auch andere Schichten, so etwa die Sicherungsschicht und die Anwendungs-

---

<sup>1</sup> Der Name ist nach dem dänischen König Blåland benannt, der im 10. Jahrhundert Dänemark und Norwegen vereinigte.

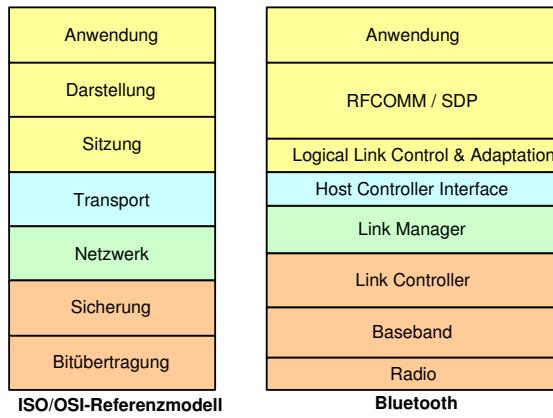


Abbildung 2.17: Protokollstack von Bluetooth im Vergleich zum ISO/OSI-Referenzmodell nach [BS01].

schicht. Abbildung 2.17 zeigt eine mögliche Gegenüberstellung des ISO/OSI-Referenzmodells zum Bluetooth-Protokollstack [BS01]. Es ist offensichtlich, dass die einzelnen Bluetooth-Schichten nicht genau in eine Schicht des OSI-Modells passen.

Eine alternative Darstellung des Bluetooth-Protokollstacks ist in Abbildung 2.18 zu sehen. Die Abbildung folgt der Zuordnung gemäß IEEE 802.15.1, das einen Standard für *Wireless Personal Area Networks (WPAN)* darstellt und auf Bluetooth aufbaut. In Abschnitt 2.4.2 wird der Zusammenhang zwischen Bluetooth und IEEE 802.15.1 näher erläutert.

## Architektur

Die Basisarchitektur von Bluetooth bildet das so genannte *Piconetz*. Ein Piconetz besteht aus einem Master-Knoten und bis zu sieben aktiven Slave-Knoten. Außer diesen Knoten kann eine größere Anzahl (bis 255) von passiven Knoten an der Kommunikation teilnehmen. Zusätzlich wird in der Spezifikation das *Scatternetz* beschrieben, das aus dem Zusammenschluss mehrerer Piconetze entsteht. Die Spezifikation definiert aber keine Routingfunktion, um Daten zwischen mehreren Piconetzen weiterzuleiten. Diese Aufgabe wird den höheren Schichten überlassen. Ein Knoten kann in mehreren Piconetzen Slave oder in einem Piconet Master und in einem zweiten Piconet Slave sein. Ein Knoten, der in mehreren Piconetzen teilnimmt, muss sich mit allen Piconetzen synchronisieren. Abbildung 2.19 stellt die möglichen Beziehungen zwischen Knoten dar.

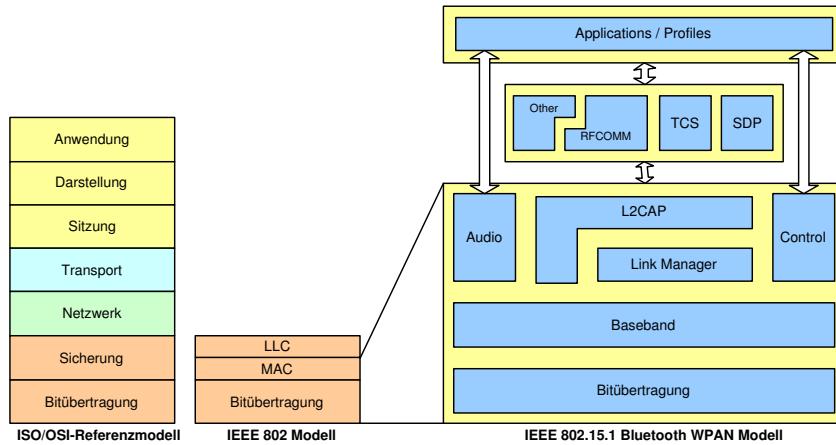


Abbildung 2.18: Protokollstack von IEEE 802.15.1 und Bluetooth in Anlehnung an das ISO/OSI-Modell. Im Standard IEEE 802.15.1 wird aufbauend auf Bluetooth ein Standard für WPANs spezifiziert, wobei ähnlich zu den anderen IEEE 802.x Standards die Bitübertragungsschicht und die Medium-zugriffsschicht definiert werden.

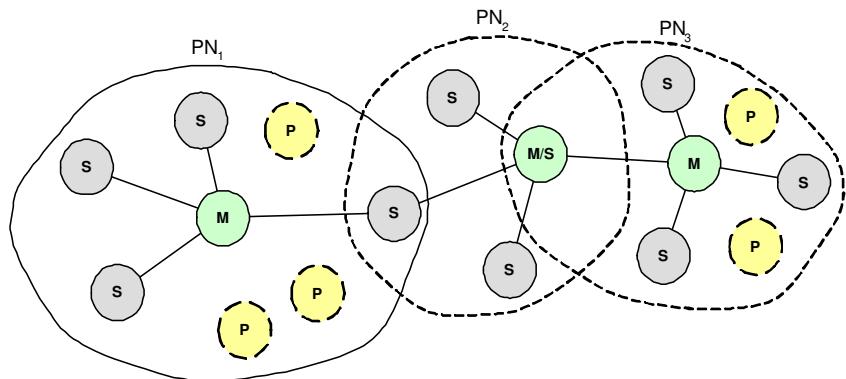


Abbildung 2.19: In der Abbildung ist ein Bluetooth Scatternet bestehend aus drei Piconetzen PN<sub>1</sub>, PN<sub>2</sub> und PN<sub>3</sub> dargestellt. Das Piconetz PN<sub>1</sub> besteht aus einem Master (M), vier Slaves (S) und drei passiven Knoten (P). Die passiven Knoten sind nicht aktive Kommunikationsteilnehmer, sind aber mit dem Piconetz synchronisiert. Ein Slave Knoten befindet sich sowohl in PN<sub>1</sub> als auch in PN<sub>2</sub>. Dieser muss sich daher mit beiden Piconetzen synchronisieren. Der Master von PN<sub>2</sub> ist gleichzeitig auch ein Slave in PN<sub>3</sub>. Die Master sind für die Kontrolle jeglicher Nachrichten verantwortlich.

## Bitübertragungsschicht

Auf der Bitübertragungsschicht setzt Bluetooth ein Frequenzsprungverfahren mit 1600 Sprüngen pro Sekunde in Kombination mit einem Zeitmultiplexverfahren ein. Die Zeit zwischen zwei Sprüngen wird als Slot bezeichnet und beträgt  $625 \mu\text{s}$ , d.h. jedes Slot verwendet eine andere Frequenz. Dabei werden die zur Verfügung stehenden Frequenzen gleichverteilt verwendet. Die Sprungsequenz wird vom Master vorgegeben, und alle Knoten mit der gleichen Sprungsequenz bilden ein Piconetz.

Bluetooth ist eigentlich für kurze Strecken ausgelegt. Es können aber auch Entfernung von 10 m, 30 m und 100 m abgedeckt werden, die entsprechend hohe Übertragungsleistungen erfordern. Da der Standard für mobile Geräte gedacht ist, spielt der Energieverbrauch eine wichtige Rolle. Aus diesem Grund sind mehrere Zustände, die wenig Energie verbrauchen, definiert. Ein Knoten, der sich in einem dieser Zustände befindet, hört eine spezielle Untermenge der Frequenzen, die so genannten Weckfrequenzen, ab. Dadurch ist ein Knoten einerseits in der Lage sich weiterhin mit dem Piconetz zu synchronisieren, andererseits kann er aufwachen, wenn es erforderlich ist.

Ein Piconetz kann von jedem beliebigen Bluetooth-Gerät aufgebaut werden, es besteht keine Unterscheidung zwischen den einzelnen Teilnehmern. Der erste Knoten, der eine Verbindung aufbaut wird der Master des Piconetzes.

## MAC-Schicht

Die Aufgaben, die auf der MAC-Schicht des ISO/OSI-Referenzmodells zu finden sind, befinden sich bei Bluetooth auf verschiedenen Schichten. Diese Funktionen werden im Folgenden beschrieben, wobei teilweise auch Funktionen behandelt werden, die über die Aufgaben der MAC-Schicht hinausgehen.

**Baseband:** Zu den Aufgaben des Basebands gehört die Bereitstellung von Verbindungen zwischen zwei Bluetooth Geräten. Bluetooth bietet zwei unterschiedliche Verbindungsarten an:

- **Synchrone-verbindungsorientierte Verbindung**

Eine synchrone-verbindungsorientierte Verbindung (Synchronous Connection Oriented Link, SCO) stellt eine Punkt-zu-Punkt-Verbindung zwischen dem Master und einem Slave her. Für diese Verbindungen reserviert der Master Übertragungszeiten. Dies sorgt das Qualitätsanforderungen erfüllt werden können. Diese Verbindungsart ist hauptsächlich für die Übertragung von Sprache, wie sie bei der Telefonkommunikation erforderlich ist, geeignet.

- **Asynchrone-verbindungslose Verbindung**

Die asynchrone-verbindungslose Verbindung (Asynchronous Connectionless Link, ACL) unterliegt keinen Qualitätsanforderungen und ist für beliebige Datenkommunikation geeignet.

Bluetooth kann entweder eine ACL-Verbindung, drei parallele SCO-Verbindungen oder parallel eine ACL-Verbindung und eine SCO-Verbindung aufrechthalten. Die Datenrate für SCO-Verbindungen beträgt 64 KBit/s. ACL-Verbindungen können mit unterschiedlichen Datenraten laufen. Die Datenraten für symmetrische Verbindungen können bis 433,9 KBit/s und asymmetrische Verbindungen bis 723,2 KBit/s in eine Richtung, und 57,6 Kbit/s in die andere Richtung betragen.

Aus der obigen Ausführung lässt sich klar erkennen, dass Bluetooth auf die Sprachübertragung einen besonderen Wert legt. Die SCO-Verbindungen sind speziell für die Sprachübertragung entworfen. Um der Güteanforderung von Sprachübertragung gerecht zu werden, wurde im Protokollstack für die Sprachübertragung ein besonderer Weg gewählt. Abbildung 2.18 stellt dies deutlich dar. Anwendungen, die Sprache übertragen, können direkt auf die Audiokomponente zugreifen.

**Link Manager:** Der *Link Manager* ist für den Auf- und Abbau von Verbindungen zwischen Bluetooth-Geräten verantwortlich. Hierunter fällt auch das An- und Abmelden von Knoten in einem Piconetz und das Aushandeln von Verbindungsparametern für Sprach- und Datendienste. Er bietet auch die erforderlichen Funktionen für die Authentifizierung der Geräte an. Weiterhin gehört die Kontrolle der Energiesparmodi – Park, Hold und Sniff – eines Knotens in die Verantwortlichkeit des Link Managers. Ein Knoten, der sich in einem der Energiesparmodi befindet, ist weiterhin mit dem Master synchronisiert, verbraucht jedoch weniger Energie. Die Link Manager mehrerer Bluetooth-Geräte nutzen das *Link Management Protocol (LMP)* zur Kommunikation. Die Bluetooth-Spezifikation definiert nur die LMP-Nachrichten, jedoch nicht wie diese implementiert werden.

Eine Besonderheit, die hier nicht unerwähnt bleiben soll, ist, dass Bluetooth eine einfache Möglichkeit des Rollentausches zwischen Master und Slave kennt. Normalerweise wird der Knoten, der die Kommunikation anstößt, zum Master und leitet die Kommunikation. Es kann jedoch in einigen Situationen von Vorteil sein, dass nach dem Verbindungsaufbau die Rollen getauscht werden. Beispielsweise könnte die Verbindung zwischen einem mobilen Gerät und einem Arbeitsplatzrechner vom mobilen Gerät angestoßen worden sein. Es ist jedoch vom Energieverbrauch her sinnvoller wenn der Arbeitsplatzrechner die Rolle des Masters übernimmt, da er keine Schwierigkeit mit der Energieversorgung besitzt.

Da die Funkkommunikation von Haus aus den Schwachpunkt der leichten Abhörbarkeit besitzt, bietet Bluetooth diverse Funktionen für eine sichere Kommunikation. Die Spezifikation definiert Funktionen für die Authentifizierung und Verschlüsselung auf der MAC-Schicht. Die Hauptfunktion ist die Authentifizierung auf der Basis einer *Challenge-Response*-Methode die einseitig oder gegenseitig durchgeführt werden kann. Weiterhin ist ein Generator für die Berechnung von Sitzungsschlüsseln definiert.

Die Sicherheitsvorkehrungen der MAC-Schicht sollten jedoch auf höheren Schichten ergänzt werden, da die hier verwendeten Schlüssellängen mit 40 und 64 Bit keinen starken Schutz gewähren. Es ist anzumerken, dass entsprechend der Spezifikation diese Sicherheitsmaßnahmen nur dem Aufbau von so genannten sicheren Domänen zwischen den Geräten dienen und deshalb keine grundlegende Schwäche darstellen.

**Logical Link Control Adaptation Protocol (L2CAP):** Das *Logical Link Control Adaptation Protocol (L2CAP)* bietet Anwendungen und Protokollen höherer Schichten den Aufbau von ACL-Verbindungen an. Wie schon erwähnt werden Audioverbindungen speziell behandelt und gehören nicht zu der Verantwortlichkeit der L2CAP-Schicht. Zu den weiteren Aufgaben von L2CAP gehört das Multiplexen von mehreren Verbindungen auf eine ACL-Verbindung, die Aufteilung von großen Paketen auf die maximale Übertragungsgröße (Maximum Transmission Unit, MTU) eines Pakets und später die Zusammensetzung der Fragmente zum eigentlichen Paket.

## Bluetooth Profile

Die Bluetooth-Spezifikation definiert nicht nur Protokolle zur Datenkommunikation, sondern auch unterschiedliche Anwendungen. In der Spezifikation werden diese Anwendungen als *Profile* bezeichnet, die in eigenen Dokumenten detailliert beschrieben werden. Hierdurch soll die Zusammenarbeit von Produkten unterschiedlicher Hersteller sichergestellt werden. Es gibt eine Reihe von Profilen, die zusammen mit der Bluetooth-Spezifikation veröffentlicht wurden. Zusätzlich zu diesen Standard-Profilen sind noch viele Profile entwickelt worden, die teilweise sehr anwendungsabhängig sind und von einem bestimmten Hersteller stammen. Aus der Menge der Profile sollen hier einige wichtige beschrieben werden.

- **Das Generic Access Profile**

Das *Generic Access Profile (GAP)* ist die Grundlage aller anderen Profile und beschreibt den Aufbau einer Verbindung. Es beschreibt alle

Anforderungen, die für diese Grundfunktion benötigt werden, dazu gehören alle Anforderungen, die mit der Verbindung, dem Gerät und dem Diensten zu tun haben.

- **Serial Port Profile**

Das *Serial Port Profile* emuliert eine RS-232 konforme serielle Schnittstelle. Hierdurch wird ein drahtloses Kabel zwischen zwei Bluetooth-Geräten etabliert, sodass Anwendungen ohne Modifikationen weiter verwendet werden können. Das Serial Port Profile baut auf dem Generic Access Profile auf, wird selbst jedoch von vielen anderen Profilen verwendet, die eine serielle Schnittstelle zwischen zwei Geräten für ihre Arbeit voraussetzen. Hierzu gehören z.B. die Profile *Dial up Networking*, *FAX* und *Synchronisation*. Das Serial Port Profile definiert einen Dienst, der in der Lage ist mehrere serielle Verbindungen auf eine drahtlose Verbindung zu multiplexen.

## Der Standard IEEE 802.15 und Bluetooth

Nach der Veröffentlichung der Bluetooth-Spezifikation gründete das IEEE eine Arbeitsgruppe, IEEE 802.15, zur Definition eines Standards für *Wireless Personal Area Networks (WPAN)*. Der Standard gibt einen Abdeckungsbe- reich von 10 m in jede Richtung ausgehend von einer Person, die dem IEEE 802.15 konforme Geräte trägt. Unter IEEE 802.15 werden insgesamt vier Standards, die sehr unterschiedliche Bereiche abdecken, und zusammenfas- send in Abbildung 2.20 dargestellt sind, beschrieben.

Standard	Beschreibung
IEEE 802.15.1	Von Bluetooth abgeleiteter Standard für WPANs mit 1 Mbit/s.
IEEE 802.15.2	Empfehlungen für die Koexistenz in lizenzenfreien Funkfre- quenzen.
IEEE 802.15.3	WPAN-Standard mit Datenraten ab 20 Mbit/s für Multimedia-Anwendungen.
IEEE 802.15.4	Standard für Anwendungen, die nur eine kleine Datenrate von max. 200 kbit/s erfordern, z.B. Spielzeuge und Sen- soren.

Abbildung 2.20: Standards unter IEEE 802.15

Der Standard IEEE 802.15.1 baut auf der Bluetooth-Spezifikation auf. Jedoch übernimmt IEEE 802.15.1 nicht die vollständige Bluetooth-Spezifikation, sondern verwendet nur einen Teil daraus. Hierzu gehören alle Protokolle von L2CAP und darunter, d.h. im Sinne von IEEE sind das die Komponen- ten, die für die Bitübertragung und den Mediumzugriff verantwortlich sind.

Das Hauptziel der IEEE ist, einen Standard zu etablieren, der ähnlich zu IEEE 802.11x leicht in vorhandene lokale Netze integriert werden kann.

## 2.5 Fazit

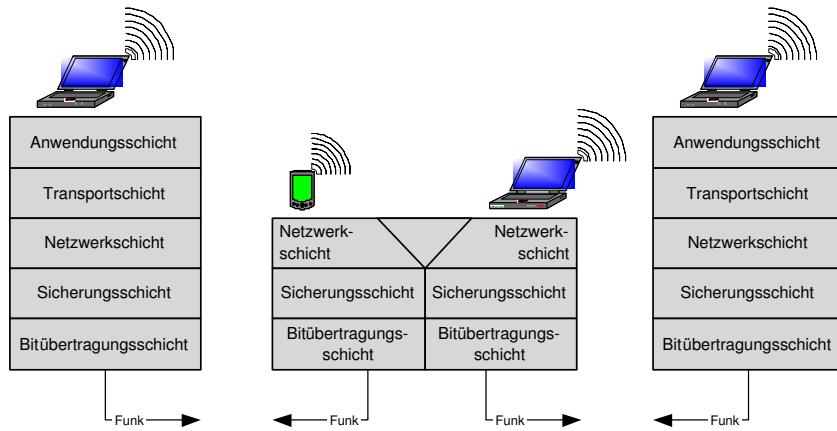


Abbildung 2.21: Referenzmodell für Ad-hoc-Netze, die in dieser Arbeit betrachtet werden.

In diesem Kapitel wurden bisher die Eigenschaften, die Klassifizierung, die Architektur und mögliche Technologien für mobile multi-hop Ad-hoc-Netze betrachtet und diskutiert. In diesem Abschnitt wird beschrieben, wie die in dieser Arbeit betrachteten Ad-hoc-Netze aussehen, und welche Eigenschaften sie besitzen. Dabei wird größtenteils auf die besprochenen Themen aus den vorherigen Abschnitten zugrückgegriffen.

Der Protokollstack der mobilen Knoten, die im Rahmen dieser Arbeit in Simulationen eingesetzt werden, ist in Abbildung 2.21 dargestellt. Ein Knoten besteht aus fünf Schichten, wobei die höheren drei Schichten, d.h. die Anwendungsschicht, die Transportschicht und die Internet-Schicht, dem TCP/IP-Modell entnommen sind. Die Host-an-Netzwerk-Schicht von TCP/IP ist in diesem Fall mit IEEE 802.11 mit 2 Mbit/s besetzt. Dieser wurde in Abschnitt 2.4.1 ausführlich beschrieben. Auf der Mediumzugriffsunterschicht ist CSMA/CA mit der Erweiterung RTS/CTS zu finden.

Im Rahmen dieser Arbeit werden flache homogene mobile multi-hop Ad-hoc-Netze betrachtet, d.h. es findet keine Unterscheidung zwischen den einzelnen Knoten statt. Alle Knoten haben die gleiche Ausstattung, sind mobil und die Kommunikationsendpunkte einer Verbindung sind mehrere Hops voneinander entfernt.

---

# KAPITEL 3

---

## Methodik

In diesem Kapitel wird die Methodik, die dieser Arbeit zu Grunde liegt und zur Untersuchung von mobilen multi-hop Ad-hoc-Netzen verwendet wurde, beschrieben. Zunächst wird in Abschnitt 3.1 das eingesetzte Simulationswerkzeug vorgestellt. In Abschnitt 3.2 werden einige Simulationsparameter mit ihrem Einfluss auf Simulationen diskutiert. Abschnitt 3.3 beschreibt die grundlegenden Einstellungen für die in Simulationen verwendeten Simulationsparameter.

### 3.1 Das Simulationstool

Für die Untersuchung von mobilen multi-hop Ad-hoc-Netzen wird hauptsächlich das Simulationswerkzeug ns-2 [FV00] eingesetzt. Das ns-2 ist ein diskret-ereignis-orientiertes Simulationswerkzeug, das vom VINT<sup>1</sup> Projekt der DARPA<sup>2</sup> in Zusammenarbeit mit der UC Berkeley<sup>3</sup>, LBL<sup>4</sup>, USC/ISI<sup>5</sup> und Xerox PARC seit 1995 entwickelt wird. Das Ziel bei der Entwicklung von ns-2 war eine offene und modulare Simulationsumgebung zu entwickeln, um heterogene Netze zu untersuchen. Der Quellcode des Simulators ist frei zugänglich, wodurch das Simulationswerkzeug von vielen Forschern auf der Welt eingesetzt und erweitert wird. Eine Erweiterung für Funknetze wurde vom CMU<sup>6</sup> MONARCH<sup>7</sup> Projekt entwickelt und in ns-2 integriert. Mit die-

---

<sup>1</sup> Virtual InterNetwork Testbed

<sup>2</sup> Defense Advanced Research Projects Agency

<sup>3</sup> University of California, Berkeley

<sup>4</sup> Lawrence Berkeley National Laboratory

<sup>5</sup> University of Southern California, Information Sciences Institute

<sup>6</sup> Carnegie Mellon University

<sup>7</sup> Mobile Networking Architectures

ser Erweiterung erlaubt ns-2 auch die Simulation von mobilen multi-hop Ad-hoc-Netzen.

Um den Untersuchungsprozess zu beschleunigen, haben sich die Entwickler von ns-2 für eine bilinguale Simulationstechnik entschieden. Der eigentliche Simulationskern ist in der Programmiersprache C++ implementiert, da hier die Optimierung des Zeit- und Speicherplatzbedarfs im Vordergrund steht. Die weniger kritischen Teile des Simulators sind in der Skriptsprache OTcl<sup>8</sup> implementiert. Diese Aufteilung erlaubt den schnellen und komfortablen Entwurf von Simulationsszenarien auf der einen und eine hohe Leistung auf der anderen Seite.

### 3.1.1 Der Protokollstack

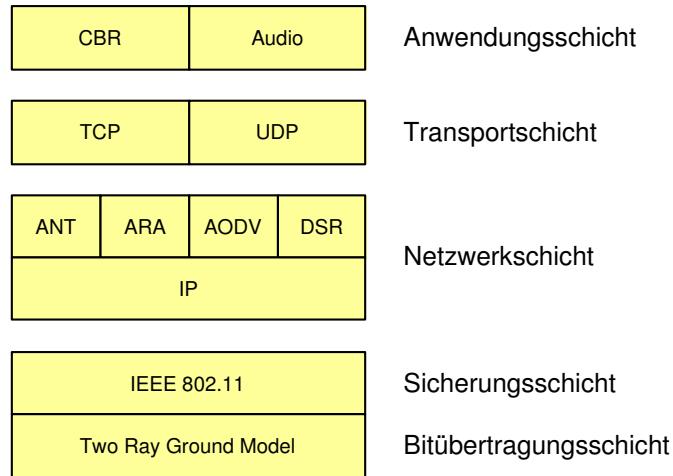


Abbildung 3.1: Protokollstapel eines Knotens im Simulationswerkzeug ns-2.

Im Simulationswerkzeug ns-2 bestehen Netzketten aus fünf Schichten (siehe Abbildung 3.1), die im Folgenden kurz beschrieben werden:

- **Anwendungsschicht**

Auf der Anwendungsschicht befinden sich die eigentlichen Anwendungen, die entweder Datenverkehr zum Versenden generieren oder ankommende Daten einsammeln. Im Simulationswerkzeug ns-2 gibt es Generatoren, die Datenverkehr für typische Anwendungen wie Telnet-, FTP- oder WWW-Verbindungen erzeugen. Es ist auch möglich, eigene

---

<sup>8</sup> OTcl ist eine objekt-orientierte Erweiterung von Tcl.

Generatoren zu implementieren. In dieser Arbeit werden Datenverkehr mit konstanter Bitrate, Audioverbindungen und FTP-Verbindungen benutzt.

- **Transportschicht**

Auf der Transportschicht stehen die typischen Internetprotokolle TCP und UDP zur Verfügung. Im Rahmen dieser Arbeit wird für die Übertragung von Datenverkehr mit konstanter Bitrate und Audioströmen UDP eingesetzt. Bei der Untersuchung der Routingalgorithmen werden auch Simulationen mit TCP gefahren.

- **Netzwerkschicht**

Das Simulationswerkzeug bietet auf der Netzwerkschicht IP an. Es wurden viele Routingalgorithmen entwickelt, die sich auf dieser Schicht befinden und auf IP aufsetzen. Die in Kapitel 5 vorgestellten Routingalgorithmen befinden sich auf der Netzwerkschicht.

Die Identifizierung von Netzknoten mit IP-Adressen findet ebenfalls auf dieser Schicht statt, weshalb die Verfahren aus Kapitel 4 auch hier zu finden sind.

- **Sicherungsschicht**

Die Sicherungsschicht eines Knotens in ns-2 besteht aus zwei Teilschichten. Auf der oberen Teilschicht befindet sich ein Medienzugriffsvorfahren. In den in dieser Arbeit durchgeführten Simulationen ist dies das CSMA/CA. Auf der unteren Teilschicht befindet sich ein Paketpuffer, der Pakete auch priorisiert behandeln kann. Dies ist beispielsweise bei der Untersuchung von Routingalgorithmen nützlich, wo Routingpakete eine höhere Priorität als Datenpakete besitzen. Zusätzlich bietet das Simulationswerkzeug ein Modul für das Address Resolution Protocol (ARP) an, welches für die Abbildung von IP-Adressen auf MAC-Adressen benutzt wird.

- **Bitübertragungsschicht**

Auf dieser Schicht wird das Übertragungsmedium und die Übertragung der einzelnen Bits modelliert. Das Übertragungsmedium ist bei der Funkkommunikation die Luftschnittstelle, die im Vergleich zu Leitungen eine höhere Bitfehlerrate besitzt. Die Modellierung des Übertragungsmediums spielt eine wichtige Rolle, da die Bitfehlerrate die Leistung der Protokolle auf den höheren Schichten beeinflussen kann.

## 3.2 Simulationsparameter und ihr Einfluss

Im Folgenden werden einige, den Simulationen zu Grunde liegende, Parameter im Detail mit ihrer Auswirkung auf die Ergebnisse diskutiert. Zu diesen Parametern gehört das Ausbreitungsmodell und das Mobilitätsmodell.

### 3.2.1 Das Ausbreitungsmodell

Das Ausbreitungsmodell beschreibt wie die übertragenen Bits in Form von Signalen von den Knoten wahrgenommen werden und ob sie korrekt empfangen werden können. Im Gegensatz zu leitungsgebundenen Netzen, bei denen die Signalausbreitung hauptsächlich durch die Leitung vorgegeben ist, wird die Signalausbreitung in Funknetzen von vielen Faktoren wie Abschattung, Reflexionen, Streuung, Beugung und die Mehrwegeausbreitung beeinflusst [Sch00].

#### Free Space Ausbreitungsmodell

Das einfachste Ausbreitungsmodell ist das so genannte *Free Space Model*, welches von einer ungestörten direkten Sichtlinie – Line of Sight (LOS) – zwischen Quell- und Zielknoten ausgeht (siehe Abbildung 3.2a)). Die Signalstärke nimmt bei diesem Modell quadratisch mit dem Abstand zwischen Quell- und Zielknoten ab [Sta02]. Dieses Modell eignet sich für kurze Distanzen zwischen Quell- und Zielknoten.

Die Empfangsleistung  $P_r(d)$  beim Empfänger kann in Abhängigkeit von der Entfernung  $d$  zum Sender nach folgender Gleichung berechnet werden [Rap96].

$$P_r(d) = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 d^2 L}$$

Dabei bezeichnet  $P_t$  die Sendeleistung,  $G_t$  und  $G_r$  den Antennengewinn vom Sender und Empfänger,  $\lambda$  die Wellenlänge und  $L$  den Systemverlust, der nicht durch die Signalausbreitung bedingt ist.

#### Two-Ray Ground Ausbreitungsmodell

Das Ausbreitungsmodell *Two-Ray Ground* berücksichtigt zwei Wege des Signals in der Berechnung. Zum einen die direkte Sichtlinie und zum anderen eine Reflexion von der Erdoberfläche (siehe Abbildung 3.2b)). Dieses Modell eignet sich für größere Distanzen zwischen Quell- und Zielknoten. Wenn die Distanz zwischen Quell- und Zielknoten einen Grenzwert unterschreitet,

wird wieder das Free-Space-Modell verwendet. Bei diesem Ausbreitungsmodell wird angenommen, dass die Erdoberfläche flach ist.

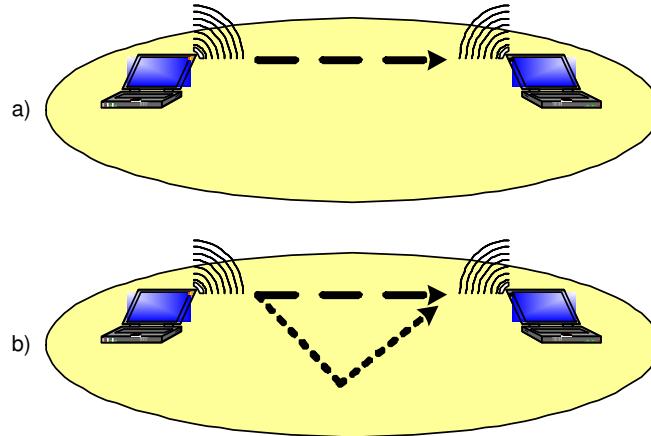


Abbildung 3.2: Die Signalausbreitungsmodelle: a) Free Space und b) Two-Ray Ground.

Die Empfangsleistung  $P_r(d)$  beim Empfänger kann in Abhängigkeit von der Entfernung  $d$  vom Sender nach folgender Gleichung bestimmt werden [Rap96].

$$P_r(d) = \frac{P_t \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2}{d^4}$$

Dabei bezeichnet  $P_t$  die Sendeleistung,  $G_t$  und  $G_r$  den Antennengewinn vom Sender und Empfänger und  $h_t$  bzw.  $h_r$  die Höhe der Antenne des Senders bzw. des Empfängers.

Von beiden Ausbreitungsmodellen werden keine Hindernisse berücksichtigt, die die Funksignale zusätzlich stören könnten. Im Rahmen dieser Arbeit wird in den Simulationen als Signalausbreitungsmodell immer das Two-Ray Ground verwendet.

### 3.2.2 Das Mobilitätsmodell

Das Mobilitätsmodell beschreibt wie sich die Knoten während der Simulation bewegen. Es gibt sehr unterschiedliche Mobilitätsmodelle [CBD02, HGPC99]. Bei der Untersuchung von mobilen multi-hop Ad-hoc-Netzen wird aber hauptsächlich das *Random Waypoint Mobility (RWM)* Modell eingesetzt. Im Folgenden wird dieses Mobilitätsmodell genauer analysiert und der Einfluss auf Simulationen diskutiert. Hierfür wird zunächst versucht, die

Knotenverteilung, die durch dieses Modell bedingt ist, analytisch herzuleiten. Das analytisch hergeleitete Ergebnis wird danach mit Simulationsergebnissen verglichen.

Beim RWM-Modell wählt ein Knoten zufällig eine Zielposition auf der Simulationsfläche aus und bewegt sich mit einer konstanten Geschwindigkeit  $v$  auf diesen Punkt zu. Nach der Ankunft an der Zielposition verbleibt der Knoten eine bestimmte Zeit  $\Delta t$  (Pausenzeit) und wählt sich danach eine neue Zielposition auf der Simulationsfläche aus. Die Wahl der Zielposition ist gleichverteilt auf der Simulationsfläche. Die Bewegungsgeschwindigkeit wird ebenfalls aus einem vorgegebenen Bereich gleichverteilt ausgewählt. Die Pausenzeit wird für einen gesamten Simulationslauf fest vorgegeben.

### RWM auf einer Strecke

Im Folgenden wird die Knotenverteilung des RWM-Modells analytisch hergeleitet. Der Einfachheitshalber beginnt die Betrachtung mit der eindimensionalen Version des Modells. Dabei wird eine konstante Geschwindigkeit und eine Pausenzeit von 0 Sekunden angenommen.

Betrachtet wird die Bewegung eines Knotens auf einer Strecke  $S_A = [0 \dots n]$ . Der Knoten wählt gleichverteilt einen Zielpunkt  $i$  auf der Strecke  $S_A$  aus. Die Wahrscheinlichkeit einen Punkt  $i$  gleichverteilt auf der Strecke  $S_A$  auszuwählen ist gegeben durch

$$p(i) = \frac{1}{n+1} \quad i = 0 \dots n.$$

Zunächst wird die Histogrammfunktion  $h(i)$  berechnet, welche die Anzahl der Besuche des Punktes  $i$  angibt. Ein Punkt gilt als besucht, wenn der Knoten auf seiner aktuellen Bewegung über den Punkt fährt. Ausgehend von der Histogrammfunktion wird dann die Verteilungsfunktion  $f(i)$  berechnet, die der Forderung

$$\sum_{i=0}^n f(i) = 1$$

genügen muss. Die Verteilungsfunktion  $f(i)$  ergibt sich aus der Normierung der Histogrammfunktion zu

$$f(i) = \frac{h(i)}{\sum_{j=0}^n h(j)} \quad i = 0 \dots n.$$

**Die Histogrammfunktion:** Die Histogrammfunktion  $h(i)$  gibt die Anzahl der Besuche an, die ein Punkt  $i$  vom Knoten erfährt. Da der Knoten sich

gleichmäßig mit einer konstanten Geschwindigkeit auf der Strecke  $S_A$  von  $j$  nach  $k$  mit  $j, k \in \{0, \dots, n\}$  bewegt, ist ein Besuch des Punktes  $i$  genau dann gegeben, wenn sich  $i$  zwischen  $j$  und  $k$  befindet. Hierzu wird die Besuchtfunktion  $v(i, j, k)$  wie folgt definiert:

$$v(i, j, k) = \begin{cases} 1 & , j \leq i \leq k \\ 0 & , \text{sonst} \end{cases}$$

Mit Hilfe der Besuchtfunktion ist die Histogrammfunktion gegeben als

$$h(i) = \sum_{j=0}^n \sum_{k=0}^n v(i, j, k).$$

Dabei kennzeichnet  $j$  alle möglichen Start- und  $k$  Endpunkte auf  $S_A$ , d.h. alle möglichen Teilstrecken auf  $S_A$ , die der Knoten fahren kann, werden betrachtet und für den Punkt  $i$  die Besuche aufsummiert.

Zunächst wird der Startpunkt  $s$  festgehalten. Die zweite Summe des Ausdrucks berechnet sich dann zu:

$$\begin{aligned} h_s(i) &= \sum_{k=0}^n v(i, s, k) \\ &= \begin{cases} i+1 & 0 \leq i \leq s \\ (n-i+1) & s < i \leq n \end{cases} \end{aligned}$$

Wenn der Punkt  $i$  zwischen 0 und dem Startpunkt  $s$  liegt, wird er vom Knoten, solange sich dieser zwischen 0 und dem Punkt  $i$  befindet, besucht. Liegt der Punkt  $i$  zwischen dem Startpunkt  $s$  und  $n$ , wird er solange besucht wie der Knoten zwischen dem Punkt  $i$  und dem Ende der Strecke ist (siehe Abbildung 3.3).

Daraus ergibt sich für die weitere Berechnung der Histogrammfunktion

$$\begin{aligned} h(i) &= \sum_{j=0}^n h_j(i) \\ &= \sum_{j=0}^{i-1} (n-i+1) + \sum_{j=i}^n (i+1) \\ &= i \cdot (n-i+1) + (n-i+1) \cdot (i+1) \\ &= (n-i+1) \cdot (2i+1) \end{aligned}$$

Aus der Histogrammfunktion erhält man die Verteilungsfunktion:

$$\begin{aligned} f(i) &= \frac{(n-i+1) \cdot (2i+1)}{\sum_{j=0}^n h(j)} \\ &= \frac{(n-i+1) \cdot (2i+1)}{1 + \frac{13}{6}n + \frac{3}{2}n^2 + \frac{1}{3}n^3} \end{aligned} \tag{3.1}$$

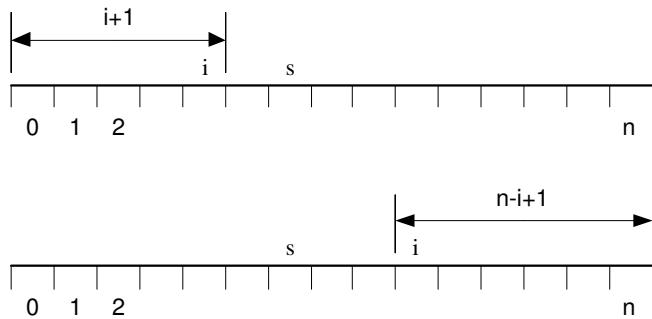


Abbildung 3.3: Bewegung eines Knotens gemäß der eindimensionalen RWM bei der ein Punkt vom Knoten besucht wird.

Die Verteilungsfunktion erfüllt die Forderung:

$$\sum_{i=0}^n f(i) = 1$$

### Interpretation der Verteilungsfunktion

In Abbildung 3.4 ist die Verteilungsfunktion der eindimensionalen Version des RWM-Modells dargestellt. Die Kurve ist in beide Richtungen symmetrisch, hat ihren Hochpunkt in der Mitte der Strecke und geht an beiden Rändern gegen Null. Die Wahrscheinlichkeit einen Knoten in der Mitte der Simulationsstrecke anzutreffen ist am höchsten und nimmt zu den Rändern hin rapide ab.

### RWM auf einer Ebene

Die analytische Herleitung der Knotenverteilung stellt sich für den zweidimensionalen Fall als erheblich schwieriger dar. Deshalb wird hier eine Abschätzung durch die eindimensionale Verteilungsfunktion vorgestellt und mit Simulationsergebnissen verglichen.

Die Verteilungsfunktion aus Gleichung 3.1 wird für die Abschätzung der Knotenverteilung auf einer quadratischen Ebene mit Kantenlänge  $n$  benutzt. Die resultierende Verteilung ist gegeben durch

$$\begin{aligned} f(x, y) &= f(x) \cdot f(y) \\ &= \frac{(-n+x)(2x+1)(n-y)(2y+1)}{n^2(n+1)^2(2n+1)^2} - 36 \end{aligned} \quad (3.2)$$

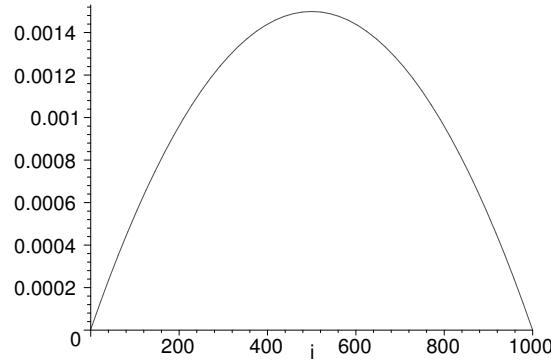


Abbildung 3.4: Verteilungsfunktion des eindimensionalen Random Waypoint Mobilitätsmodells auf einer Strecke der Länge 1000 m.

Die Funktion aus Gleichung 3.2 erfüllt die Forderung

$$\sum_{x,y=0}^n f(x,y) = 1$$

und kann daher als eine Verteilungsfunktion aufgefasst werden. In Abbildung 3.5 ist die Verteilungsfunktion grafisch dargestellt. Ähnlich zu der Verteilung im eindimensionalen Fall ist die Kurve in alle Richtungen symmetrisch und besitzt einen Hochpunkt in der Mitte der Fläche.

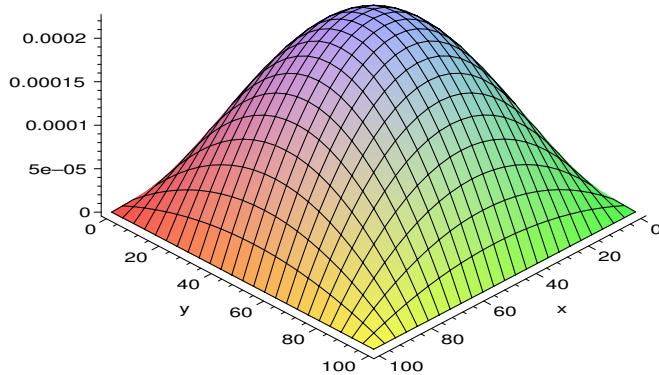


Abbildung 3.5: Verteilungsfunktion des 2D-Random-Waypoint Mobilitätsmodells.

Im analytischen Modell wurden zwei wichtige Parameter des RWM-Modells

vernachlässigt. Dies sind die Pausenzeit, die ein Knoten zwischen zwei Bewegungen einlegt, und die Geschwindigkeit, mit der sich ein Knoten bewegt. Ergebnisse aus Simulationen haben gezeigt, dass der Einfluss der Pausenzeit eine enorme Rolle spielt.

Abbildung 3.6 zeigt sechs Graphen, die aus Simulationen stammen. Die Ergebnisse aus den Simulationen berücksichtigen im Gegensatz zum analytischen Modell auch die Pausenzeit der Knoten. Die Graphen zeigen die Knotenverteilung mit Pausenzeiten von 0, 30, 60, 120, 300 und 600 Sekunden. Ähnlich zum analytischen Ergebnis aus Abbildung 3.5 zeigen alle Graphen hier auch eine Symmetrie in alle Richtungen. Beim Vergleich des Graphen mit Pausenzeit von 0 Sekunden ist zu erkennen, dass der Graph insgesamt steiler ist als der Graph aus Abbildung 3.5. Aus den Graphen lässt sich schließen, dass mit zunehmender Pausenzeit die Knotenverteilung gleichmäßiger wird. Der Grund dafür ist, dass durch die Pausenzeit eine gewisse Basis gelegt wird mit der sich die Knoten an einem Punkt aufhalten. Es ist hier anzumerken, dass der Graph in Abbildung 3.6(f) immer noch einen Hochpunkt in der Mitte der Ebene besitzt. Der Unterschied ist nun nicht mehr so groß wie bei dem Graphen mit Pausenzeit von 0 Sekunden.

### 3.3 Simulationsumgebung

Es gibt eine Reihe von Parametern, die in den Simulationen benutzt und variiert wurden, um unterschiedliche Situationen zu modellieren. Dieser Abschnitt beschreibt die in dieser Arbeit eingesetzten Werte und Einstellungen.

#### 3.3.1 Grundeinstellungen

Die Grundeinstellungen sind in allen Simulationen gleich. Dies betrifft die Größe der Simulationsfläche, die Anzahl und Art der Knoten und die Simulationszeit. Auf Abweichungen von diesen Standardwerten wird an entsprechender Stelle hingewiesen.

- Die Simulationsfläche hat eine Größe von  $1500 \text{ m} \times 300 \text{ m}$ .
- Auf der Simulationsfläche befinden sich 50 mobile Knoten mit einer IEEE 802.11 2 Mbit/s Funkschnittstelle und einer Reichweite von 250 m (siehe Abschnitt 2.4.1).
- Die Ausbreitung der Funksignale wird durch das Two-Ray Ground Modell beschrieben.

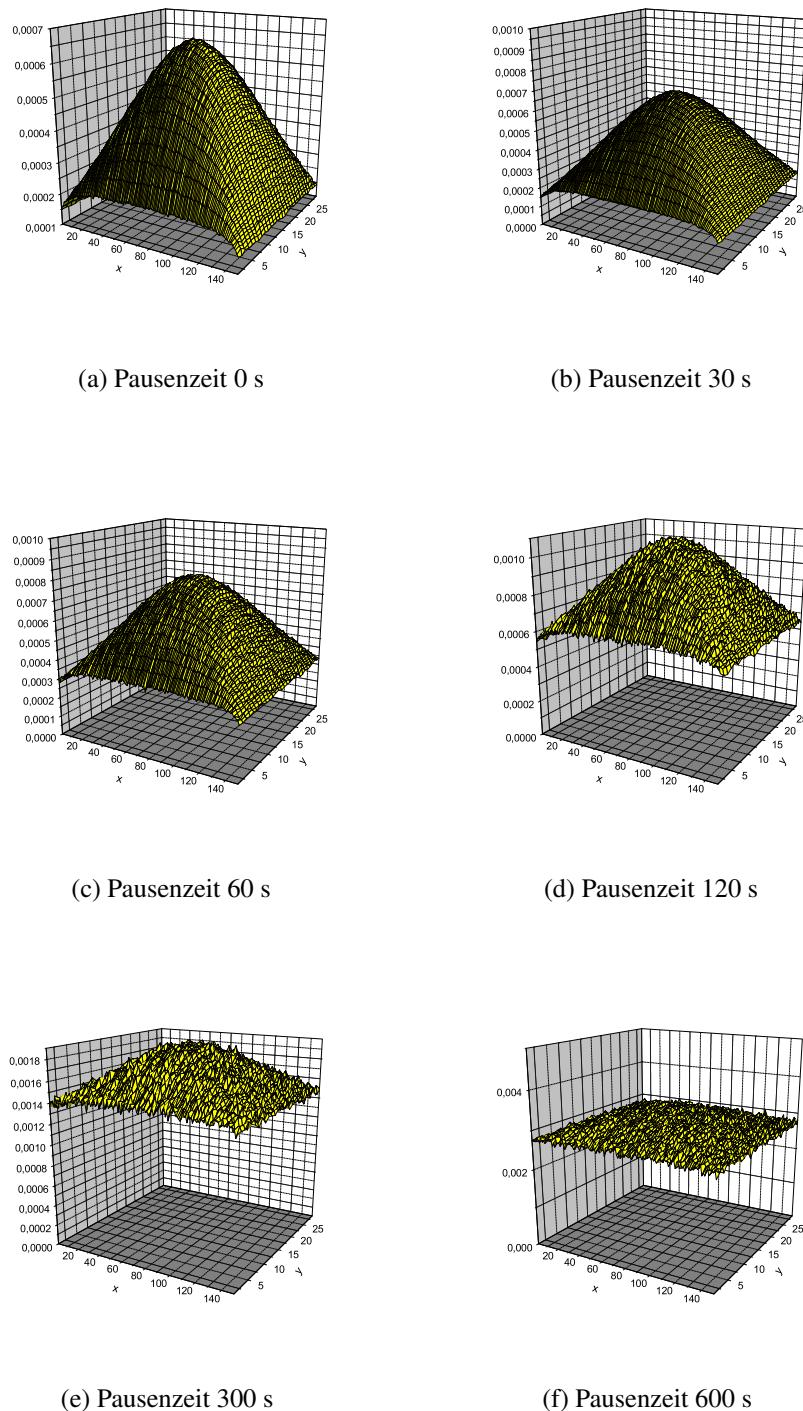


Abbildung 3.6: Knotenverteilung beim Mobilitätsmodell RWM mit unterschiedlichen Pausenzeiten.

- Die Knoten haben auf der Verbindungsschicht einen Puffer in der Pakete zur Abarbeitung abgelegt werden. Sobald der Puffer voll ist werden ankommende Pakete verworfen. Dadurch hat die Größe des Puffers einen Einfluss auf Simulationsergebnisse, da z.B. während einer Pfadsuche der Puffer schnell voll laufen kann. Die Kapazität eines Puffers auf der Verbindungsschicht eines Knotens beträgt 50 Pakete.
- Die Simulationszeit beträgt 900 Sekunden.
- Für jeden Messpunkt wurden 10 unabhängige Simulationsdurchläufe durchgeführt. Die dargestellten Ergebnisse wie Durchschnitt und Konfidenzintervall wurden auf der Basis dieser 10 Simulationsergebnisse berechnet.

### 3.3.2 Bewegungsszenarien

Eine besondere Eigenschaft von mobilen multi-hop Ad-hoc-Netzen ist, dass durch die Knotenmobilität die Netztopologie Veränderungen unterworfen ist. Umso höher die Knotenmobilität ist, desto öfter verändert sich die Netztopologie (siehe Abbildung 3.7). Die Mobilität der Knoten wird in dieser Arbeit nach dem Random-Waypoint Mobility Modell beschrieben, das zwei Parameter besitzt:

Pausenzeit	Verb.-abbrüche		Pfadwechsel		Ziel unerreichbar	
	Mittel	Konfidenz	Mittel	Konfidenz	Mittel	Konfidenz
0	7082	118	39573	1509	82	42
30	5567	140	36011	776	55	52
60	4809	78	31101	1278	65	49
120	3723	139	27792	877	9	12
300	2087	75	15551	693	0	0
600	1117	58	7705	397	0	0
900	0	0	0	0	0	0

Abbildung 3.7: Mittlere Anzahl der Verbindungsabbrüche und Pfadwechsel und ihre  $\alpha = 0,05$ -Konfidenzintervalle bei einer maximalen Bewegungsgeschwindigkeit von 10 m/s.

#### • Bewegungsgeschwindigkeit

Die Bewegungsgeschwindigkeit eines Knotens wird gleichverteilt aus dem Intervall  $(0, v_{max}]$  zufällig gewählt. In dieser Arbeit werden unterschiedliche maximale Bewegungsgeschwindigkeiten benutzt, wobei  $v_{max} = 1 \text{ m/s}$  für Szenarien mit wenig Mobilität,  $v_{max} = 5 \text{ m/s}$  für mittlere Mobilität und  $v_{max} = 10 \text{ m/s}$  für hohe Mobilität eingesetzt wird.

- **Pausenzeit**

Die Pausenzeit legt fest wie lange ein Knoten nach der Ankunft auf der Zielposition verharrt, bevor er eine neue Zielposition auswählt. In dieser Arbeit wird die Pausenzeit durch sieben unterschiedliche Zeiten ausgedrückt, diese sind 0, 30, 60, 120, 300, 600 und 900 Sekunden.

In den Szenarien mit 900 Sekunden Pausenzeit ist das Netz starr, da sich die Knoten nicht bewegen, dabei spielt die Bewegungsgeschwindigkeit keine Rolle. Bei einer Pausenzeit von 0 Sekunden befinden sich die Knoten immer in Bewegung.

Szenarien mit einer maximalen Knotengeschwindigkeit von 1 m/s beschreiben eine große Menschenmenge, z.B. in einer Einkaufsstraße, oder in einem großen Einkaufszentrum, die sich zwar langsam, jedoch stetig, bewegt. Ein Beispiel für ein Szenario mit maximaler Knotengeschwindigkeit von 10 m/s ist durch eine Menge von Robotern, die gemeinsam arbeiten, gegeben. Beispielsweise könnte eine Menge von Roboter-Gabelstaplern eine Lagerhalle verwalten. Neue Kisten müssen in der Lagerhalle deponiert und bestellte Waren ausgeliefert werden. Die Gabelstapler bewegen sich zwischen dem aktuellen Platz einer Kiste und dem zukünftigen Abstellplatz schnell. Während der Gabelstapler die Kiste an der richtigen Position abstellt, steht er still. Dies wird durch die Pausenzeit ausgedrückt.

### 3.3.3 Kommunikationsmuster

Das Kommunikationsmuster gibt an was für ein Typ von Anwendung die Daten erzeugt.

#### Konstante Bitrate

Bei einem Datenverkehr mit konstanter Bitrate (Constant Bit Rate, CBR) erzeugt der Quellknoten in gleichen Zeitabständen Pakete mit einer festgelegten Größe und versendet sie an den Zielknoten. Die unterschiedlichen Größen, die bei diesem Kommunikationsmuster variiert werden, sind:

- **Anzahl paralleler Verbindungen**

Es werden gleichzeitig 1, 3, 5, 10 oder 15 Verbindungen aufgebaut, dabei übernimmt kein Knoten im Netz eine doppelte Rolle. Ein Knoten ist entweder ein Quellknoten, ein Zielknoten oder ein Knoten, der nur Pakete weiterleitet. Bei insgesamt 50 Knoten und einer Verbindungsanzahl von 5, gibt es 5 Quell- und 5 Zielknoten. Die restlichen 40 Knoten im Netz fungieren als Router.

- **Anzahl der Datenpakete pro Sekunde**

Jeder Quellknoten sendet 4 Pakete pro Sekunde.

- **Paketgröße**

Die Paketgröße auf der Anwendungsschicht beträgt standardmäßig 512 Byte. Es werden jedoch auch Simulationen mit 64 Byte durchgeführt. Bei Abweichung vom Standardwert wird dies an der jeweiligen Stelle vermerkt.

## TCP-Verbindungen

Die Leistung der Routingalgorithmen wird zusätzlich hinsichtlich der Übertragung von Daten, mit dem am meisten verwendeten Transportprotokoll TCP, bewertet. Ziel hierbei ist es zu untersuchen, wie die Leistung von TCP durch die einzelnen Routingalgorithmen beeinflusst wird. Die Parameter in diesen Simulationen sind:

- **Anwendung**

Auf der Anwendungsschicht wird FTP für die Generierung von Daten eingesetzt.

- **Anzahl paralleler Verbindungen**

Es werden gleichzeitig 5 Verbindungen aufgebaut.

- **TCP-Variante**

Auf der Transportschicht wird TCP/SACK (Selective Acknowledgement) eingesetzt.

- **Paketgröße**

Die Paketgröße auf der Anwendungsschicht beträgt 1460 Byte.

- **Pausenzeit**

Die Knoten legen zwischen den Bewegungen eine Pause von 1 Sekunde ein.

## Echtzeitdaten

Die Routingalgorithmen werden hinsichtlich ihrer Leistung bei der Übertragung von zeitkritischen Daten bewertet. Hierzu wurden typische Audioverbindungen gewählt. Mit einer Datenrate von 13 kBit/s wird die Sprachübertragung in GSM-Netzen simuliert. Die veränderlichen Größen bei diesen Simulationen sind:

- **Anzahl paralleler Audioverbindungen**

Es werden gleichzeitig 1, 3, 5, 10, 15, 20 und 25 Duplex-Verbindungen aufgebaut, d.h. jeder Quellknoten ist gleichzeitig auch ein Zielknoten. Jedoch nimmt ein Knoten nur an einer Duplex-Verbindung teil. Hierdurch wird die symmetrische Kommunikation von Telefongesprächen simuliert. Bei 5 Duplex-Verbindungen gibt es daher 10 Verbindungen.

- **Anzahl der Datenpakete pro Sekunde**

Jeder Quellknoten sendet 26 Pakete pro Sekunde.

- **Paketgröße**

Die Paketgröße auf der Anwendungsschicht beträgt 64 Byte.

- **Pausenzeit**

Die Knoten legen zwischen den Bewegungen keine Pause ein.



---

## KAPITEL 4

---

# Adressierung in Ad-hoc-Netzen

Unter einem Ad-hoc-Netz wird nicht nur ein Netzwerk verstanden, das sich spontan und ohne das Vorhandensein von Infrastruktur aufbauen lässt, sondern es wird auch angenommen, dass das Netz sich selbstständig konfiguriert ohne dass der Benutzer aktiv eingreifen muss. Die Konfiguration bezieht sich primär auf die Versorgung der Knoten mit IP-Adressen, sodass die vom Internet bekannten Anwendungen und Protokolle benutzt werden können.

Ein Hauptproblem in mobilen multi-hop Ad-hoc-Netzen ist die effiziente Ermittlung von Pfaden zwischen Kommunikationspartnern, was durch die Knotenmobilität erschwert wird. Daher befassen sich die meisten Forschungsarbeiten zu Ad-hoc-Netzen mit dieser Fragestellung [Joh94, RT99, HJ00, Toh02], siehe auch Kapitel 5. Bevor jedoch ein Pfad zwischen zwei Knoten in einem Ad-hoc-Netz ermittelt werden kann, müssen die Knoten identifiziert werden. Die Identifizierung ist im Allgemeinen durch die Adressierung der Knoten sichergestellt. Zur Adressierung der Knoten eines Netzwerkes werden zwei Komponenten benötigt: i.) einheitliche Adressen und ii.) ein Verfahren, welches den Knoten die Adressen eindeutig zuweist.

In diesem Kapitel wird die automatische Adresskonfiguration von Knoten in einem Ad-hoc-Netz auf der Basis von IP-Adressen betrachtet. Dieses Kapitel ist im Weiteren wie folgt aufgebaut. In Abschnitt 4.1 werden die Grundlagen zur Adresskonfiguration vorgestellt. In Abschnitt 4.2 wird die Adressierung in lokalen Netzen (LAN) beschrieben und Argumente aufgeführt, warum sie für mobile multi-hop Ad-hoc-Netze nicht geeignet ist. In Abschnitt 4.3 werden Ansätze zur Adressierung von Ad-hoc-Netzen aus der Literatur diskutiert. In Abschnitt 4.4 wird die Agentenbasierte Adressierung vorgestellt. Die Simulationsparameter werden in Abschnitt 4.5 beschrieben. Die Ergebnisse der Untersuchungen und ein Vergleich der unterschiedlichen Verfahren werden in Abschnitt 4.6 diskutiert. Das Kapitel schließt in Abschnitt 4.7 mit einer Diskussion.

## 4.1 Grundlagen zur Adresskonfiguration

Im Internet und in lokalen Netzen werden IP-Adressen verwendet, die von einem Administrator den jeweiligen Knoten zugewiesen werden. Dies geschieht hauptsächlich während der Installation und Konfiguration. Ein Knoten bekommt nur dann eine neue IP-Adresse, wenn seine Position im Netzwerk geändert wird. Dies liegt daran, dass die IP-Adressen auch Information über das Netzwerk enthalten.

### Was benötigt ein IP-Host?

Neben der eigentlichen IP-Adresse benötigen die Knoten weitere Informationen, die für eine erfolgreiche Anbindung an das Netz erforderlich sind. Zu den benötigten Informationen gehören [Gut01b]:

- **IP-Adresse**

Ein Knoten benötigt für jede Netzwerkschnittstelle (Network Interface Card, NIC) eine IP-Adresse. IPv4-Adressen sind 32 Bit lang und bestehen aus einem Netz- und einem Hostanteil. Im Gegensatz dazu bestehen IPv6-Adressen aus 128 Bit.

- **Subnet-Maske**

Mit der Subnet-Maske kann ein Knoten die Netzidentifikation aus der IP-Adresse extrahieren.

- **Adresse des Standardrouters**

Wenn Knoten aus unterschiedlichen Netzen miteinander kommunizieren wollen, müssen die Nachrichten an einen Router geschickt werden, der sie weiterleitet. Der Router, den ein Knoten normalerweise benutzt, ist der Standardrouter.

- **Domänenname**

Knoten im Internet werden nicht nur durch ihre IP-Adresse angesprochen. Es werden auch Namen verwendet, die für Menschen einfacher zu merken sind. Namen sind hierarchisch angeordnet und bestehen aus zwei Teilen: Hostname und Domänenname. Den Teil des Namens ohne den Hostnamen nennt man Domänenname.

- **Domänennamen-Server**

Um beliebige IP-Adressen in Namen bzw. Namen in IP-Adressen umzuwandeln, existiert eine verteilte Datenbank, das so genannte *Domain*

*Name System (DNS)*. Zugriff auf diese verteilte Datenbank erhalten Knoten über so genannte *Domain Name Server*.

## Arten der Konfiguration

Ein Knoten kann die benötigten Informationen auf unterschiedlichen Wegen erhalten, wodurch eine Klassifikation der Konfigurationsarten möglich ist.

- **Statische Konfiguration**

Ein Administrator besucht jeden Knoten im Netzwerk und führt die Konfiguration manuell durch. Der Aufwand steigt mit wachsender Anzahl der Knoten im Netzwerk rapide an. Für die Durchführung der Konfiguration wird ein Netzwerkspezialist benötigt.

- **Dynamische Konfiguration**

Die Knoten können die benötigten Informationen von einem bestimmten Server beziehen. Die dynamische Konfiguration erfordert einen speziellen Konfigurationsserver. Die Administration wird nahezu auf die Verwaltung des Konfigurationsservers reduziert.

- **Automatische Konfiguration**

In diesem Fall können die Knoten im Netz, ohne den Eingriff einer externen Stelle oder eines Benutzers, die Konfiguration selbstständig durchführen.

## Anforderungen an Adressierungsverfahren

Ein Adressierungsverfahren für mobile multi-hop Ad-hoc-Netze muss anderen Anforderungen genügen als ein Verfahren für Festnetze. Gerade diese Anforderungen werden von existierenden Adressierungsverfahren nicht vollständig oder ungenügend erfüllt.

- **Multi-hop-Konfiguration**

In mobilen multi-hop Ad-hoc-Netzen können nicht alle Knoten direkt miteinander kommunizieren. Vielmehr müssen die Nachrichten über Zwischenknoten weitergeleitet werden. Deshalb ist es erforderlich, dass ein Adressierungsverfahren diesen Aspekt berücksichtigt.

- **Eindeutigkeit der vergebenen Adressen**

Die Adressen in einem Netz müssen eindeutig sein, d.h. nur einmal vorkommen. Ansonsten kann keine zuverlässige Kommunikation zwischen den Knoten aufgebaut werden. Gerade die Mobilität von Ad-hoc-Netzen und die Ungewissheit über die Netzteilnehmer kann zu Adresskonflikten im Netz führen.

- **Verwaltung der Adressen**

Die Adressressourcen sind auch in mobilen Ad-hoc-Netzen nicht unendlich. Deswegen ist es erforderlich, dass Knoten Adressen freigeben, wenn sie sie nicht mehr benötigen oder aus dem Netz scheiden. In mobilen Ad-hoc-Netzen kann es auch vorkommen, dass Knoten aus dem Netz ausscheiden ohne sich abzumelden. Aus diesem Grund ist es die Aufgabe des Adressierungsverfahrens, nicht mehr benutzte Adressen zur Wiederverwendung zu kennzeichnen.

- **Behandlung der Aufteilung und Vereinigung von Netzen**

Auf einem Areal können sich gleichzeitig mehrere Ad-hoc-Netze befinden, die sich auch teilweise überschneiden können. Deshalb ist es erforderlich, dass ein Adressierungsverfahren die Knoten mehrerer Ad-hoc-Netze auch mit unterschiedlichen Adressen versorgt, sodass die Knoten unterscheiden können, ob sie im gleichen Netz sind.

Die Existenz mehrerer Ad-hoc-Netze kann unterschiedliche Gründe haben. Einige der Ad-hoc-Netze können sich vereinigen, um ein größeres Ad-hoc-Netz zu bilden. Deshalb muss das Adressierungsverfahren die Readressierung eines Teiles oder des gesamten Ad-hoc-Netzes durchführen können.

Ein Ad-hoc-Netz kann sich auch aufteilen, sodass aus einem Ad-hoc-Netz mehrere neue Ad-hoc-Netze entstehen. Die Knoten in einem neu entstandenen Ad-hoc-Netz sollten als ein unterschiedliches Netz behandelt werden. Dies erfordert die Readressierung der Knoten in den so entstandenen Ad-hoc-Netzen.

- **Sicherheit**

Die Sicherheit spielt auf jeder Ebene der Kommunikation in Ad-hoc-Netzen eine wichtige Rolle, da es sehr schwierig ist, zwischen „Angreifern“ und „normalen“ Knoten zu unterscheiden. Im Rahmen der Adresskonfiguration wird dieser Aspekt kaum in Betracht gezogen. Wie sich später zeigt, sind alle Ansätze für die Adressierung offen für Angriffe. Dabei kann ein Knoten sehr einfach den Dienst blockieren, in dem er entweder alle Adressen besetzt, oder Adressen als belegt kennzeichnet. Wünschenswert wäre eine Authentifizierung vor der Zuweisung einer Adresse.

## 4.2 Adressierung in lokalen Netzen

Das Ziel der Adressierung ist die Zuweisung einer eindeutigen und gleichartigen Bezeichnung der Knoten in einem Netz, um diese für Kommunikationsverbindungen wiederzufinden. Im Internet werden zur Adressierung IP-Adressen verwendet, die hierarchisch aufgebaut sind, und sowohl einen Knoten als auch das Netz, an dem der Knoten angeschlossen ist, eindeutig identifizieren. Die IP-Adresse dient dabei als eine logische Adresse auf der Netzwerkschicht und wird zur Pfadfindung benutzt. Für die eigentliche Zustellung eines Paketes an einen konkreten Host in einem lokalen Netz wird die MAC-Adresse der Sicherungsschicht benötigt. Die Abbildung der IP-Adressen auf die MAC-Adressen und umgekehrt wird durch die beiden Protokolle *Address Resolution Protocol* (ARP) [Plu82] und *Reverse Address Resolution Protocol* (RARP) [FMMT84] durchgeführt.

Die Zuweisung der IP-Adressen an Knoten erfolgt meistens manuell durch einen Administrator. Ein Knoten behält die zugewiesene Adresse typischerweise, solange er nicht an ein anderes Netz angeschlossen wird. Bei großen Netzen erfordert diese Vorgehensweise einen hohen Verwaltungsaufwand.

### 4.2.1 Dynamic Host Configuration Protocol

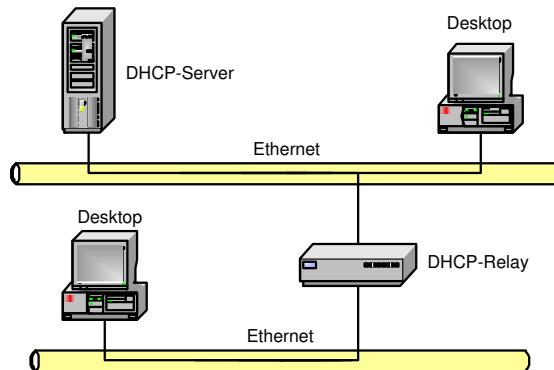


Abbildung 4.1: Die Konfiguration von Netzwerkknoten mit DHCP.

Das *Dynamic Host Configuration Protocol (DHCP)* [Dro97] ist eine Erweiterung des BootP [CG85] Protokolls und dient der dynamischen Verwaltung von IP-Adressen. Mit DHCP können Knoten während des Bootvorgangs oder der Netzanmeldung benötigte Netzwerkinformationen von einem *DHCP-Server* anfordern. Dazu gehört eine gültige IP-Adresse, die Subnetzmaske, Adres-

sen der primären und sekundären Nameserver, und weitere Informationen, wie beispielsweise eine bestimmte Bootdatei.

DHCP besteht aus zwei Komponenten. Die erste Komponente ist ein Protokoll, welches zur Übermittlung von Netzwerkinformationen von einem DHCP-Server an einen DHCP-Client dient. Die zweite Komponente ist ein Verfahren für die Zuweisung von Netzwerkadressen an Hosts. DHCP kennt drei Arten der Zuweisung von Netzwerkadressen, die parallel eingesetzt werden können und unterschiedliche Administrationszwecke unterstützen.

- **Automatic allocation:** Ein Host bekommt eine permanente Netzwerkadresse zugewiesen.
- **Dynamic allocation:** Ein Host bekommt für eine beschränkte Zeit eine Netzwerkadresse zugewiesen.
- **Manual allocation:** Die Netzwerkadresse, die ein Host bekommt, bestimmt der Netzwerkadministrator. DHCP wird in diesem Modus nur zur Übertragung der Informationen an den Client benutzt.

Von diesen drei Arten der Adresszuweisung ist nur die dynamische Adresszuweisung in der Lage, nicht mehr benutzte Adressen wieder zu verwenden. Deshalb eignet sie sich für den Einsatz in Umgebungen in denen Hosts nur temporär eine Netzwerkadresse benötigen. Die manuelle Adressierung erlaubt die zentrale Konfiguration von Hosts. Dabei spezifiziert der Netzwerkadministrator die Zuweisung und muss nicht jeden Host selbst aufsuchen. Sie eignet sich vor allem für Hosts, die bestimmte Dienste anbieten. Der Unterschied zwischen automatic und manual allocation ist, dass bei automatic allocation die vergebene Netzwerkadresse aus dem für den DHCP-Server zur Verfügung stehenden Pool gewählt wird, und bei manual allocation der Netzwerkadministrator die Netzwerkadresse wählt.

Bei der dynamischen Adresszuweisung wird die IP-Adresse eines Knotens, der sich vom Netz abmeldet, als frei markiert und kann vom DHCP-Server an andere Knoten vergeben werden. Sollte sich der Knoten später wieder am Netz anmelden, kann er eine andere IP-Adresse zugewiesen bekommen, wobei DHCP bemüht ist, Knoten die gleiche Adresse zuzuweisen. Für die Versorgung der Knoten steht dem DHCP-Server ein Pool von IP-Adressen zur Verfügung. Wenn dieser alle IP-Adressen aus seinem Vorrat vergeben hat, kann er keine weiteren Knoten mehr bedienen.

Die von einem DHCP-Server zugewiesenen IP-Adressen sind nur eine bestimmte Zeit lang gültig. Nach Ablauf dieser Frist müssen die Knoten die Lebensdauer ihrer Adressen verlängern.

Ein DHCP-Server ist für die Versorgung eines Subnetzes konzipiert. Müssen mehrere Subnetze durch einen DHCP-Server bedient werden, können

spezielle Hilfsknoten eingesetzt werden, die zwischen den Knoten und dem DHCP-Server vermitteln. Die Hilfsknoten werden als *DHCP-Relays* bezeichnet (siehe Abbildung 4.1).

### 4.2.2 Zero Configuration Networking

Die *Zero Configuration Networking Work Group* der IETF arbeitet an einer Protokollfamilie, die eine automatische Konfiguration von Netzwerken erlaubt. Die Arbeitsgruppe definiert unterschiedliche Bereiche.

- (i) IP-Adresskonfiguration [CA01],
- (ii) Abbildung von Namen auf IP-Adressen [CTAG01],
- (iii) Service-Discovery [Gut01c] und
- (iv) IP-Multicastkonfiguration [Gut01a].

Im Arbeitsbereich (i) wird die automatische Generierung einer IP-Adresse aus dem Netz 169.254/16, welches für die Nutzung auf einer lokalen Verbindungsstrecke reserviert ist, beschrieben. Das Verfahren ist für die Konfiguration von Knoten, die am gleichen Leitungsabschnitt angeschlossen sind, entworfen, d.h. die zu konfigurierenden Knoten müssen alle Nachrichten untereinander empfangen können.

Die Zuweisung an einen Knoten erfolgt nach dem folgenden Verfahren: Ein neuer Knoten wählt sich aus dem reservierten Adressbereich zufällig eine IP-Adresse aus und überprüft, ob diese Adresse schon im Netz benutzt wird. Für die Überprüfung wird ARP verwendet. Wenn die gewählte Adresse besetzt ist, wiederholt der Knoten das Verfahren bis er eine freie IP-Adresse gefunden hat.

### 4.2.3 Mobile-IP

Mobile-IP [Per98] wurde als Erweiterung zu IP vorgeschlagen, um mobilen Knoten temporären Zugriff auf fremde Netze zu gewähren. Der mobile Knoten ist zwischen den temporären Netzanbindungen nicht an das Netzwerk angebunden, d.h. wenn der mobile Knoten sich an einem Gastnetz angemeldet hat, bleibt er auch bis zum Ende der Sitzung angemeldet, und verändert seine Adresse nicht.

Das Konzept von Mobile-IP baut auf zwei definierten Knoten, dem *Home-Agent* und dem *Foreign-Agent*, auf (siehe Abbildung 4.2). Der mobile Knoten besitzt eine feste IP-Adresse, die so genannte *home address*, die er verwendet,

wenn er in seinem Heimatnetz ist. Beim Anmelden an einem Gastnetz bekommt der mobile Knoten eine temporäre Adresse, die so genannte Care-of-Address, vom Foreign-Agent des Gastnetzes zugewiesen, welche dem Home-Agent des mobilen Knotens bekannt gegeben wird. Mobile-IP kennt zwei Arten, um einem mobilen Knoten eine Care-of-Address zu zuweisen: i.) Foreign agent care-of address: Hierbei benutzt der mobile Knoten eine IP-Adresse des Foreign-Agents, ii.) co-located care-of-address: in diesem Fall bekommt der mobile Knoten eine temporäre IP-Adresse, die er im Gastnetzwerk benutzen kann. Im zweiten Fall wird der Einsatz von DHCP für die Versorgung von mobilen Knoten mit IP-Adressen vorgesehen. Mit dem Wissen über die Care-of-Address ist der Home-Agent in der Lage, die Pakete für den mobilen Knoten an seine aktuelle Position weiterzuleiten.

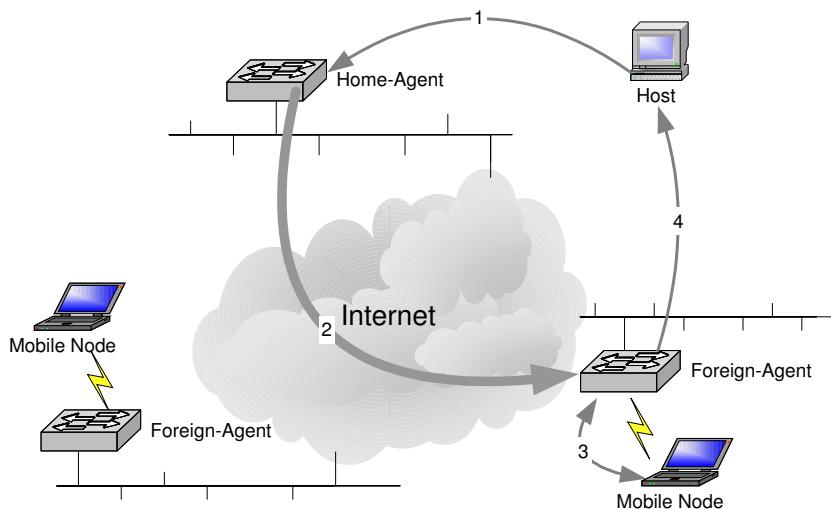


Abbildung 4.2: Prinzip von Mobile-IP.

### 4.3 Ansätze zur Adressierung von Ad-hoc-Netzen

Die Verfahren aus dem vorherigen Abschnitt wurden hauptsächlich mit dem Ziel entworfen, den Konfigurationsaufwand von Netzen zu verringern. Die Dynamik von mobilen multi-hop Ad-hoc-Netzen wird daher durch diese Verfahren nicht ausreichend widergespiegelt.

Bei DHCP wird ein ständig erreichbarer DHCP-Server vorausgesetzt, der in mobilen multi-hop Ad-hoc-Netzen nicht gewährleistet werden kann. Durch die Arbeiten der Zero Configuration Networking Arbeitsgruppe soll die Kon-

figuration von Knoten ohne die Existenz eines DHCP-Servers möglich sein; dies ist jedoch auf die Konfiguration von Knoten in einem Leitungsabschnitt beschränkt. Mobile-IP wurde für mobile Knoten entworfen, die temporär an ein Netz angeschlossen werden, jedoch zwischen zwei Netzanbindungen ausgeschaltet sind. Knoten in einem Ad-hoc-Netz erfordern jedoch Verfahren, die einen dynamischen Wechsel der Konfiguration erlauben.

In diesem Abschnitt werden Ansätze aus der Literatur beschrieben, die für die dynamische Konfiguration von Ad-hoc-Netzen entworfen wurden.

### 4.3.1 Autokonfiguration für Ad-hoc-Netze

In [PMW<sup>+</sup>01] und [PKP01] wird ein Verfahren für die Adresskonfiguration von Ad-hoc-Netzen vorgeschlagen, welches eine Erweiterung der Methode der Zero Configuration Networking Arbeitsgruppe ist. Im Gegensatz zum Verfahren aus [CA01], welches die Konfiguration von Knoten an einem Link reguliert, wird hier die multi-hop Architektur von Ad-hoc-Netzen mit berücksichtigt, d.h. die Kontrollnachrichten können über andere Knoten weitergeleitet werden. Die beiden Dokumente [PMW<sup>+</sup>01] und [PKP01] unterscheiden sich in Details, das grundlegende Verfahren ist bei beiden jedoch gleich. Deshalb wird im Folgenden eine leicht abstrahierte Version beschrieben.

Ein neuer Knoten wählt sich zufällig eine Adresse aus dem reservierten Netz 169.254/16 bei IPv4 oder eine Adresse mit dem Präfix MANET\_PREFIX bei IPv6. Danach sendet der Knoten eine Broadcastnachricht, um zu prüfen, ob die gewählte Adresse schon im Netz benutzt wird. Der Knoten wartet ADDRESS\_DISCOVERY Zeiteinheiten auf eine Antwort. Wenn sich innerhalb dieser Zeit kein Knoten meldet, geht der Knoten davon aus, dass die gewählte Adresse zur Zeit nicht im Netz verwendet wird und benutzt diese Adresse. Wenn die gewählte Adresse im Netz schon verwendet wird, erhält der Knoten eine entsprechende Nachricht von dem Knoten, der die Adresse besitzt. In diesem Fall wiederholt der Knoten den Vorgang bis zum Erfolg. Ein neuer Knoten verwendet eine temporäre Adresse aus einem speziell reservierten Adressbereich während er eine gültige Adresse sucht. Es wird angenommen, dass die Zeit, um eine freie Adresse zu finden, sehr kurz ist, und daher keine Adresskollision entsteht.

Ein Nachteil dieses Ansatzes ist, dass keine netzweite Eindeutigkeit der verwendeten Adressen garantiert ist. Insbesondere wenn sich ein mobiles multi-hop Ad-hoc-Netz aufteilt und später wieder vereint, kann es passieren, dass mehrere Knoten die gleiche Adresse verwenden. Für die Überprüfung der Adressen nach der Vereinigung des aufgeteilten Netzes ist kein Mechanismus vorgesehen.

### 4.3.2 MANETconf

In [NP02] wird ein Verfahren für die Konfiguration von Knoten in einem mobilen Ad-hoc-Netz vorgestellt, welches die multi-hop Eigenschaft von Ad-hoc-Netzen berücksichtigt. Die Autoren führen die Adressierung von Knoten in einem Ad-hoc-Netz auf das Problem der verteilten Einigung auf eine Größe zurück.

Ein neuer Knoten, der eine Adresse benötigt, spricht einen im Ad-hoc-Netz befindlichen Knoten und benutzt ihn als Stellvertreter. Der Stellvertreter wählt eine zufällige Adresse aus und informiert alle Knoten im Netz mit einer Broadcastnachricht über die gewählte Adresse. Daraufhin müssen alle Knoten im Netz die neue Adresse bestätigen. Hierzu senden alle Knoten im Netz entsprechende positive oder negative Bestätigungsnotizen an den Stellvertreter. Die ausgewählte Adresse wird dem neuen Knoten nur zugewiesen, wenn alle Knoten im Ad-hoc-Netz die Adresse positiv bestätigt haben, ansonsten wählt der Stellvertreter eine neue zufällige Adresse aus und wiederholt den Vorgang.

Um unterschiedliche Ad-hoc-Netze zu unterscheiden, besitzt jedes Ad-hoc-Netz eine so genannte Partitionsidentifikation, welche durch den Knoten mit der kleinsten IP-Adresse im Ad-hoc-Netz erstellt und bekannt gegeben wird. Die Knoten eines bestimmten Ad-hoc-Netzes sind durch die Partitionsidentifikation gekennzeichnet. Für die Bekanntgabe der Partitionsidentifikation schlagen die Autoren zwei Verfahren vor. Das erste Verfahren ist verteilt. Jeder Knoten berechnet die Partitionsidentifikation aus den Informationen seiner Routingtabelle. Hier besteht jedoch die Gefahr, dass Inkonsistenzen in den Routingtabellen zu Fehlern in der Berechnung führen können. Das zweite Verfahren basiert auf der regelmäßigen Bekanntgabe der Partitionsidentifikation. Hierzu sendet der Knoten mit der kleinsten IP-Adresse im Netz regelmäßig eine Broadcastnachricht mit der Partitionsidentifikation aus. Auf der Basis der Partitionsidentifikation ist das Verfahren in der Lage die Aufteilung und Vereinigung von Ad-hoc-Netzen zu erkennen.

Die Schwachstellen des Ansatzes liegen bei der konsistenten Vergabe von eindeutigen Adressen in einem Netz. Es ist durchaus möglich, dass zwei unterschiedliche Knoten die gleiche IP-Adresse für zwei neue Knoten wählen. Obwohl die Autoren die Vermeidung von Broadcastnachrichten ansprechen, basieren die robusten Teile des Verfahrens auf dem Fluten von Informationen im Netz und im Falle der Erkennung der Aufteilung und Vereinigung von Netzen sogar auf regelmäßigem fluten des Netzes.

### 4.3.3 Buddy basierte Adressierung

In [MP02] wird ein Verfahren auf der Basis von Binären-Buddy-Systemen, die aus der Speicherverwaltung bekannt sind, vorgestellt. Dabei muss die Anzahl der zur Verfügung stehenden Adressen einer 2er Potenz entsprechen.

Der erste Knoten im Ad-hoc-Netz besitzt eine bestimmte Menge von Adressen und wählt sich die erste Adresse aus. Wenn ein neuer Knoten um eine Adresse anfragt, teilt der Knoten die ihm zur Verfügung stehende Anzahl an Adressen in zwei Bereiche. Den ersten Teil der Adressen behält der Knoten selbst und den zweiten Teil bekommt der neue Knoten. In Abbildung 4.3 ist an einem Beispiel die verteilte Generierung von Adressen dargestellt. Am Anfang existiert nur der Knoten A, dem 16 Adressen zur Verfügung stehen. Knoten A hat die Adresse 0. Wenn nun ein zweiter Knoten hinzukommt, Knoten B, teilt A den ihm zur Verfügung stehenden Adressbereich in zwei Teile und gibt dem neuen Knoten den zweiten Teil des Adressbereichs. Knoten B nimmt die Adresse 8. Dem Knoten A steht nun der Adressbereich [1, 7] und dem Knoten B der Adressbereich [9, 15] zur Verfügung, um neue Knoten mit Adressen zu versorgen. Durch diese Vorgehensweise werden Konflikte bei der Vergabe von neuen Adressen von vornherein vermieden. Das Verfahren erlaubt auch die konfliktfreie parallele Bedienung mehrerer neuer Knoten von unterschiedlichen bereits existierenden Knoten aus.

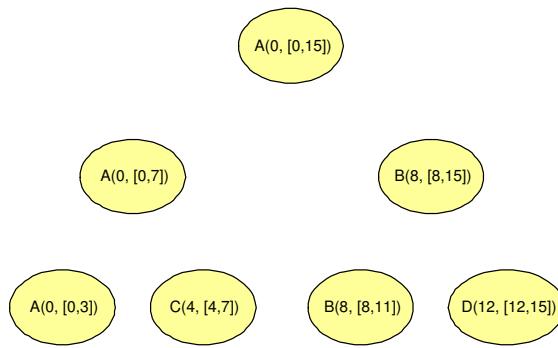


Abbildung 4.3: Ablauf der Adressgenerierung bei der Buddy basierten Adressierung.

Zur Erkennung der Aufteilung und Vereinigung von Ad-hoc-Netzen wird auf das Verfahren aus [NP02] zurückgegriffen, das in Abschnitt 4.3.2 vorgestellt wurde. Die Schwächen des Verfahrens liegen in der schlechten Ausnutzung der zur Verfügung stehenden Adressen. Wenn neue Knoten verstärkt an einem bestimmten Knoten anfragen, kann dies schnell dazu führen, dass der Knoten keine freien Adressen mehr hat. Zusätzlich besitzt das Verfahren die Schwä-

chen bei der Erkennung der Aufteilung und Vereinigung von Ad-hoc-Netzen, die schon oben diskutiert wurden.

#### 4.3.4 Prophet Address Allocation

In [ZNM03] wird das *Prophet Address Allocation* vorgestellt, das ähnlich zum Verfahren von [MP02] arbeitet. Im Gegensatz zum Buddy-System wird hier eine Funktion eingesetzt, um Adressen für neue Knoten zu berechnen. Die Funktion ist ähnlich zu einem Pseudozufallszahlengenerator und erzeugt eine Sequenz von Nummern. Die Autoren fordern von der Funktion, dass der Abstand des Erscheinen einer Nummer sehr groß ist und möglichst jede Nummer nur einmal vorkommen soll, d.h. im Sinne von Pseudozufallszahlengeneratoren soll die Funktion eine sehr große Periodenlänge besitzen. Durch die Wahl eines entsprechend großen Wertebereichs wird diesen Anforderungen Rechnung getragen.

Der erste Knoten im Ad-hoc-Netz wählt sich zufällig eine Adresse und einen Initialwert für die Funktion. Dadurch ist die Sequenz aller Nummern (=Adressen) im Ad-hoc-Netz bekannt. Hierauf basiert die Bezeichnung des Verfahrens, der erste Knoten im Netz kennt die zukünftigen Adressen im Voraus und hat die Verantwortung eine günstige Konstellation für die Funktion auszuwählen, sodass die Konfliktwahrscheinlichkeit von Adressen, d.h. die mehrmalige Generierung einer Nummer, klein ist. Hierzu probiert der erste Knoten mehrere Sequenzen aus und überprüft diese auf Wiederholungen.

Zur Erkennung der Aufteilung und Vereinigung von Ad-hoc-Netzen wird wiederum auf das Verfahren aus [NP02] zurückgegriffen, das hier ein wenig modifiziert ist.

#### 4.3.5 Klassifikation der Verfahren

Die vorgestellten Verfahren aus der Literatur zur Adresskonfiguration von Ad-hoc-Netzen lassen sich nach der Art der Generierung einer Adresse für einen neuen Knoten wie folgt klassifizieren:

- **Reaktive Verfahren**

Bei dieser Klasse müssen sich die Knoten im Ad-hoc-Netz auf die an einen neuen Knoten zu vergebende Adresse einigen. Entweder wählt sich der neue Knoten selbst oder aber ein bereits im Ad-hoc-Netz befindlicher Knoten die neue Adresse aus. Daraufhin werden alle Knoten im Netz um Erlaubnis befragt, d.h. das Ad-hoc-Netz reagiert auf die Anfrage eines neuen Knotens.

Zu dieser Klasse gehören die Ansätze aus den Abschnitten 4.2.2, 4.3.1 und 4.3.2.

- **Proaktive Verfahren**

Jeder Knoten im Ad-hoc-Netz kann selbstständig und unabhängig von den anderen Knoten im Ad-hoc-Netz Adressen an neue Knoten vergeben. Bei diesen Verfahren agiert jeder Knoten stellvertretend für das gesamte Ad-hoc-Netz.

Die Verfahren aus Abschnitt 4.3.3 und 4.3.4 zählen zu dieser Klasse.

## 4.4 Die Agentenbasierte Adressierung

In diesem Abschnitt wird die *Agentenbasierte Adressierung* von mobilen multi-hop Ad-hoc-Netzen vorgestellt [GR02b, GR02a]. Das Ziel beim Entwurf war es, ein Verfahren für die Adresskonfiguration zu entwickeln, welches die Knoten in mobilen multi-hop Ad-hoc-Netzen schnell und zuverlässig adressiert. Um dieses Ziel zu erreichen, wurden die folgenden Anforderungen aufgestellt:

- **Eindeutigkeit der Adressen**

Die Knoten in einem mobilen multi-hop Ad-hoc-Netz müssen eindeutige IP-Adressen bekommen.

- **Einfacher Dienst**

Der benötigte Dienst soll von jedem Knoten in einem mobilen multi-hop Ad-hoc-Netz übernommen werden können. Ein Knoten, der den Adressierungsdienst anbietet, wird als *Adressierungsagent* (AA) bezeichnet.

- **Multi-hop-Konfiguration**

Die Adressierung von Knoten, die den Adressierungsagenten nicht direkt erreichen können, muss gewährleistet sein.

- **Robustheit**

Das Verfahren soll robust gegen die Aufteilung eines Ad-hoc-Netzes und die Vereinigung von mehreren Ad-hoc-Netzen sein.

- **Adaptivität**

Das Verfahren soll sich an Veränderungen in der Netztopologie anpassen. Bei Bedarf soll die Rekonfiguration des gesamten Netzes oder eines Teiles durchgeführt werden. Hierbei existieren zwei Teilanforderungen, die erfüllt werden müssen:

- (i) Wenn kein Adressierungsagent in einem Ad-hoc-Netz existiert, muss einer der Knoten die Aufgaben des Adressierungsagenten übernehmen.
- (ii) Wenn mehrere Adressierungsagenten sich in einem Ad-hoc-Netz befinden, sollen sich alle bis auf einen abschalten.

Die Wahl des Adressierungsagenten soll verteilt und unabhängig von der Anzahl der Knoten im Netz sein. Die Anforderung (i) stellt sicher, dass ein Ad-hoc-Netz sich selbst konfiguriert. Dies ist der Fall, wenn ein Ad-hoc-Netz neu aufgebaut wird oder durch die Aufteilung aus einem existierenden Ad-hoc-Netz hervorgeht. Die Selbstkonfiguration wird aber auch bei einem Ausfallen der Adressierungsagenten erforderlich. Durch Anforderung (ii) wird die Eindeutigkeit der Adressen in einem Ad-hoc-Netz, das sich durch die Vereinigung von mehreren Netzen gebildet hat, sichergestellt und die benötigte Last reduziert.

#### 4.4.1 Zustandsgraph eines Knotens

Bei der Agentenbasierten Adressierung befindet sich ein Knoten in einem von drei möglichen Zuständen (siehe Abbildung 4.4):

- **Unbound:** Der Knoten besitzt keine gültige IP-Adresse.
- **Bound:** Der Knoten besitzt eine gültige IP-Adresse.
- **Address-Agent:** Der Knoten besitzt eine gültige IP-Adresse und bietet den Adressierungsdienst an.

Ein Knoten befindet sich nach dem Start im Zustand *Unbound*, d.h., er besitzt keine gültige IP-Adresse. Von diesem Zustand ausgehend kann der Knoten sich in zwei andere Zustände bewegen. Wenn der ADDRESS\_AGENT\_DISCOVERY Timer abläuft, wechselt der Knoten in den Zustand *Adressierungsagent* (AA), wodurch er selbst zum Adressierungsagenten wird und andere Knoten bedienen kann. Bekommt der Knoten nach dem Start und vor dem Ablauf des ADDRESS\_AGENT\_DISCOVERY Timers eine *Verify*-Nachricht (VP) von einem Adressierungsagenten, antwortet der Knoten mit einer *Address-Request*-Nachricht (AR). Dadurch geht der Knoten in den Zustand *Bound*, d.h. der Knoten hat eine gültige IP-Adresse. Der Knoten bleibt im Zustand *Bound*, solange er vom Adressierungsagenten *Verify*-Nachrichten empfängt und auf diese mit *Address-Confirm*-Nachrichten (AC) antwortet. Den Zustand *Bound* verlässt der Knoten, wenn er VERIFY\_RECEIVE\_TIMER Zeiteinheiten keine *Verify*-Nachricht vom Adressierungsagenten erhält oder auf eine *Verify*-Nachricht nicht antwortet. Den Adressierungsagentenzustand verlässt ein

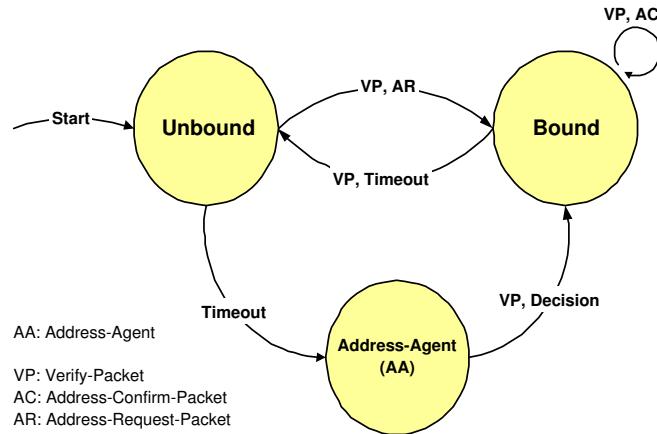


Abbildung 4.4: Der Zustandsgraph eines Knotens bei der Agentenbasierten Adressierung.

Knoten nur, wenn sich ein anderer Adressierungsagent im gleichen Ad-hoc-Netz befindet, und der Knoten die Wahl zum Adressierungsagenten verliert.

#### 4.4.2 Der Adressierungsagent

Die Basiskomponente der *Agentenbasierten Adressierung* bildet der *Adressierungsagent*, der für die Adressierung eines Ad-hoc-Netzes verantwortlich ist. Die Funktionalität des Adressierungsagenten kann von jedem Knoten übernommen werden. Zur eindeutigen Adressierung der Knoten verwaltet der Adressierungsagent eine *Address-List* (AL) der Knoten, die sich im Ad-hoc-Netz befinden. Die Address-List enthält die Zuordnungen von IP-Adressen zu MAC-Adressen.

Der Adressierungsagent sendet regelmäßig *Verify*-Nachrichten aus, welche die Address-List und einen Zeitstempel enthalten (siehe Abbildung 4.5(a)). Jeder Knoten, der eine Verify-Nachricht erhält, überprüft, ob seine Adresse in der Address-List enthalten ist und schickt eine *Address-Confirm*-Nachricht an den Adressierungsagenten, falls er weiterhin im Ad-hoc-Netz bleiben möchte (siehe Abbildung 4.5(b)). Ein neuer Knoten, der sich in der Address-List der Verify-Nachricht nicht wiederfindet, meldet sich beim Adressierungsagenten mit einer *Address-Request*-Nachricht an (siehe Abbildung 4.5(b)). Erhält der Adressierungsagent nach Ablauf des ADDRESS\_CONFIRM\_TIMERS keine Nachricht von einem Knoten, der in der Address-List enthalten ist, wird seine IP-Adresse markiert.

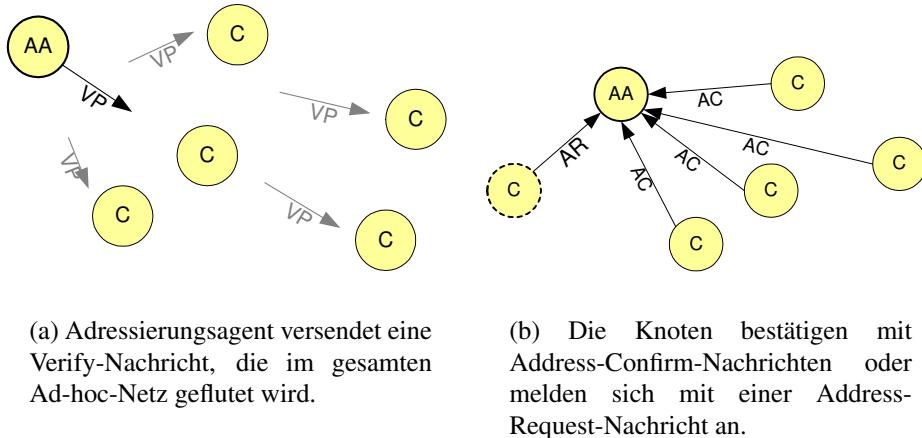


Abbildung 4.5: Konfiguration mit der Agentenbasierten Adressierung.

### Auswahl des Adressierungsagenten

Ein neuer Knoten wartet ADDRESS\_AGENT\_DISCOVERY Zeiteinheiten auf eine Verify-Nachricht. Wenn er innerhalb dieser Zeit keine Verify-Nachricht von einem Adressierungsagenten empfängt, geht er davon aus, dass kein Adressierungsagent vorhanden ist und wechselt selbst in den Adressierungsagentenzustand und verschickt eine Verify-Nachricht.

Wenn sich in einem Ad-hoc-Netz mehrere Adressierungsagenten befinden, schalten sich alle bis auf einen Adressierungsagenten ab, sodass die eindeutige Verwaltung der Knoten gewährleistet bleibt und der benötigte Overhead reduziert wird. Die Reduktion der Adressierungsagenten auf eins erfolgt nach dem folgenden Verfahren:

Wenn ein Adressierungsagent  $AA_k$  eine Verify-Nachricht eines anderen Adressierungsagenten  $AA_l$  empfängt, verfährt er wie folgt:  $AA_k$  berechnet die Anzahl der Knoten in seiner eigenen Address-List und die Anzahl der Knoten in der empfangenen Verify-Nachricht.  $AA_k$  verlässt den Adressierungsagentenzustand, wenn die Anzahl der ihm bekannten Knoten kleiner ist als die von  $AA_l$ . Wenn die Anzahl der bekannten Knoten von  $AA_l$  und  $AA_k$  gleich ist, entscheidet die MAC-Adresse der beiden Agenten. Der Adressierungsagent mit der kleineren MAC-Adresse verbleibt im Adressierungsagentenzustand.

### Ausfall des Adressierungsagenten

Knoten können in einem mobilen multi-hop Ad-hoc-Netz ohne Ankündigung aus dem Netz ausfallen. Was passiert, wenn der Adressierungsagent ausfällt?

Ein Knoten, der eine gültige Adresse besitzt, befindet sich im Zustand Bound

und verbleibt in diesem Zustand solange er Verify-Nachrichten vom Adressierungsagenten empfängt und diese bestätigt. Wenn der Adressierungsagent ausfällt, ist er nicht mehr in der Lage Verify-Nachrichten zu versenden. Durch das Ausbleiben der Verify-Nachrichten nimmt der Knoten an, dass der Adressierungsagent ausgefallen ist und wechselt in den Zustand Unbound.

Der Ausfall des Adressierungsagenten führt dazu, dass alle Knoten in den Zustand Unbound wechseln. Um zu verhindern, dass alle Knoten gleichzeitig in den Adressierungsagenten Zustand wechseln und das Netz mit Verify-Nachrichten überflutet, wählen alle Knoten zufällige Wartezeiten zwischen  $[0, VP\_SEND\_INTERVAL]$ . Nach Ablauf der Wartezeit und bei keinem Empfang einer Verify-Nachricht wechselt der Knoten in den Zustand des Adressierungsagenten und startet mit dem Versenden von Verify-Nachrichten.

### Paketverlust

Was passiert, wenn ein Knoten die eine Adresse haben möchte, versucht den Adressierungsagenten zu erreichen, jedoch erfolglos bleibt oder der Knoten keine Antwort vom Adressierungsagenten bekommt?

Ein neuer Knoten, der eine Verify-Nachricht von einem Adressierungsagenten empfängt, fordert mit einer Address-Request-Nachricht eine Adresse an. Sollte er vor Ablauf des ADDRESS\_REQUEST\_TIMER Timers keine Nachricht vom Adressierungsagenten erhalten, wiederholt er seine Anfrage. Danach nimmt der Knoten an, dass der Adressierungsagent nicht erreichbar ist und wechselt selbst in den Zustand Adressierungsagent.

#### 4.4.3 Robustheit des Verfahrens

##### Aufteilung eines Netzes

Wenn sich ein Ad-hoc-Netz in mehrere Netze aufteilt, verbleibt der Adressierungsagent in einem der Netze. In den übrigen Netzen müssen neue Adressierungsagenten gewählt werden. Die Situation ist mit dem Ausfall des Adressierungsagenten vergleichbar.

##### Vereinigung von Netzen

Empfängt der Adressierungsagent eines Ad-hoc-Netzes die Verify-Nachrichten von anderen Adressierungsagenten, kann dies zur Vereinigung der Netze zu einem größeren Netz führen. Hierzu müssen die Adressierungsagenten der beteiligten Netze sich auf den zukünftigen Adressierungsagenten einigen. Dies

geschieht mit dem Verfahren aus Abschnitt 4.4.2. Wenn die Netze eine unterschiedliche Anzahl an Knoten besitzen, verbleibt der Adressierungsagent aus dem größten Netz als Adressierungsagent. Ansonsten verbleibt der Adressierungsagent mit der kleinsten MAC-Adresse. Danach müssen alle Knoten aus den anderen Netzen rekonfiguriert werden.

Der Adressierungsagent kennt die MAC-Adressen aller Knoten in den beteiligten Netzen, da sie in den Verify-Nachrichten enthalten sind. Er generiert für alle neuen Knoten eine Adresse und versendet eine Verify-Nachricht mit gesetztem Merged Flag und fügt die subnetID der beteiligten Netze ein. Sobald die Knoten die Verify-Nachricht erhalten, können sie ihre neuen Adressen übernehmen. Damit ist die Rekonfiguration des vereinigten Netzes abgeschlossen.

#### 4.4.4 Generierung von Adressen

Im Rahmen der Agentenbasierten Adressierung wird die Benutzung von IPv6-Site-Local-Adressen vorgeschlagen, die folgendes Format [HD98] haben.

Bit	10	38	16	64
	1111111011	0	subnetID	interfaceID

Hier sind nur die subnetID und interfaceID wichtig. Die interfaceID ist weltweit eindeutig und von der MAC-Adresse abgeleitet und in [Oci01] beschrieben. Die subnetID ist spezifisch für ein Ad-hoc-Netz und wird vom Adressierungsagenten aus seiner eigenen MAC-Adresse berechnet. Der Adressierungsagent berechnet die subnetID, wenn er in den Adressierungsagentenzustand wechselt.

Wenn ein Knoten eine IP-Adresse anfordert, generiert der Adressierungsagent eine neue IP-Adresse, die von den MAC-Adressen des Adressierungsagenten und von der Adresse des anfragenden Knotens abgeleitet ist (siehe Abbildung 4.6). Dabei wird die interfaceID aus der vollständigen MAC-Adresse des anfragenden Knotens erzeugt.

#### 4.4.5 Einfluss von Timern und Parametern

Die Leistung der Agentenbasierten Adressierung wird durch die im Weiteren aufgeführten Parameter beeinflusst:

- VP\_SEND\_INTERVAL: Sendeintervall von Verify-Nachrichten

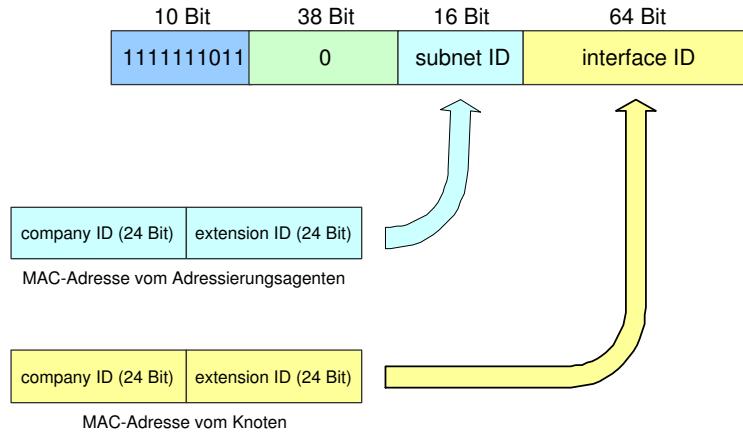


Abbildung 4.6: Konstruktion einer IPv6-Site-Local-Adresse aus den MAC-Adressen vom Adressierungsagenten und anfragenden Knoten.

Wenn das Sendeintervall von Verify-Nachrichten zu klein ist, wird unnötiger Overhead erzeugt. Ist jedoch das Sendeintervall zu groß gewählt, muss gegebenenfalls ein neuer Knoten zu lange warten, bis er eine Adresse erhält und mit den anderen Knoten kommunizieren kann. Der Standardwert für diese Größe ist 1 Sekunde.

- ADDRESS\_AGENT\_DISCOVERY: Wartezeit eines neuen Knotens auf eine Verify-Nachricht

Dieser Timer gibt an, wie lange ein neuer Knoten wartet, bis er in den Adressierungsagenten-Zustand übergeht und selbst Adressierungsagent wird. Der Wert dieses Timers hängt sehr stark vom Sendeintervall der Verify-Nachrichten ab. Der Standardwert für diesen Timer entspricht  $2 \times VP\_SEND\_INTERVAL$ .

- ADDRESS\_REQUEST\_TIMER: Wartezeit eines neuen Knotens auf die folgende Verify-Nachricht nach dem Versenden einer Address-Request-Nachricht.

Ein neuer Knoten setzt diesen Timer auf  $3 \times VP\_SEND\_INTERVAL$ . Nach Ablauf des Timers wiederholt er seine Anfrage. Bei erneutem Misserfolg geht er in den Adressierungsagentenzustand.

- ADDRESS\_CONFIRM\_TIMER: Innerhalb dieses Zeitfensters müssen sich die Knoten in einem Ad-hoc-Netz nach dem Empfang einer Verify-Nachricht bei dem Adressierungsagenten mit einer Address-Confirm-Nachricht melden. Der Standardwert für diesen Timer entspricht  $2 \times VP\_SEND\_INTERVAL$ .

- VERIFY\_RECEIVE\_TIMER: Ein Knoten mit gültiger Adresse, der sich im Zustand Bound befindet, wartet maximal diese Zeit ab, bevor er in den Zustand Unbound übergeht. Der Standardwert für diesen Timer entspricht  $3 \times \text{VP\_SEND\_INTERVAL}$ .

## 4.5 Adressierungsszenarien

Die Simulationsumgebung entspricht in den allgemeinen Parametern der Beschreibung aus Kapitel 3. In diesem Abschnitt werden die speziellen Einstellungen und Parameter beschrieben, die in den Simulationen für dieses Kapitel benutzt wurden.

Das Hauptinteresse bei den Untersuchungen gilt der Zeit, die benötigt wird, um ein bestimmtes Ad-hoc-Netz *vollständig zu adressieren*. Diese Zeit ist wichtig, da die Kommunikation im Netz erst nach der Konfiguration der Knoten beginnen kann.

Um das Verhalten der Verfahren in verschiedenen Situationen zu untersuchen, wurden mehrere Szenarien ausgewählt, in denen eine partielle oder vollständige Neuadressierung eines Ad-hoc-Netzes durchgeführt werden muss. Im folgenden Abschnitt wird auf diese Szenarien näher eingegangen.

### 4.5.1 Vollständige Adressierung eines Netzes

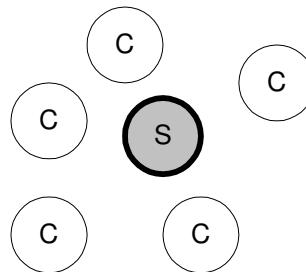


Abbildung 4.7: Ad-hoc-Netz mit einer Ebene, wobei alle Knoten direkt miteinander kommunizieren können.

In diesem Szenario existiert eine Menge von Knoten in der Simulationsumgebung, die keine gültigen IP-Adressen besitzen. Ziel ist die Zuweisung von eindeutigen Adressen an alle Knoten. Zum Zeitpunkt  $t = 0$  wird der Adress-Server, sollte dieser vorhanden sein, aktiv. Ansonsten ist es die Aufgabe des

Adressierungsverfahren einen Knoten auszuwählen, der diese Aufgabe übernimmt. Danach fangen die Knoten an, Adressanfragen zu stellen. Dies ist etwa mit dem Kaltstart einer ganzen Computerfarm vergleichbar. Dabei fungiert ein Knoten als Adress-Server ( $S$ ), der die anderen Knoten ( $C$ ) mit Adressen versorgt (siehe Abbildung 4.7). In diesem Szenario werden zwei Fälle unterschieden:

- (1) **Eine Ebene:** Alle Knoten können den Adress-Server direkt erreichen. Es ist keine Weiterleitung des Steuerverkehrs erforderlich. Dieses Szenario ist analog zu einer Gruppenbesprechung mit wenigen Teilnehmern, in der alle Teilnehmer ihre Laptops verwenden und einer der Laptops als Adress-Server dient.
- (2) **Mehrere Ebenen:** Nicht alle Knoten können den Adress-Server direkt erreichen. Der Steuerverkehr muss über andere Knoten weitergeleitet werden (siehe Abbildung 4.8). Knoten, die sich auf der gleichen Ebene befinden, können direkt miteinander kommunizieren.

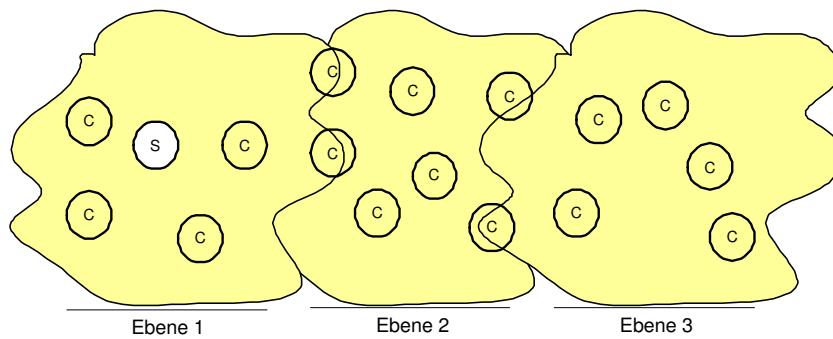


Abbildung 4.8: Anmeldung über mehrere Hops. Der Adress-Server befindet sich auf Ebene 1 und die restlichen Knoten sind auf drei Ebenen verteilt.

Diese beiden Fälle des Szenarios dienen zur Untersuchung der Eignung der Adressierungsverfahren für multi-hop Ad-hoc-Netze.

#### 4.5.2 Vereinigung von mehreren Netzen

In diesem Szenario entsteht aus mehreren Ad-hoc-Netzen ein neues Ad-hoc-Netz. Die Knoten aus den ursprünglichen Netzen bewegen sich aufeinander zu bis sie nur noch ein Netz bilden (siehe Abbildung 4.9). Nach der vollständigen Verschmelzung der ursprünglichen Netze, darf im neuen Netz nur

noch ein Adress-Server aktiv sein, um die Eindeutigkeit der Adressen zu gewährleisten. Hierfür muss unter Umständen das gesamte Netz oder ein Teil readressiert werden.

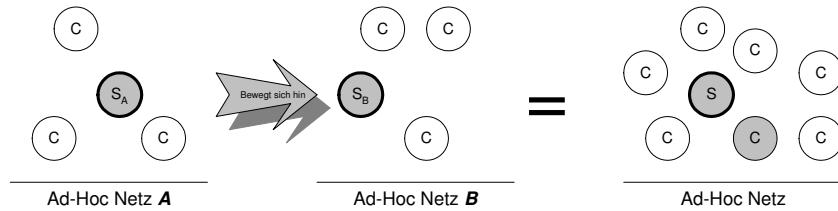


Abbildung 4.9: Vereinigung von Ad-hoc-Netzen.

Ein Beispiel aus der Realität zur Verdeutlichung dieses Szenarios ist etwa das Zusammentreffen zweier Arbeitsgruppen, die vorher für sich alleine gearbeitet hatten und nun gemeinsame Ergebnisse besprechen wollen. Aus beiden Gruppen entsteht eine neue und größere Gruppe.

#### 4.5.3 Aufteilung eines Netzes in mehrere Netze

Das dritte Szenario ist das Gegenstück zu dem letzten Fall. Ein großes Netz teilt sich in mehrere kleinere Ad-hoc-Netze auf. Die neuen Ad-hoc-Netze benötigen einen eigenen Adress-Server, der die Knoten versorgt (siehe Abbildung 4.10). Ein vergleichbares Beispiel aus der Realität könnte die folgende Situation sein. Innerhalb einer großen Gruppe bilden sich mehrere kleine Arbeitsgruppen, die eigene Besprechungen durchführen wollen.

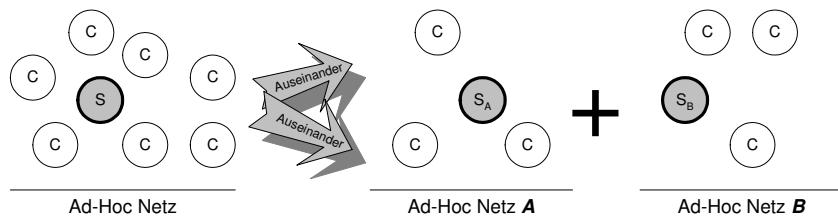


Abbildung 4.10: Aufspaltung eines Ad-hoc-Netzes in mehrere kleine Netze.

Die letzten beiden Szenarien dienen zur Untersuchung der *Adaptivität* und *Robustheit* der Verfahren. Dabei ist vorstellbar, dass dynamisch mehrere Ad-hoc-Netze entstehen, deren Aufbau und Topologie sich mit der Zeit ändert,

d.h. neue Ad-hoc-Netze verschmelzen und andere teilen sich in neue Ad-hoc-Netze auf.

## 4.6 Ergebnisse

Ergebnisse aus Simulationen werden in diesem Abschnitt zur Diskussion der Leistungsfähigkeit der Agentenbasierten Adressierung präsentiert. Die Diskussion folgt den vorgestellten Szenarien aus Abschnitt 4.5. Um eine Vergleichsmöglichkeit zu haben, wurde die Adressierung durch Autokonfiguration ebenfalls implementiert. In den Simulationen mit einer Adressierung mittels Autokonfiguration verfährt jeder Knoten nach dem Verfahren aus Abschnitt 4.3.1.

### 4.6.1 Vollständige Adressierung eines Netzes

Die Ergebnisse dieses Abschnitts beruhen auf Simulationen mit bis zu 50 Knoten, die auf mehrere Ebenen verteilt sind. In den Fällen mit nur einer Ebene können alle Knoten direkt miteinander kommunizieren, sodass keine multi-hop Kommunikation nötig ist. In den Simulationen mit mehreren Ebenen können nur die Knoten auf der gleichen Ebene direkt miteinander kommunizieren. Die Kommunikation zweier Knoten auf unterschiedlichen Ebenen erfolgt per multi-hop. In den Simulationen mit der Agentenbasierten Adressierung befindet sich der Adressierungsagent immer auf der ersten Ebene.

Die Simulationen wurden mit 10, 15, 20, 25, 30, 35, 40, 45 und 50 Knoten, die auf 1, 2, 3, 4 und 5 Ebenen verteilt sind, durchgeführt. Nach Möglichkeit wurden die Knoten auf alle vorhandenen Ebenen gleichverteilt. War die Anzahl der Knoten nicht ohne Rest durch die Anzahl der Ebenen teilbar, wurden die restlichen Knoten auf der letzten Ebene platziert. So befanden sich z.B. bei einer Simulation mit 35 Knoten und 3 Ebenen auf den ersten beiden Ebenen jeweils 11 und auf der letzten Ebene 13 Knoten.

#### Autokonfiguration

Bei den Simulationen mit Autokonfiguration wurde die Zeit gemessen, bis alle Knoten im Ad-hoc-Netz eine Adresse gewählt, das Netz auf Adresskollision getestet und das Ergebnis abgewartet haben.

Abbildung 4.11 stellt die Ergebnisse der Adressierung durch Autokonfiguration mit bis zu fünf Ebenen, auf die bis zu 50 Knoten verteilt wurden, dar.

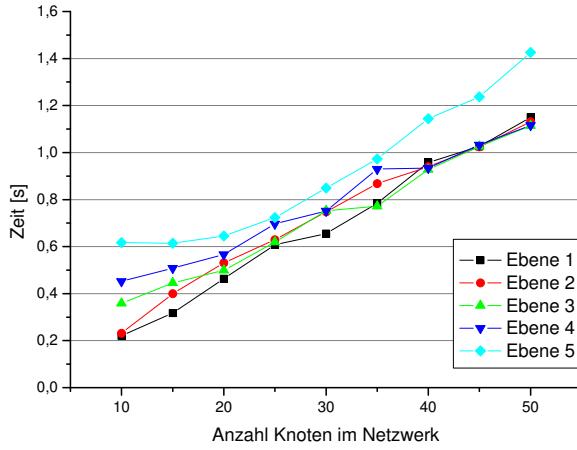
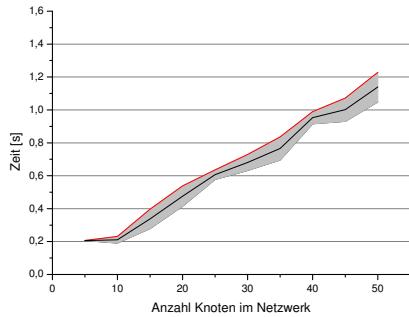


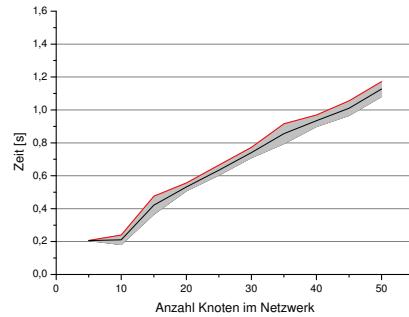
Abbildung 4.11: Vollständige Adressierung eines Ad-hoc-Netzes über fünf Ebenen mit Autokonfiguration.

Alle Ergebnisse liegen sehr nahe beieinander im Bereich von 0,2 - 1,6 Sekunden. Hierbei liegen die Ergebnisse der Simulationen mit 1-4 Ebenen unter 1,2 Sekunden und nur die Ergebnisse der Simulationen mit 5 Ebenen überschreiten diese Grenze. Die benötigte Zeit, um ein bestimmtes Ad-hoc-Netz vollständig zu adressieren, wächst linear mit der Anzahl der Knoten. Nur in den Simulationen mit 50 Knoten gibt es Abweichungen, die auf Paketkollisionen zurückzuführen sind. Bemerkenswert ist auch, dass bei den Simulationen mit 5 Ebenen erst ab 20 Knoten die benötigte Zeit anwächst. Dies ist auf die räumliche Verteilung der Knoten zurückzuführen. Auch die am Anfang benötigte höhere Zeit ist damit zu erklären. Obwohl bei den Simulationen mit 10 Knoten, also 2 Knoten pro Ebene, auch eine Kommunikation möglich sein sollte, ist es des öfteren aufgetreten, dass Pakete verworfen wurden. Für die vollständige Adressierung eines Ad-hoc-Netzes mit 50 Knoten, die über 5 Ebenen verteilt sind, benötigt die Autokonfiguration weniger als 1,6 Sekunden. In der Abbildung 4.12 sind die Ergebnisse für die jeweiligen Szenarien mit 1, 2, 3, 4 und 5 Ebenen mit  $\alpha = 0,05$ -Konfidenzintervall dargestellt.

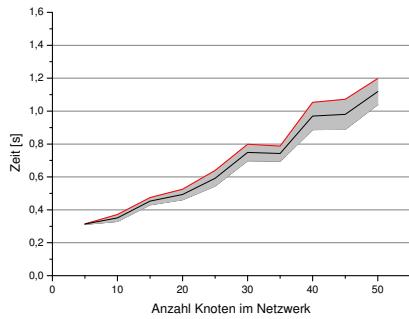
Die für die vollständige Adressierung mit Autokonfiguration notwendige Zeit hängt von zwei Faktoren ab: i.) die Zeit, die der Knoten auf eine Nachricht aus dem Netz wartet und ii.) die Anzahl der Versuche, bis der Knoten eine freie IP-Adresse findet. Für die zu wartende Zeit wird in [PKP01] der Defaultwert von  $60 \text{ ms} \times \text{Netzdurchmesser}$  vorgeschlagen. In den durchgeföhrten Simulationen betrug der maximale Netzdurchmesser 4, womit die Wartezeit



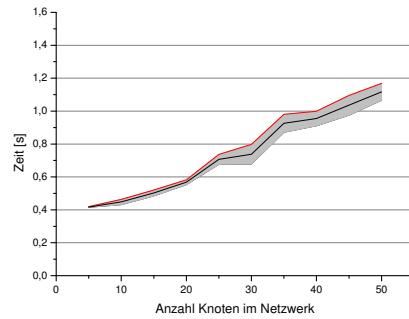
(a) Anmeldung auf einer Ebene.



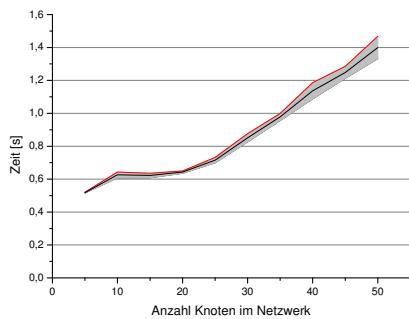
(b) Anmeldung über zwei Ebenen.



(c) Anmeldung über drei Ebenen.



(d) Anmeldung über vier Ebenen.



(e) Anmeldung über fünf Ebenen.

Abbildung 4.12: Vollständige Adressierung eines Ad-hoc-Netzes mit Auto-konfiguration mit  $\alpha = 0,05$ -Konfidenzintervall.

bei 240 ms liegt. Bei größeren Netzen könnte sich die Wartezeit als großer Nachteil erweisen, z.B. beträgt die Wartezeit bei einem Netzdurchmesser von 40 schon 2,4 Sekunden.

Die durchschnittliche Anzahl der Versuche, bis der Knoten  $n$  eine freie Adresse findet, kann analog zum Geburtstagsproblem berechnet werden, wenn man annimmt, dass schon  $n - 1$  Knoten im Netz sind und eine Adresse haben. Dabei wird von einem Klasse B Netz ausgegangen, d.h.  $2^{16}$  Adressen. Für den 50. Knoten beträgt die Wahrscheinlichkeit für einen Konflikt beim ersten Versuch 0,018. Für den 305. Knoten beträgt die Konfliktwahrscheinlichkeit beim ersten Versuch schon 0,507 und den 780. Knoten 0,99. Bei wenigen Knoten kann man also davon ausgehen, dass keine Adresskonflikte auftreten.

### Agentenbasierte Adressierung

Bei den Simulationen mit der Agentenbasierten Adressierung wurde die Zeit von der Versendung der ersten Verify-Nachricht vom Adressierungsagenten, dem Erhalt aller Bestätigungsnotizen von den Knoten bis zur Versendung einer zweiten Verify-Nachricht vom Adressierungsagenten, die die eindeutigen Adressen enthält, gemessen. Wichtig ist hier, dass der Adressierungsagent nach dem Empfang der letzten Bestätigungsnotiz die zweite Verify-Nachricht versendet.

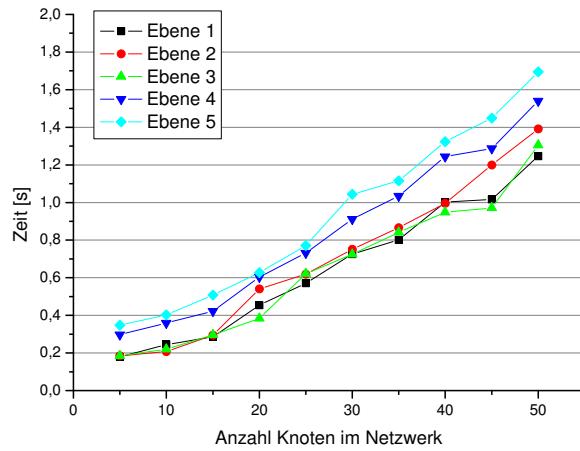
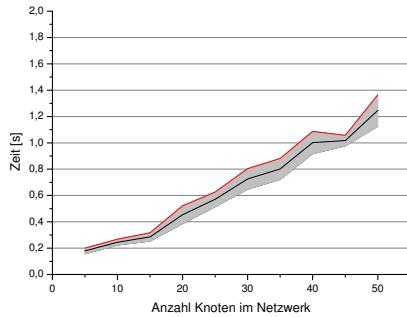
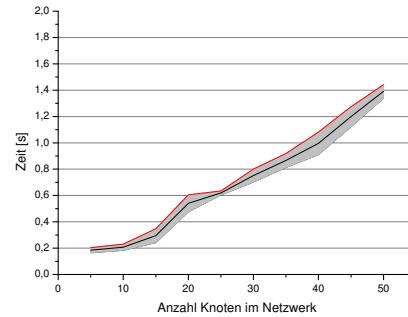


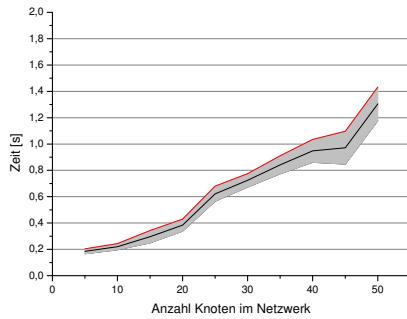
Abbildung 4.13: Vollständige Adressierung eines Ad-hoc-Netzes über mehrere Ebenen mit der Agentenbasierten Adressierung.



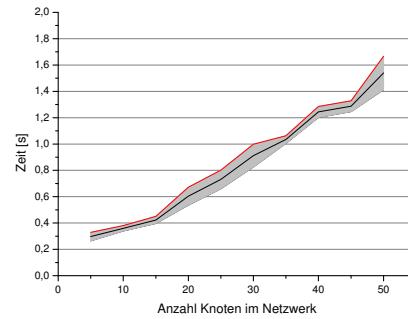
(a) Anmeldung auf einer Ebene.



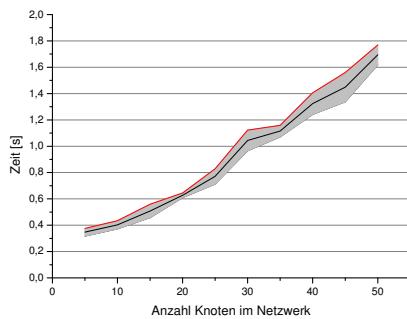
(b) Anmeldung über zwei Ebenen.



(c) Anmeldung über drei Ebenen.



(d) Anmeldung über vier Ebenen.



(e) Anmeldung über fünf Ebenen.

Abbildung 4.14: Vollständige Adressierung eines Ad-hoc-Netzes mit der Agentenbasierten Adressierung mit  $\alpha = 0,05$ -Konfidenzintervall.

Die Abbildung 4.13 stellt die Ergebnisse der Simulationen mit der Agentenbasierten Adressierung dar. Alle Ergebnisse sind nahe beieinander und den Ergebnissen der Autokonfiguration ähnlich. Das ist nicht weiter verwunderlich, da der Unterschied zwischen den beiden Ansätzen bei diesen Szenarien sehr klein ist. Im Gegensatz zu der Autokonfiguration müssen die Knoten bei der Agentenbasierten Adressierung auf eine Verify-Nachricht vom Adressierungsagenten warten, bevor sie sich bei ihm anmelden. Mit zunehmender Anzahl von Ebenen nimmt die benötigte Zeit für die Adressierung zu. Der eigentliche Faktor ist aber die Anzahl der Knoten, da nicht alle Knoten gleichzeitig das Netz fluten können, d.h. die benötigte Zeit wächst linear mit der Anzahl der Knoten an. In der Abbildung 4.14 sind die Ergebnisse für die jeweiligen Szenarien mit 1, 2, 3, 4 und 5 Ebenen mit  $\alpha = 0,05$ -Konfidenzintervall dargestellt.

#### 4.6.2 Vereinigung von zwei Netzen zu einem Netz

Bei diesem Szenario werden zwei Ad-hoc-Netze betrachtet. Die Knoten von Ad-hoc-Netz-1 bewegen sich auf Ad-hoc-Netz-2 mit maximal 5 m/s zu, bis die Knoten der beiden Netze sich zu einem Netz vereinigt haben. Dabei sind die Knoten in beiden Netzen konzentrisch angeordnet. Bei den Simulationen mit der Agentenbasierten Adressierung bildet der Adressierungsagent den Mittelpunkt und bei den Simulationen mit Autokonfiguration ist der Mittelpunkt virtuell.

Die Simulationen wurden mit 10, 20, 30, 40 und 50 Knoten durchgeführt, die auf beide Ad-hoc-Netze aufgeteilt wurden. Die Knoten wurden auf beide Netze prozentual von 10% bis 90% in 10% Schritten aufgeteilt. In den Graphen ist auf der  $x$ -Achse der Anteil der Knoten aufgezeichnet, die sich in Ad-hoc-Netz-2 befinden.

#### Autokonfiguration

Bei den Simulationen mit Autokonfiguration wurde, sobald alle Knoten einen zusammenhängenden Graphen bilden, eine Readressierung der Knoten in Ad-hoc-Netz-1 durchgeführt. Die Readressierung wurde vom Simulationsskript angestoßen, da der Ansatz die Vereinigung von zwei Netzen nicht selbstständig erkennen kann. Gemessen wurde die Zeit vom Anstoß der Readressierung bis zum vollständigen Readressieren der Knoten in Ad-hoc-Netz-1.

In Abbildung 4.15 sind die Ergebnisse für die Adressierung mit Autokonfiguration dargestellt. Alle Ergebnisse liegen zwischen 0,2 und 0,9 Sekunden. Aus den Ergebnissen sind zwei Trends zu erkennen. Der erste ist, dass mit zunehmender Anzahl von Knoten die benötigte Zeit für die Readressierung

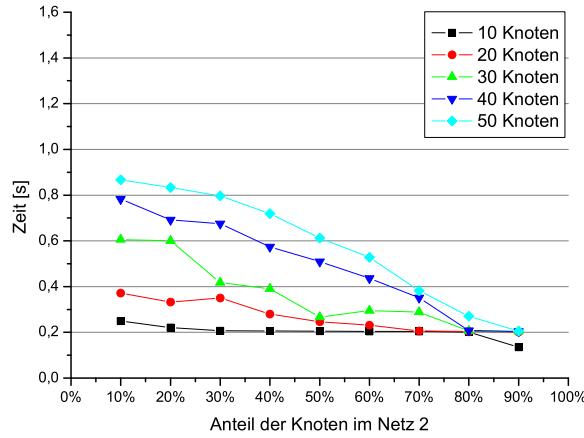


Abbildung 4.15: Vereinigung zweier Ad-hoc-Netze zu einem neuen Ad-hoc-Netz mit Autokonfiguration.

der Knoten wächst. Der zweite Trend ist, dass mit der Zunahme des Anteils an Knoten in Ad-hoc-Netz-2 die benötigte Zeit abnimmt. Beide Trends waren zu erwarten, da bei größerer Gesamtzahl an Knoten mehr Daten im Netzwerk geflutet werden und dies mehr Zeit in Anspruch nimmt. In der Abbildung 4.16 sind die Ergebnisse für die jeweiligen Szenarien mit 10, 20, 30, 40 und 50 Knoten mit  $\alpha = 0,05$ -Konfidenzintervall dargestellt.

### Agentenbasierte Adressierung

Bei den Simulationen mit der Agentenbasierten Adressierung existiert in beiden Netzen jeweils ein Adressierungsagent. Sobald die Knoten einen zusammenhängenden Graphen bilden und somit beide Netze Nachrichten austauschen können, sind auch die Adressierungsagenten in der Lage, die Verify-Pakete des anderen zu empfangen. Danach wird die Wahl zwischen den Adressierungsagenten durchgeführt. Die Knoten des Netzes, dessen Adressierungsagent die Wahl verloren, werden readressiert. Die im Folgenden dargestellten Ergebnisse entsprechen der Zeit in der zunächst ein Adressierungsagent von einem anderen Adressierungsagenten eine Verify-Nachricht empfängt und im Anschluss eine Readressierung der entsprechenden Knoten durchgeführt wird.

In Abbildung 4.17 sind die Ergebnisse für die Agentenbasierte Adressierung abgebildet. Die Zeit für die vollständige Adressierung des Endnetzes bewegt

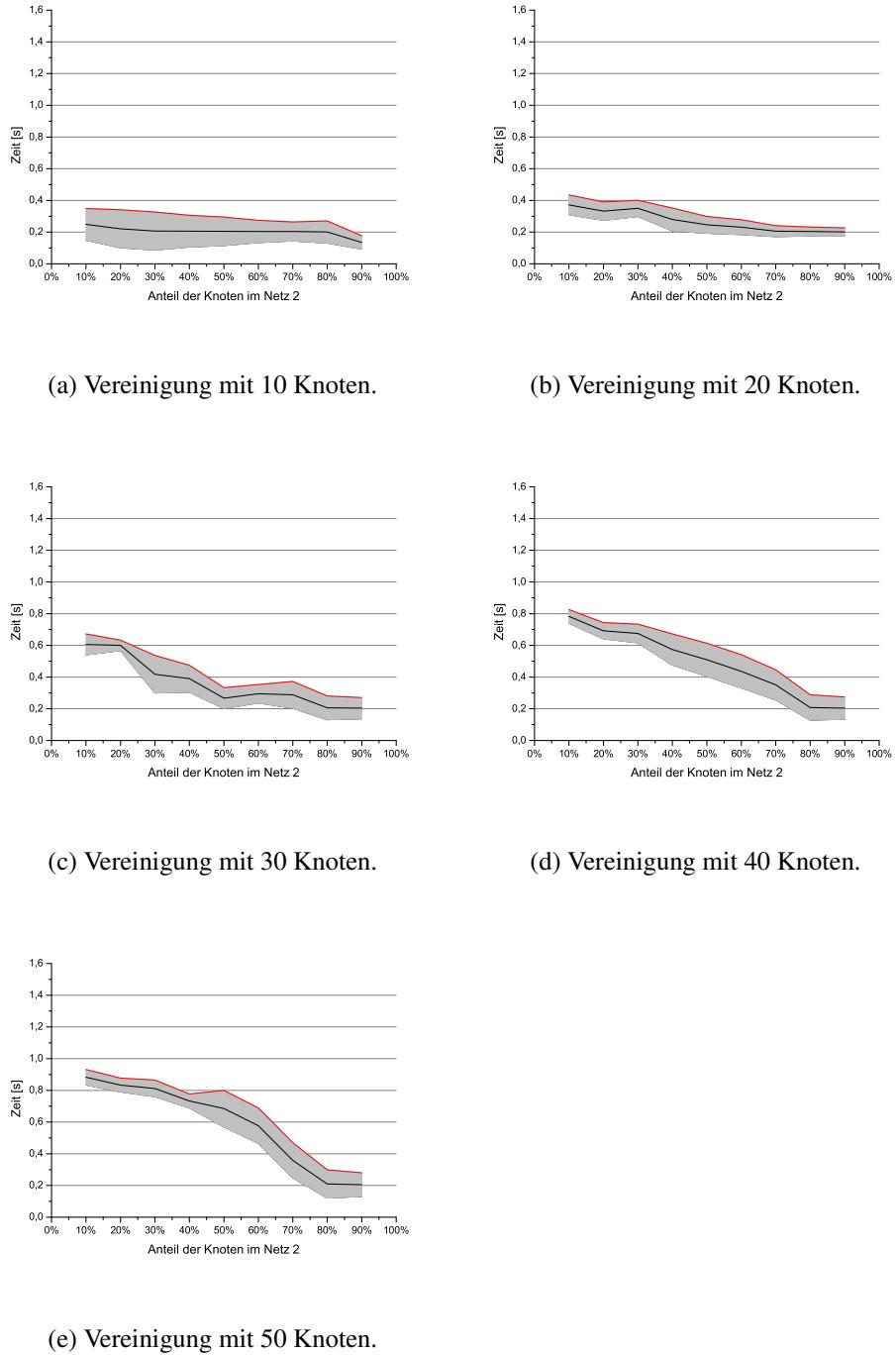


Abbildung 4.16: Vereinigung zweier Ad-hoc-Netze zu einem neuen Ad-hoc-Netz mit Autokonfiguration mit  $\alpha = 0,05$ -Konfidenzintervall.

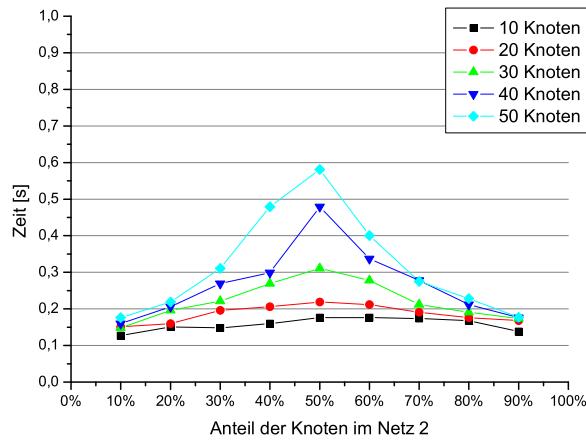


Abbildung 4.17: Vereinigung zweier Ad-hoc-Netze zu einem neuen Ad-hoc-Netz mit Agentenbasierter Adressierung.

sich zwischen 0,1 und 0,6 Sekunden. Aus den Ergebnissen lässt sich erkennen, dass die benötigte Zeit bis etwa 50% anwächst und danach abnimmt, wobei in den Szenarien mit mehr Knoten die benötigte Zeit auch anwächst.

Der Grund für die Spitze bei 50% liegt darin, dass die Entscheidung bei der Wahl des Adressierungsagenten primär auf der Anzahl der Knoten getroffen wird, die bei den Adressierungsagenten registriert sind. Deshalb fällt die Entscheidung bis 49% auf den Adressierungsagenten in Ad-hoc-Netz-1. Ab 51% fällt die Entscheidung für den Adressierungsagenten in Ad-hoc-Netz-2. Bei exakt 50% und somit gleicher Anzahl an registrierten Knoten wird die Entscheidung aufgrund der MAC-Adresse gefällt. Der Knoten mit der kleineren MAC-Adresse gewinnt die Wahl. In der Abbildung 4.18 sind die Ergebnisse für die jeweiligen Szenarien mit 10, 20, 30, 40 und 50 Knoten mit  $\alpha = 0,05$ -Konfidenzintervall dargestellt.

### 4.6.3 Aufteilung eines Netzes in mehrere Netze

In diesem Szenario teilt sich ein großes Ad-hoc-Netz in mehrere kleinere Netze auf, dabei benötigen die im ursprünglichen Netz verbleibenden Knoten keine neuen Adressen. Im Gegensatz dazu benötigen die Knoten, die sich abgespalten haben und nun in einem neuen Netz auftauchen neue Adressen, um mit den anderen Knoten zu kommunizieren.

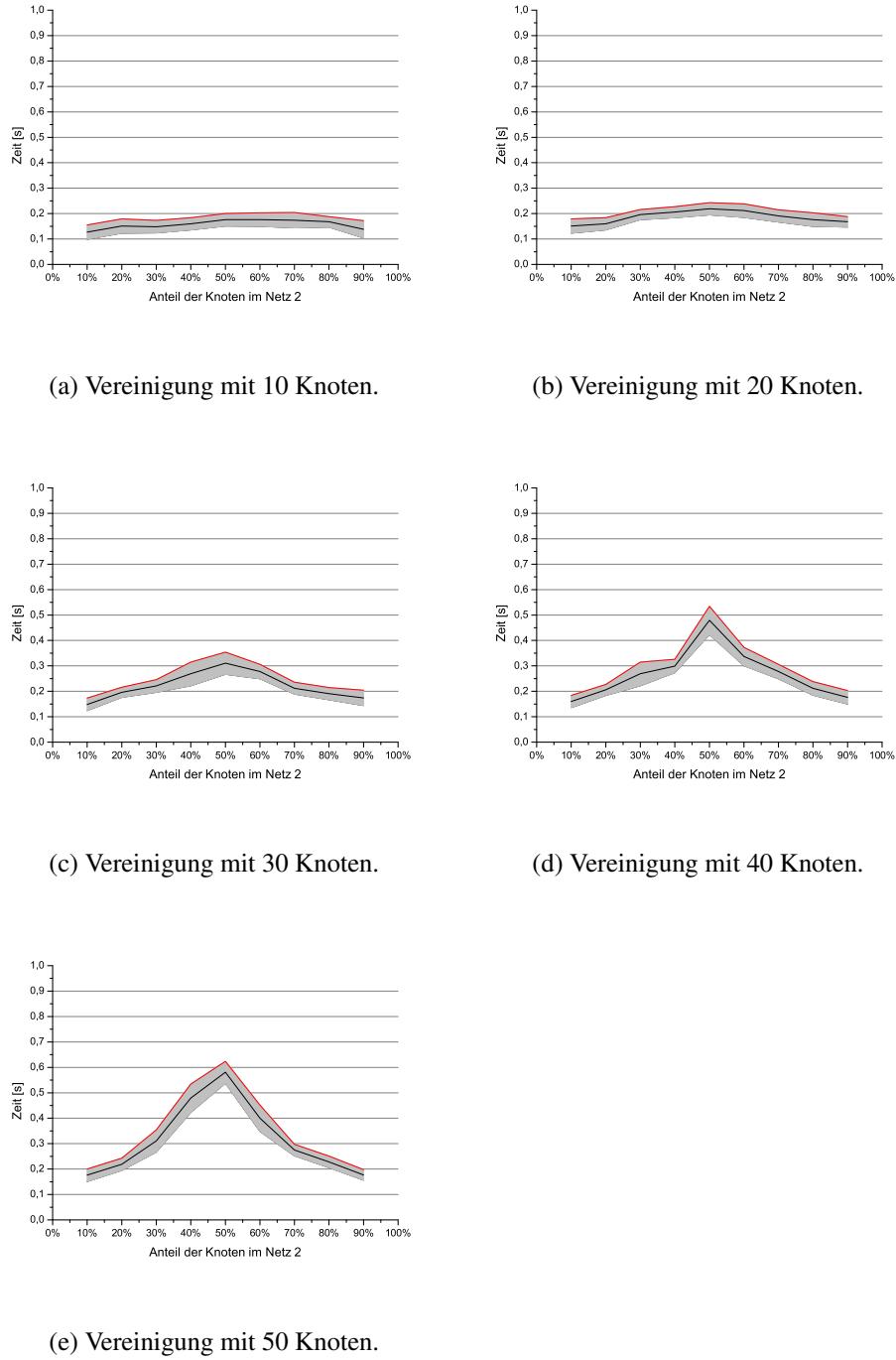


Abbildung 4.18: Vereinigung zweier Ad-hoc-Netze zu einem neuen Ad-hoc-Netz mit Agentenbasierter Adressierung mit  $\alpha = 0,05$ -Konfidenzintervall.

## Autokonfiguration

Die Adressierung mit Autokonfiguration sieht keinen Mechanismus für das Erkennen von Aufteilung bzw. Vereinigung von Ad-hoc-Netzen vor. Um bei den Untersuchungen eine Vergleichsmöglichkeit zu haben, wurde die Erkennung vom Simulationsskript aus gesteuert. Dabei wird wie folgt vorgegangen. Die Knoten im ursprünglichen Netz behalten ihre Adressen. Die Knoten im neuen Netz werden readressiert.

Die Simulationsergebnisse stimmen mit den Ergebnissen für das erste Szenario mit entsprechender Anzahl an Knoten und einer Ebene überein, da die gemessene Zeit ab der Abspaltung bis zur vollständigen Readressierung des neuen Netzes gemessen wurde.

## Agentenbasierte Adressierung

Der Adressierungsagent registriert bei seinen regelmäßigen Aktualisierungen die ausgeschiedenen Knoten. Diese IP-Adressen werden markiert und können später neu vergeben werden. Die Knoten im neuen Netz warten auf die Verify-Nachricht eines Adressierungsagenten. Danach beginnt die Adressierung. Die Bestimmung des Adressierungsagenten verläuft gemäß dem Verfahren aus Abschnitt 4.4.2.

Auch bei der Agentenbasierten Adressierung stimmen die Ergebnisse mit den Ergebnissen des ersten Szenarios überein, da die Messung sich nur auf das abgespaltene Netz bezieht und damit identisch mit der vollständigen Adressierung eines kleinen Netzes ist.

## Zusammenfassung der Ergebnisse

Die vorgestellte Leistungsbetrachtung konzentrierte sich auf die benötigte Zeit, um ein bestimmtes Ad-hoc-Netz zu adressieren. Andere Kenngrößen wie Overhead wurden nicht betrachtet. Nachfolgend erfolgt eine Zusammenfassung der diskutierten Ergebnisse für beide betrachteten Verfahren.

## Autokonfiguration

Die Autokonfiguration arbeitet schnell, zuverlässig und besitzt den Vorteil, dass keine zentrale Einheit für die Adressierung benötigt wird. Deshalb eignet es sich grundsätzlich für den Einsatz in Ad-hoc-Netzen. Das Verfahren besitzt in der vorgestellten Version in [PKP01] auch einige Schwächen, die gegen einen sinnvollen Einsatz des Verfahrens sprechen. So ist die Eindeutig-

keit der Adressen im gesamten Ad-hoc-Netz nicht gewährleistet. Die Überprüfung findet jeweils nur einmal beim Start eines Knotens statt. Weiterhin kennt das Verfahren keine Mechanismen für das Erkennen der Aufteilung und Vereinigung von Ad-hoc-Netzen.

Die in diesem Abschnitt gezeigten Ergebnisse basieren auf einigen Annahmen, die in der Realität nicht zutreffen müssen. Die Wartezeit der Knoten ist ein sehr wichtiger Faktor bei den Ergebnissen. In den hier durchgeföhrten Simulationen wurde die Wartezeit der Knoten entsprechend den Szenarien angepasst. Es wird auch angenommen, dass alle Knoten in einem Ad-hoc-Netz einen zusammenhängenden Graphen bilden und die verwendete Fluttechnik tatsächlich alle Knoten des Netzes erreicht, was nicht zutreffen muss.

### Agentenbasierte Adressierung

Die Agentenbasierte Adressierung erfüllt die gestellten Anforderungen aus Abschnitt 4.4. Sie eignet sich für den Einsatz in Ad-hoc-Netzen, da sie die gestellte Aufgabe sowohl schnell als auch zuverlässig erfüllt. Im Gegensatz zu anderen Verfahren ist die Agentenbasierte Adressierung in der Lage, die Aufteilung und Vereinigung von Ad-hoc-Netzen zu erkennen. Die Eindeutigkeit der vergebenen Adressen wird durch den Adressierungsagenten sichergestellt. Weiterhin ist das Verfahren in der Lage die Anzahl der Adressierungsagenten dynamisch anzupassen.

Bei dem diskutierten Verfahren wurde die tatsächliche benötigte Zeit gemessen. Das in Abschnitt 4.4 vorgestellte Verfahren basiert jedoch auf der regelmäßigen Aussendung von Verify-Nachrichten. Hierdurch entspricht die benötigte Zeit für die Adressierung immer einem Vielfachen des Sendeintervalls von Verify-Nachrichten. Hiermit hängt auch ein weiteres Problem zusammen, nämlich die Bestimmung des Sendeintervalls von Verify-Nachrichten. Es ist wünschenswert, dass das Sendeintervall dynamisch an die Mobilität des Ad-hoc-Netzes angepasst wird. Ähnlich zum Slow-Start-Mechanismus von TCP könnte der Adressierungsagent nach dem Starten in kurzen Intervallen Verify-Nachrichten aussenden und später das Intervall vergrößern. Nach einer Aufteilung des Netzes und der Vereinigung von mehreren Netzen könnte das Intervall wieder verkleinert werden.

## 4.7 Fazit

In diesem Kapitel wurde die automatische Adresskonfiguration in mobilen multi-hop Ad-hoc-Netzen diskutiert und die Agentenbasierte Adressierung vorgestellt und bewertet. Die Untersuchung beruht auf einer Auswahl von

Szenarien, die für zukünftige Ad-hoc-Netze typische Situationen darstellen.

Die Agentenbasierte Adressierung erfüllt die geforderten Anforderungen von mobilen Ad-hoc-Netzen und arbeitet zuverlässig. Das Verfahren ist einfach, robust, adaptiv und in der Lage, die Aufteilung und Vereinigung von Ad-hoc-Netzen zu erkennen. Ein Manko des Verfahrens ist die hohe Last, die für die Adressierung benötigt und durch das Fluten von Daten im Netzwerk verursacht wird.

Die Adressierung ist eine wichtige Grundlage der Kommunikation in Netzen. Daher erfordert diese Fragestellung in mobilen Ad-hoc-Netzen weitere Forschungsarbeiten. Offene Fragen in diesem Bereich sind die Skalierbarkeit, der Schutz gegen Angriffe, die Reduzierung der erzeugten Last und die Anbindung von Ad-hoc-Netzen an vorhandene lokale Netze oder das Internet.



---

## KAPITEL 5

---

# Routing in Ad-hoc-Netzen

Eine der wichtigsten Funktionen eines Netzes ist das Routing, das für die effiziente Wegewahl zuständig ist und dadurch die Übertragung von jeglichen Informationen in einem Netz beeinflusst. Deshalb hängt die Leistungsfähigkeit eines Netzes sehr stark mit dem Routing zusammen. Dies betrifft sowohl die Verfügbarkeit des Netzes im Allgemeinen, die Anzahl der bedienbaren Teilnehmer und die Anzahl der parallel im Netz übertragbaren Verbindungen. Diesen hohen Stellenwert hat das Routing auch in mobilen multi-hop Ad-hoc-Netzen. Jedoch sind Ad-hoc-Netze durch ihre inhärent schwierige Umgebung vielen zusätzlichen Problemen ausgesetzt, die sich auch auf das Routing auswirken.

Das Routing ist *eines* der Hauptprobleme in mobilen multi-hop Ad-hoc-Netzen, an dem seit Jahrzehnten gearbeitet wird. Durch die ständige Topologieänderung, welche durch die Knotenmobilität bedingt ist, müssen die Routingalgorithmen anderen Anforderungen genügen, als ihre Verwandten in Festnetzen. In den letzten Jahren wurden viele Routingalgorithmen vorgestellt. In [Per01] und [Toh02] findet man eine Übersicht. Jedoch ist keiner der Routingalgorithmen für alle Anwendungen geeignet. Oft sind die Routingalgorithmen im Bereich der mobilen Ad-hoc-Netze für spezielle Anwendungen konzipiert. Durch die Anwendungen sind besondere Rahmenbedingungen gegeben, wodurch Restriktionen entstehen, oder eine bestimmte Ausstattung der Knoten vorausgesetzt wird. Hier ist insbesondere die Benutzung von GPS für die örtliche Bestimmung der Knoten zu nennen. Im Allgemeinen kann jedoch nicht von solch einer Ausstattung ausgegangen werden.

In diesem Kapitel wird ein neuer Routingalgorithmus für mobile multi-hop Ad-hoc-Netze vorgestellt, der auf *Schwarmintelligenz* und im Speziellen auf *Ameisenalgorithmen* basiert. Die Besonderheit von Ameisenalgorithmen ist, dass sie durch die Kooperation von vielen einzelnen Individuen komplexe Probleme effizient lösen können. Die Ameisenalgorithmen ahmen das Ver-

halten von einfachen Ameisen zur Lösungsfindung nach. Die eigentlichen Akteure dabei sind die einzelnen Ameisen, die ihren Teil zur Lösung beitragen. Vor allem stellt die Art der Kommunikation der Ameisen untereinander einen sehr interessanten Ansatz dar, wodurch der Aufwand des Routings verringert werden kann.

Der Rest des Kapitels ist wie folgt aufgebaut. Abschnitt 5.1 stellt die Anforderungen, eine Klassifikation und die bekanntesten Routingalgorithmen für mobile multi-hop Ad-hoc-Netze vor. In Abschnitt 5.2 wird eine Einführung in das Gebiet der Schwarmintelligenz gegeben und in Abschnitt 5.3 die Grundlagen von Ameisenalgorithmen vorgestellt. Anschließend stellt Abschnitt 5.4 dann den Ameisenroutingalgorithmus im Detail vor. In Abschnitt 5.5 werden Ergebnisse aus Simulationen präsentiert und mit existierenden Routingalgorithmen verglichen. In Abschnitt 5.6 werden verwandte Arbeiten zum Ameisenroutingalgorithmus besprochen und die Unterschiede diskutiert. Das Kapitel schließt mit einer Zusammenfassung in Abschnitt 5.7 ab.

## 5.1 Routingalgorithmen für Ad-hoc-Netze

Die Entwicklung von Ad-hoc-Netzen begann parallel zur Entwicklung des Internets (siehe Abschnitt 2.1.2). In den ersten Ad-hoc-Netzen wurden zuerst Varianten der Distance-Vector Algorithmen und später dann der Link-State Algorithmen eingesetzt. Die Entwicklung von unterschiedlichen Ansätzen, um das Routing effizienter zu gestalten, begann Ende der 80’er und Anfang der 90’er Jahre des vergangenen Jahrhunderts. Innerhalb der IETF hat sich zu dieser Zeit eine Arbeitsgruppe, die MANET WG, gebildet, die sich mit der Entwicklung von Routingalgorithmen für mobile multi-hop Ad-hoc-Netze beschäftigt. Das Ziel der Arbeitsgruppe ist es, einen offenen Routingalgorithmus zu standardisieren [Iet].

### 5.1.1 Anforderungen an Routingalgorithmen

Die MANET Arbeitsgruppe der IETF stellt einige Anforderungen [MC98] an Routingalgorithmen für mobile Ad-hoc-Netze. Die wichtigsten Anforderungen sind:

- **Verteilte Verfahren**

Mobile multi-hop Ad-hoc-Netze sind ihrem Wesen nach Systeme über die keine garantierten Aussagen über Aufbau und Verhalten gemacht werden können. Knoten können sich zu beliebigen Zeiten am Netz anmelden, abmelden oder auch ganz vom Netz ausscheiden. Weiterhin

kann durch die Knotenmobilität eine Aufteilung des Netzes erfolgen, die temporär oder auch langfristig sein kann. Die Erreichbarkeit von Knoten ist in keiner Weise gewährleistet. Deshalb sind Verfahren, welche eine zentrale Steuerung erfordern, für mobile multi-hop Ad-hoc-Netze ungeeignet, da bei nicht Erreichbarkeit der zentralen Steuerung die gesamte Kommunikation zusammenbricht.

- **On-Demand Routing**

Es ist nicht zu erwarten, dass die Kommunikation in einem mobilen multi-hop Ad-hoc-Netz gleichmäßig auf alle Knoten verteilt ist. Aus diesem Grund sollen die Verfahren nach Bedarf arbeiten, um sowohl die Last zu minimieren, als auch andere Ressourcen wie Energie und Speicher zu schonen.

- **Unidirektionale Verbindungen**

Es ist wünschenswert, dass Routingalgorithmen in der Lage sind mit asymmetrischen Verbindungen umzugehen. In solchen Netzen erfolgt der Transport von Paketen in Hin- und Rückrichtung auf unterschiedlichen Pfaden. Dies kann bei der Funkkommunikation eine wichtige Rolle spielen, da Knoten in einem Netz mit unterschiedlichen Sendeeinheiten und Empfangseinheiten arbeiten können, die z.B. unterschiedliche Reichweiten besitzen.

- **Energiesparsam**

Die Knoten in einem mobilen Ad-hoc-Netz greifen als Energielieferanten auf Batterien zurück. Da die Energiequelle von Knoten eines der kostbarsten Ressourcen in einem Ad-hoc-Netz darstellt, ist der Umgang mit dieser Ressource besonders wichtig. Deshalb ist es wünschenswert, dass Routingalgorithmen Ruhephasen für die Knoten berücksichtigen. Ein Knoten, der sich in einer Ruhephase befindet, sollte immer noch im Netz erreichbar sein, jedoch nicht mehr aktiv an der Routingfunktion des Netzes teilnehmen.

- **Sicherheit**

Die Sicherheit spielt eine besonders wichtige Rolle in mobilen Ad-hoc-Netzen. Die Funkkommunikation kann erheblich einfacher abgehört werden als die Kommunikation in Festnetzen. Angreifer können durch Injizierung von falschen Routinginformationen die Wahl der Kommunikationswege beeinflussen. Dadurch können jedoch auch Knoten von der Kommunikation ausgeschlossen werden.

Es gibt keinen Routingalgorithmus für mobile multi-hop Ad-hoc-Netze, der alle Anforderungen erfüllt. Die meisten Routingalgorithmen konzentrieren

sich auf die effiziente Erfüllung der eigentlichen Routingfunktion. Jedoch wurden für die verschiedensten Routingansätze Erweiterungen vorgeschlagen, um weitere Anforderungen zu erfüllen. Diese betreffen hauptsächlich entweder die Energieschonung oder die Sicherheit.

### 5.1.2 Klassifikation von Routingalgorithmen

Routingalgorithmen können auf unterschiedliche Arten klassifiziert werden. Eine traditionelle Klassifikation besteht in der Unterscheidung, wie Routinginformationen im Netz gesammelt und verteilt werden. Dies erlaubt die Unterscheidung in *proaktive*, *reaktive* und *hybride* Routingalgorithmen [RT99, DCY00, Toh02]. Abbildung 5.1 stellt die Klassifikation der Routingalgorithmen grafisch dar. Zu jeder Klasse sind die bekanntesten Vertreter aufgeführt.

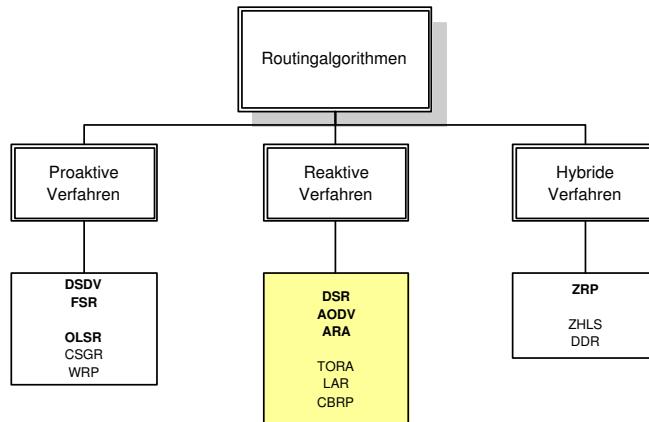


Abbildung 5.1: Klassifikation von Routingalgorithmen.

Proaktive Routingalgorithmen werden auch *tabellenbasierte* Routingalgorithmen genannt. Die Routingalgorithmen dieser Klasse versuchen, für jede Kommunikationskombination im Netz einen vorberechneten Pfad bereitzuhalten. Deswegen müssen die Algorithmen dieser Klasse kontinuierlich Routinginformationen sammeln und im Netz an alle Knoten verteilen, sodass die Knoten aktuelle Routinginformationen haben. Aus diesem Grund besitzt diese Klasse von Routingalgorithmen einen hohen Overhead, da das Sammeln und Verteilen von Routinginformationen aufwendig ist. Auf der anderen Seite können Verbindungen sehr schnell aufgebaut werden, weil die Routinginformationen schon vorhanden sind.

Im Gegensatz zu dieser Vorgehensweise ermitteln reaktive Routingalgorithmen, erst wenn er benötigt wird, einen Pfad zwischen dem Quell- und dem Zielknoten. Diese Klasse wird deshalb auch *On-Demand Routing* genannt. Der Vorteil ist, dass keine unnötigen Informationen im Netz anfallen und deshalb der Overhead im Vergleich zu den proaktiven Routingalgorithmen geringer ist. Der Nachteil dieser Klasse ist, dass der Aufbau einer bisher unbekannten Verbindung verzögert wird, da vorher ein Pfad zwischen dem Quell- und Zielknoten ermittelt werden muss. Dies kann unter Umständen eine längere Zeit in Anspruch nehmen.

Die hybriden Routingalgorithmen bestehen aus der Kombination von Routingalgorithmen aus den ersten beiden Klassen. Diese Klasse von Routingalgorithmen versucht die Vorteile von proaktiven und reaktiven Routingalgorithmen zu vereinen, ohne die Nachteile in Kauf nehmen zu müssen. Das Ergebnis ist meist ein komplexes Routingprotokoll, dass den Einsatz von unterschiedlichen Routingalgorithmen vorsieht.

Es hat sich gezeigt, dass in mobilen Ad-hoc-Netzen reaktive Routingalgorithmen eine bessere Leistung zeigen als proaktive Routingalgorithmen [DCY00, Per01, RT99, Toh02]. Der Grund liegt am hohen Overhead von proaktiven Routingalgorithmen, der durch das periodische Sammeln und Verteilen von Routinginformationen verursacht wird. Untersuchungen haben auch gezeigt, dass die in den Routingtabellen gespeicherten Informationen teilweise veraltet sind und sich dadurch nachteilig auf die Leistung auswirken.

### 5.1.3 Destination-Sequenced Distance-Vector Routing

Bei den Distance-Vector-Verfahren besitzt ein Knoten in seiner Routingtabelle einen Eintrag für jeden Zielknoten. Ein Routingtabelleneintrag gibt die Anzahl der Sprünge an, mit denen der Zielknoten über einen bestimmten direkten Nachbarn erreicht werden kann. Ein Knoten kennt also keinen vollständigen Pfad zwischen einem bestimmten Quell- und Zielknoten. Vielmehr besitzt jeder Knoten die Kenntnis, wie er ein Paket zu einem Zielknoten weiterleiten muss, damit dieser über den kürzesten Pfad transportiert wird. Um die Routingtabellen aktuell zu halten, müssen die Routinginformationen periodisch zwischen Nachbarknoten aktualisiert werden. Dies erzeugt eine nicht unerhebliche Last in großen Netzen. Es wird auch entsprechend viel Speicherplatz benötigt, da Routinginformationen für alle vorhandenen Knoten im Netz gespeichert werden müssen.

Das *Destination-Sequenced Distance-Vector Routing (DSDV)* [PB94, Per01] benutzt das klassische Distance-Vector-Routing, um kürzeste Pfade in mobilen Ad-hoc-Netzen zu berechnen. Bei DSDV werden die Routingtabellen aktualisiert, sobald wichtige Ereignisse im Netz eintreten, wie z.B. die Ände-

rung der Nachbarschaft eines Knotens durch Knotenbewegung. Da hierdurch sehr viel Last im Netzwerk erzeugt wird, was in mobilen Ad-hoc-Netzen unerwünscht ist, wurde diese Vorgehensweise erweitert. Das DSDV kennt zwei unterschiedliche Arten des Aktualisierens von Routingtabellen. Beim *full-dump* werden vollständige Routingtabellen und beim *incremental-dump* nur die Unterschiede zum letzten full-dump unter den Knoten ausgetauscht. Bei DSDV besitzt jeder Routingtabelleneintrag eine Sequenznummer, die so genannte Destination-Sequence die vom Zielknoten generiert wird. Hierdurch sind die Knoten in der Lage, alte Informationen von neuen zu unterscheiden und darauf basierend Schleifen zu verhindern.

#### 5.1.4 Ad hoc On-Demand Distance Vector Routing

Das *Ad hoc On-Demand Distance Vector Routing (AODV)* [PRD02] ist ein reaktives Verfahren, das auf dem klassischen Distance-Vector-Routing basiert. AODV wurde entworfen, um Schwächen des Vorgängers – dem *Destination-Sequenced Distance Vector* – zu verbessern. Das Ziel war die Anzahl der benötigten Pakete, die im gesamten Netz geflutet werden, zu minimieren. Bei DSDV lösen Knotenbewegungen die Aktualisierung von Routinginformationen im gesamten Netz aus. Dies führt jedoch zu einer sehr hohen Last im Netz. Im Gegensatz zu DSDV haben Knotenbewegungen nur eine lokale Auswirkung bei AODV. Dies wird dadurch erreicht, dass eine Aktualisierung von Informationen nur dann initiiert wird, wenn eine laufende Datenübertragung von dem Ereignis betroffen ist. Weiterhin werden nicht mehr für alle Knoten Informationen in den Routingtabellen bereitgehalten, sondern nur die benötigten. Die Einträge der Routingtabellen besitzen eine bestimmte Lebensdauer, die bei jeder Verwendung aktualisiert werden. Die Einträge, die innerhalb der Lebensdauer nicht benutzt wurden, werden dagegen nach Ablauf der Lebensdauer gelöscht. Das AODV besteht aus zwei Phasen, die abwechselnd verwendet werden:

- (1) **Pfadsuche:** dient dem Auffinden eines Pfades zwischen dem Quell- und Zielknoten.
- (2) **Pfadpflege:** dient der Pflege des aktuell benutzten Pfades zwischen einem Quell- und Zielknoten.

Ein Quellknoten, der Pakete zu einem Zielknoten schicken möchte, sucht in seiner Routingtabelle nach einem Pfad. Ist die Suche erfolgreich, kann die Übertragung der Pakete beginnen. Bei erfolgloser Suche geht der Quellknoten in die erste Phase – die Pfadsuche – über.

## Pfadsuche

In dieser Phase schickt der Quellknoten ein *Route-Request*-Paket aus, das im gesamten Netz geflutet wird. Das Route-Request-Paket besteht aus den Adressen des Quellknotens und des Zielknotens, sowie einem Zähler für die zurückgelegte Distanz und jeweils einer Sequenznummer des Quellknotens und des Zielknotens. Weiterhin enthält das Route-Request-Paket eine so genannte *Broadcast-ID*, die der Quellknoten mit jeder Route-Request erhöht. Die Adresse des Quellknotens und die Broadcast-ID stellen zusammen die eindeutige Identifizierung des Route-Request-Pakets sicher. Die Sequenznummern dienen der Vermeidung von Schleifen. Die Sequenznummer des Zielknotens übernimmt der Quellknoten aus seiner Routingtabelle.

Ein Knoten, der ein Route-Request-Paket erhält, kann aufgrund der Quellknotenadresse und der Broadcast-ID überprüfen, ob er das Paket schon einmal erhalten hat. Somit können Duplikate erkannt und verworfen werden. Ansonsten erstellt der Knoten aus der Quellknotenadresse, der Sequenznummer des Quellknotens, der Adresse des Nachbarknotens von dem der Knoten das Paket erhalten hat und aus dem Distanzzähler einen so genannten *Reverse-Route*-Eintrag. Dieser Eintrag dient dem Knoten, um etwaige *Route-Reply*-Pakete an den Quellknoten weiterzuleiten. Die Reverse-Route-Einträge besitzen auch eine Lebenszeit. Wenn sie innerhalb dieser Zeitspanne nicht verwendet wurden, werden sie gelöscht.

Die Route-Request-Pakete werden mit so genannten *Route-Reply*-Paketen beantwortet, die prinzipiell von jedem Knoten im Netz stammen können. Um die Verwendung von veralteten Informationen zu verhindern, ist die Beantwortung jedoch an zwei Bedingungen verknüpft. Die erste Bedingung ist, dass ein beantwortungswilliger Knoten einen gültigen Eintrag in seiner Routingtabelle zum Zielknoten haben muss. Die zweite Bedingung ist, dass die Sequenznummer des Eintrags mindestens genau so groß ist wie die im Route-Request-Paket. Der Zielknoten kann selbstverständlich auf ein Route-Request-Paket immer antworten. Jeder Knoten der ein Route-Reply-Paket empfängt, richtet einen so genannten *Forward-Path*-Eintrag in seiner Routingtabelle ein. Dadurch ist er dann in der Lage später Datenpakete, die an den Zielknoten gerichtet sind, weiterzuleiten.

Sobald der Quellknoten ein Route-Reply-Paket erhält, kann die Datenübertragung beginnen. Wenn der Quellknoten mehrere Route-Reply-Pakete erhält, aktualisiert er seine Routingtabelle mit den Informationen aus den Route-Reply-Paketen, die einen besseren Pfad enthalten. Ein Pfad zwischen einem Quellknoten und einem Zielknoten wird solange verwendet, wie er gebraucht wird.

In Abbildung 5.2 ist die Pfadsuche von AODV an einem Beispiel dargestellt.

Im Beispiel antwortet nur der Zielknoten auf das Route-Request-Paket vom Quellknoten.

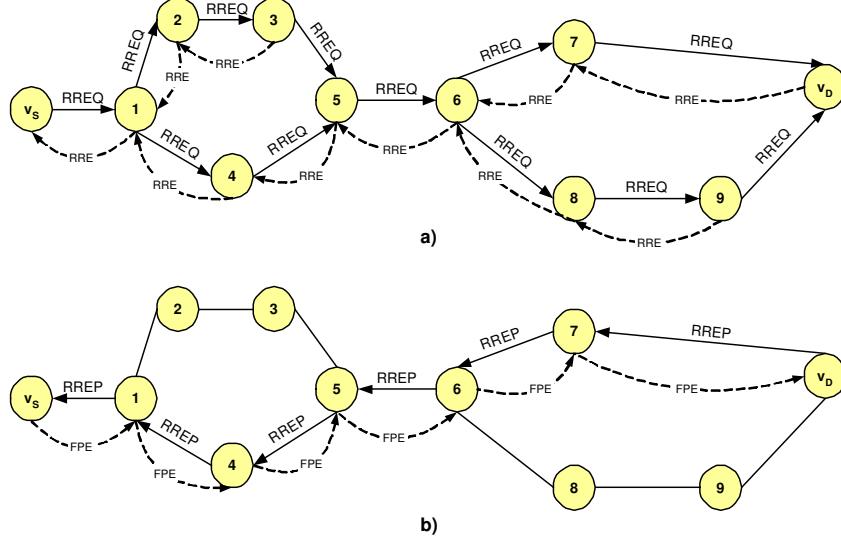


Abbildung 5.2: Die Pfadsuche bei AODV. a) Der Quellknoten  $v_s$  broadcastet ein Route-Request-Paket (RREQ) in das Netz. Jeder Knoten der das RREQ zum ersten mal empfängt legt einen Reverse-Route-Entry (RRE) Eintrag in seiner Routingtabelle an. b) Der Zielknoten antwortet mit einem Route-Reply-Paket (RREP), das mit Hilfe der RRE-Einträge an den Quellknoten transportiert wird. Dabei legt jeder Knoten, der das RREP-Paket erhält, einen Forward-Path-Entry (FPE) in seiner Routingtabelle an, sodass er in der Lage ist, zukünftige Datenpakete an den Zielknoten weiterzuleiten.

## Pfadpflege

Die Phase der Pfadpflege dient dazu, Fehler auf einem vorhandenen Pfad zu bereinigen. Fehler können z.B. durch Knotenmobilität verursacht werden, wodurch eine Verbindung zwischen zwei Knoten abbricht. AODV unterscheidet folgende Fälle:

- Der Quellknoten bewegt sich. In diesem Fall initiiert der Quellknoten eine neue Pfadsuche.
- Der Zielknoten oder ein anderer Knoten bewegt sich. Der Knoten, der den Fehler entdeckt, unterrichtet den Quellknoten über den Fehler mittels eines *Route-Error*-Pakets. Nach Erhalt dieser Information kann

der Quellknoten, sollte der Pfad noch benötigt werden, wiederum eine Pfadsuche starten.

### 5.1.5 Dynamic Source Routing

Das *Dynamic Source Routing (DSR)* ist ein reaktives Verfahren und basiert auf dem Source-Routing mit einigen Anpassungen für mobile Ad-hoc-Netze.

Die Idee hinter dem Source-Routing ist, dass der Sender für jedes Paket angibt, welchen Weg es durch das Netz nehmen soll. Hierzu wird in den Paketkopf der gesamte Weg vom Quellknoten bis zum Zielknoten eingetragen, z.B.  $v_S, v_1, v_2, \dots, v_D$ . Das bedeutet, der Quellknoten  $v_S$  sendet das Paket an den Knoten  $v_1$ , der es an  $v_2$  weiterleitet. Dies wird solange fortgesetzt bis das Paket beim Zielknoten  $v_D$  ankommt. Der Vorteil dieser Vorgehensweise ist, dass die Knoten im Netz sich nicht um Routinginformationen kümmern müssen. Ein Knoten, der ein Paket weiterleitet, schaut im Kopf des Paketes nach, an welchen Nachbarn er das Paket als nächstes senden muss. Die Hauptfrage hier ist, wie der Quellknoten einen Pfad zum Zielknoten ermittelt.

Das Dynamic Source Routing besteht aus zwei Phasen:

- (1) **Pfadsuche:** dient dem Auffinden eines Pfades zwischen einem Quell- und einem Zielknoten.
- (2) **Pfadpflege:** dient der Wartung existierender Pfade. Diese Phase kommt immer dann zum Einsatz, wenn auf einem aktuell benutzten Pfad ein Fehler auftritt.

Wenn ein Quellknoten ein Paket an einen Zielknoten versenden möchte, sucht er in seiner Routingtabelle nach einem Pfad zu diesem. Wird er fündig, kopiert er den vollständigen Pfad in den Paketkopf und sendet es an den nächsten Knoten im Pfad. Sollte der Quellknoten keinen Pfad in seiner Routingtabelle haben, tritt er in die Phase der Pfadsuche ein.

#### Pfadsuche

Der Quellknoten sendet in dieser Phase ein *Route-Request*-Paket in das Netz aus. Das Route-Request-Paket wird im Netz geflutet, sodass alle Knoten es empfangen. Abbildung 5.3 stellt die Pfadsuche von DSR grafisch dar.

Um das Netz nicht unnötig zu belasten, besteht das Route-Request-Paket nur aus einer Sequenznummer, der Adresse des Zielknotens und einer *Pfadliste*. Die Pfadliste besteht aus den Adressen der besuchten Knoten. Am Anfang enthält sie lediglich die Adresse des Quellknotens. Jeder Knoten, der das

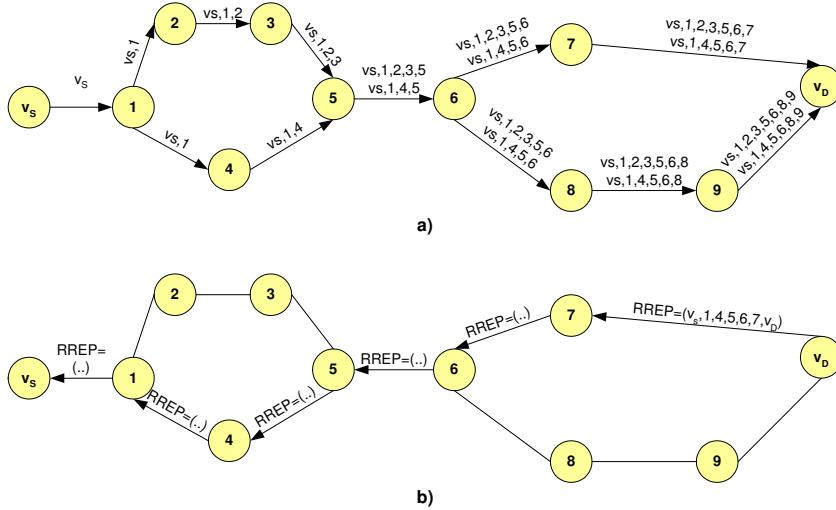


Abbildung 5.3: Die Pfadsuche von DSR. Abbildung a) zeigt den ersten Schritt. Der Quellknoten  $v_s$  schickt ein Route-Request-Paket, das sich auf dem Weg zum Zielknoten die besuchten Knoten merkt. Abbildung b) zeigt wie der Zielknoten  $v_d$  dem Quellknoten mit einem Route-Reply-Paket antwortet. Wie im Beispiel zu sehen ist, kann es passieren, dass der Zielknoten das Route-Request-Paket auf mehreren Wegen mit unterschiedlichen Pfadlängen erhält. Im Beispiel wählt der Zielknoten den kürzesten Pfad aus, um dem Quellknoten zu antworten.

Route-Request-Paket empfängt und weiterleitet, fügt seine eigene Adresse an die Pfadliste an. Auf diese Weise erreicht das Route-Request-Paket den Zielknoten.

Der Zielknoten extrahiert aus dem empfangenen Route-Request-Paket die Pfadliste, und fügt seine eigene Adresse an sie an. Daraufhin erzeugt der Zielknoten ein so genanntes *Route-Reply*-Paket und schickt es an den Quellknoten. Das Route-Reply-Paket wird auf dem umgekehrten Pfad, welcher aus der Pfadliste entnommen wurde, übertragen.

Nachdem der Quellknoten das Route-Reply-Paket vom Zielknoten erhalten hat, kann die eigentliche Kommunikation beginnen. Der Quellknoten versieht die Datenpakete mit dem erhaltenen Pfad. Somit können diese vom Quellknoten zum Zielknoten und umgekehrt übertragen werden.

Solange keine Probleme bei der Übertragung von Datenpaketen auftreten, verweilt der Algorithmus in diesem Zustand. Sollte bei der Weiterleitung zwischen zwei Knoten ein Fehler auftreten, z.B. wegen der Knotenmobilität, tritt DSR in seine zweite Phase ein.

## Pfadpflege

Die Pfadpflege dient der Beseitigung von Pfadfehlern auf dem aktuellen Pfad zwischen Quellknoten und Zielknoten. Erkennt ein Knoten einen Verbindungsabbruch zu seinem Nachbarn, so benachrichtigt er den Quellknoten über den Fehler mit einem *Route-Error*-Paket. Nach dem Empfang des Route-Error-Pakets hält der Quellknoten die Datenübertragung an und initiiert eine neue Pfadsuche, um einen intakten Pfad zum Zielknoten zu finden. Dies geschieht wie im vorherigen Abschnitt beschrieben.

Die Pfadpflege wird durch weitere Maßnahmen verbessert. Wenn der Quellknoten mehrere Pfade zum Zielknoten kennt, kann er ohne Verzögerung die Datenübertragung über eine der ihm bekannten Pfade weiterführen. Eine weitere Maßnahme ist die so genannte Alterung von Pfaden. Dabei merkt sich ein Knoten den Zeitpunkt der letzten Benutzung eines Pfads. In regelmäßigen Intervallen werden alle Pfade auf den Zeitpunkt ihres Einsatzes überprüft und die Pfade, die längere Zeit nicht benutzt wurden, werden gelöscht. Dadurch soll verhindert werden, dass durch veraltete Pfadinformationen die Datenübertragung zusätzlich verzögert wird, da ein Knoten diese als intakt annehmen und für die Datenübertragung verwenden könnte.

### 5.1.6 Andere Routingalgorithmen für MANETs

Eine ausführliche Diskussion der unterschiedlichsten MANET Routingalgorithmen würde den Rahmen dieser Arbeit sprengen. Deshalb sollen in diesem Abschnitt drei interessante Routingalgorithmen kurz vorgestellt werden, die sich durch ihr Konzept von anderen unterscheiden.

#### Fisheye State Routing

Das *Fisheye State Routing (FSR)* [PGC00] ist ein proaktives Verfahren und basiert auf dem Link-State-Routing. Beim Link-State-Routing besitzt jeder Knoten im Netz eine globale Übersicht des Netzes. Hierzu ermittelt jeder Knoten seine direkten Nachbarn und sammelt Informationen über diese. Danach werden die Informationen im *gesamten* Netz mit allen anderen Knoten ausgetauscht. Der Austausch der Informationen kann regelmäßig oder in Abhängigkeit von Ereignissen stattfinden. Auf dieser Grundlage kann jeder Knoten für sich die kürzesten Pfade im Netzwerk berechnen. Für die Verteilung der gesammelten Information wird Flooding verwendet.

Das FSR versucht die hohe Last des Link-State-Routings, die durch das Austauschen der Routinginformationen verursacht wird, zu reduzieren. Hierzu werden die Knoten in Zonen aufgeteilt. Jeden Knoten umgeben zentrisch

von ihm ausgehend mehrere Zonen (siehe Abbildung 5.4). Die Austauschfrequenz der Routinginformationen nimmt mit zunehmender Distanz vom Knoten ab, d.h. ein Knoten tauscht mit den Knoten in der zentralen Zone öfter Routinginformationen aus, als mit Knoten in entfernten Zonen. Hierdurch erklärt sich auch der Name, die Sicht – die Aktualität der Routinginformationen – eines Knotens auf das Netzwerk nimmt mit der Distanz ab, wie bei einem Fischauge. FSR benutzt das gleiche Verfahren wie DSDV, um das Austauschen von Routinginformationen effizienter zu gestalten.

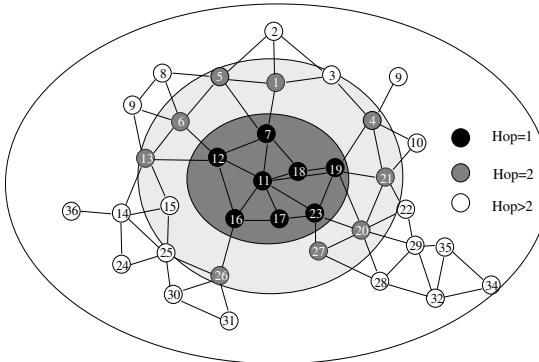


Abbildung 5.4: Zonenaufteilung beim Fisheye State Routing [PGC00]. Dargestellt ist die Sicht auf das Netz ausgehend von Knoten 11 mit drei Zonen und Abständen von 1-, 2- und >2-Hops.

## Optimized Link State Routing Protocol

Das *Optimized Link State Routing Protocol (OLSR)* [CJL<sup>+</sup>01, ACJ<sup>+</sup>03] basiert auf dem Link-State-Routing mit einigen Optimierungen für mobile Ad-hoc-Netze. OLSR versucht die Last, die durch die Verteilung der Routinginformationen entsteht, zu reduzieren. Hierzu wird das Konzept des *Multipoint-Relays* benutzt (siehe Abbildung 5.5). Dabei wird eine Teilmenge der Knoten im Netzwerk ausgezeichnet, die für das Flooding von Broadcastnachrichten zuständig ist. Da nun nicht mehr alle Knoten am Flooding beteiligt sind, reduziert sich der Aufwand hierfür. Der problematische Teil des Ansatzes liegt in der effizienten Bestimmung der Multipoint-Relay-Menge.

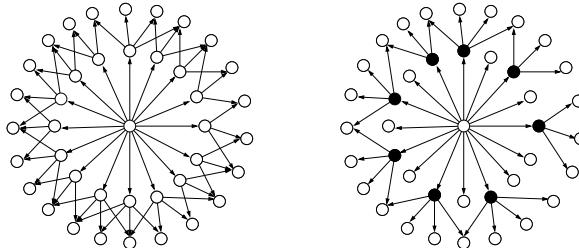


Abbildung 5.5: In OLSR eingesetzter Multipoint-Relay [ACJ<sup>+</sup>03]. Links: Einfaches Fluten des Netzes vom mittlerem Knoten aus. Rechts: Fluten des Netzes mit Hilfe der Multipoint-Relays.

### Zone Routing Protocol

Das *Zone Routing Protocol* (ZRP) [HP99] ist ein hybrides Routingprotokoll, dass auf der Unterteilung eines Ad-hoc-Netzwerkes in mehrere so genannte Cluster basiert. Die Größe eines Clusters kann derart gewählt werden, dass innerhalb eines Clusters auch multi-hop Kommunikation vorkommt. Innerhalb eines Clusters wird ein anderer Routingalgorithmus eingesetzt als für die Zwischen-Cluster-Kommunikation. Hierdurch wird versucht, die Vorteile von proaktiven und reaktiven Routingalgorithmen auszunutzen, da man davon ausgeht, dass der Verkehr innerhalb eines Clusters überwiegt. Der Einsatz von proaktiven Routingalgorithmen innerhalb eines Clusters und reaktiven Routingalgorithmen für Zwischen-Cluster-Kommunikation ist sinnvoll. Das ZRP besteht deshalb aus drei Subprotokollen. Das proaktive *Intrazone Routing Protocol* für den Einsatz in einem Cluster, das reaktive *Interzone Routing Protocol* für die Kommunikation zwischen unterschiedlichen Clustern und das *Bordercast Resolution Protocol*.

## 5.2 Schwarmintelligenz

Unter dem Begriff *Schwarmintelligenz* versteht man Algorithmen und Lösungsansätze, die durch das Kollektivverhalten von Insekten und anderen Tieren inspiriert sind [BDT99]. Einfache Individuen wie Termiten, Bienen, Ameisen und andere Insekten, die in einer Kolonie leben und an sich mit sehr beschränkten Fähigkeiten ausgestattet sind, sind in der Lage hoch komplexe Aufgaben effizient zu lösen. Die Besonderheit an diesem Phänomen ist, dass die Tiere für die Kooperation keine zentrale Steuerung besitzen. Zur Lösung eines Problems organisieren sich die Tiere ohne die Hilfe einer Kontrollin-

stanz. Diese Selbstorganisation ist faszinierend, da sie von der Natur kopiert und für die Lösung von mathematisch technischen Problemen verwendet werden kann.

### 5.2.1 Insektenschwärme in der Natur

Die Zusammenarbeit und die Problemlösungsfähigkeit von Insektenschwärmern wird an einigen Beispielen diskutiert [LL68, Grö85]. Diese Beispiele zeigen, welche unterschiedlichen Aufgaben Insektenschwärme lösen müssen, und wie die Kooperation unter den einzelnen Mitgliedern eines Schwarms im Einzelnen aussieht.

Die Nestpflege und der Nestbau stellen für jeden Insektenschwarm große Aufgaben dar, die immer wieder angegangen werden müssen. Die Aufgaben werden sehr unterschiedlich gelöst. Interessant ist, dass der Nestbau und die Nestpflege durch parallele Ausführung von einzelnen Aufgaben beschleunigt wird. Die Aufteilung der Arbeit hängt davon ab, ob der Insektenstaat aus einem oder mehreren Insektentypen besteht. Bei Insektenstaaten, die aus unterschiedlichen Typen von Tieren bestehen, ist die Aufgabenteilung durch physische Gegebenheiten wie Größe vorgegeben. Ansonsten gibt es entweder spezialisierte Tiere oder jede anfallende Aufgabe kann von jedem Tier durchgeführt werden. Eine interessante Beobachtung ist, dass die Aufteilung der Arbeit nicht statisch ist, sondern dynamisch von der aktuellen Situation abhängt. Fallen die Arbeiter einer Aufgabe aus, so übernehmen andere Arbeiter auch deren Arbeit, d.h. das Ausfallen von einzelnen Arbeitern oder einer ganzen Gruppe wird durch andere Tiere kompensiert. Dies gilt sogar dann, wenn der Insektenstaat aus mehreren Insektentypen besteht. Wenn eine Klasse von Tieren ausfällt, übernimmt eine andere Klasse auch deren Arbeit.

Neben der Nestpflege ist die Futtersuche eine der wichtigsten Aufgaben einer Insektenkolonie. Einige Ameisenarten jagen mit bis zu 200.000 Tieren gleichzeitig. Dabei wird eine Fläche von mehreren hundert Metern durchforstet. Die Aufteilung und organisierte Jagd stellt eine große Aufgabe dar. Bei der Futtersuche kann der Weg durch Gegenstände oder eine größere Spalte, z.B. eine Wasserrinne, blockiert sein. In diesem Fall muss das Hindernis weggeräumt, umgangen oder überbrückt werden. Eine bestimmte Ameisenart ist in der Lage eine Brücke zu bauen, die aus einzelnen Ameisen besteht, um solche Hindernisse zu überqueren. Die Organisation eines solchen Baus muss organisiert und später wieder aufgelöst werden. Sollten mehrere Futterquellen gleichzeitig gefunden werden, stellt sich die Frage der effizienten Ausbeutung der Futterquellen. Bei Bienen und einigen Ameisenarten wurde beobachtet, dass die Ausbeutung von Futterquellen sehr raffiniert ist und in Abhängigkeit der Distanz zum Nest und der Futterqualität durchgeführt wird.

Eine andere alltägliche Arbeit ist die Fütterung der Brut, also des Nachwuchses. Diese Aufgabe umfasst den Transport von Futter in die Tiefen des Nestes, die Fütterung des Nachwuchses und die Reinigung des Nestes. Insekten sind nicht nur Jäger, sondern auch Gejagte. Deshalb besitzen Insektenschwärme ein hoch ausgereiftes Verteidigungssystem, wobei natürlich ein Insekt alleine wenig gegen einen Angreifer anrichten kann. Die Information über einen Angriff wird umgehend in der Kolonie verbreitet, wodurch eine Vielzahl von Verteidigern aktiv werden.

Diese Beispiele zeigen wie Insektenschwärme selbständig in der Lage sind, komplexe Aufgaben durch Kooperation zu lösen und ohne das Vorhandensein einer zentralen Steuerung und Verwaltung zu organisieren.

### 5.2.2 Selbstorganisation

Die *Selbstorganisation* ist eine globale Erscheinung, die durch die Interaktion von Individuen auf unterer Ebene hervorgerufen wird [CDR98]. Die Interaktion der Komponenten basiert dabei nur auf lokalen Informationen, ohne die Einbeziehung von globalen Zielen. Beispielsweise bedeutet dies im Falle der Futtersuche einer Ameisenkolonie, dass eine einzelne Ameise kein Wissen über die zu lösende Aufgabe besitzt. Die Selbstorganisation ist durch vier grundlegende Funktionen bestimmt [BDT99]:

- **Positive Rückkopplung**

Die Selbstorganisation wird essenziell durch positive Rückkopplung beeinflusst. Sie bewirkt die Ergreifung von neuen Aktionen und die Verstärkung von existierenden Aktionen.

- **Negative Rückkopplung**

Die negative Rückkopplung ist die Gegenmaßnahme zur positiven Rückkopplung, um das gesamte System zu stabilisieren.

- **Fluktuation des Verhaltens**

Zufällige Ereignisse sichern die Entdeckung und die Einbeziehung unbekannter Wege in die Problemlösung. Dadurch wird ein mögliches starres Verhalten verhindert.

- **Interaktion zwischen den Individuen**

Die Selbstorganisation erfordert die Interaktion der Individuen; obwohl es möglich ist, dass ein einzelnes Individuum die Selbstorganisation durchführt, ist für eine effiziente Problemlösung die Zusammenarbeit

der Einzelnen gefragt. Auch wenn unterschiedliche Individuen verschiedene Signaturen besitzen, sollten sie in der Lage sein, die Signatur der zur Gemeinschaft gehörigen Individuen zu erkennen und zu nutzen.

Selbstorganisierende Systeme zeichnen sich durch bestimmte Eigenschaften aus:

- Aufbau einer temporären Struktur in einem ursprünglich inhomogenen Medium. Die Strukturen sind beispielsweise durch Netze gegeben.
- Koexistenz mehrerer stabiler Zustände. Da die Strukturen durch den Zufall geprägt sind, kann es dazu kommen, dass gleichzeitig mehrere unterschiedliche Lösungen verstärkt werden und in unterschiedliche Zustände führen.
- Existenz von Variationen, wenn einige Parameter geändert werden. Die Veränderung einiger Parameter kann dazu führen, dass nun andere Wege bevorzugt werden. Bei den Ameisenalgorithmen kann dies zu einer Veränderung der Pheromonkonzentrationsverteilung führen. Dadurch werden andere Pfade bevorzugt.

### 5.2.3 Kommunikation in einem Schwarm

Bisher wurde beschrieben wie ein Schwarm durch die Kooperation von einfachen Individuen wie Ameisen komplexe Aufgaben lösen kann. Jedoch wurde die Frage, wie die Individuen in einem Schwarm untereinander kommunizieren, offen gelassen. Die Mitglieder eines Schwarms, z.B. einer Ameisenkolonie, besitzen unterschiedliche Kommunikationsfähigkeiten. Sie können durch Sicht-, Tast-, Gehör-, Geschmack- und Riechkontakt kommunizieren [LL68, Grö85]. Im Allgemeinen kann die Kommunikation in einem Insektenschwarm in zwei Klassen unterschieden werden:

- **Direkte Kommunikation**

Die erste Klasse stellt die direkte Kommunikation dar. Dabei befinden sich mehrere Tiere in direkter Reichweite. Die Kommunikation erfolgt durch Berührung, den Austausch von Flüssigkeiten oder einfach durch Sichtkontakt.

- **Indirekte Kommunikation**

Die zweite Klasse ist durch die indirekte Kommunikation gegeben. Hierbei findet die Kommunikation durch die Modifikation der Umgebung statt, die auch *Stigmergy* genannt wird. Man unterscheidet zwei Arten

von Stigmergy [SHBR96, Whi97b]. Die erste Art wird als *sematectonic Stigmergy* bezeichnet, das die physische Veränderung der Umgebung umfasst. Ein Beispiel hierfür ist der Bau eines Nestes.

Die zweite Art von Stigmergy wird als *zeichenbasierte Stigmergy* bezeichnet. In diesem Fall wird die Umgebung nicht direkt verändert, sondern ein Zeichen wird abgelegt, welches nach einer Zeit wieder verschwinden kann. Bei den Ameisenkolonien ist die zeichenbasierte Stigmergy sehr stark ausgeprägt. Sie verwenden als Zeichen chemische Botenstoffe – Pheromone. Die Ameisen sind zum einen in der Lage die Zeichen wahrzunehmen und zum anderen die Stärke der Zeichen zu unterscheiden. Dadurch können die Ameisen Entscheidungen fällen und neue Aktionen starten.

## 5.3 Ameisenalgorithmen

Ameisenalgorithmen sind ein Teilbereich der Schwarmintelligenz. Sie sind Multi-Agentensysteme, die auf dem Verhalten von einzelnen Ameisen basieren [DD99, BDT99]. Dabei stellt jede Ameise einen Agenten dar, der unabhängig von den anderen agiert. Der Begriff Ameisenalgorithmus (*Ant Algorithm*) wurde zum ersten mal von Dorigo et al. in [DMC91b, DMC91a] eingeführt.

Im Folgenden wird die Idee und die formale Beschreibung des Ameisenalgorithmus vorgestellt, der zuerst als heuristische Lösung für das Problem des Handelsreisenden vorgeschlagen wurde [DMC91b, DMC91a]. Später wurde der Ameisenalgorithmus für die Lösung von anderen Optimierungsproblemen angepasst, wie das Quadratic Assignment Problem (QAP), Job-Scheduling und Graphfärbung. Eine Übersicht gibt [BDT99].

### 5.3.1 Verhalten von Ameisen bei der Futtersuche

Die Idee, die den Ameisenalgorithmen zu Grunde liegt, ist die formale Umsetzung des Verhaltens von Ameisenkolonien bei der Futtersuche [CDR98]. Suchen Ameisen nach Nahrung, starten sie von ihrem Nest und laufen in Richtung der Futterquelle. Erreicht eine Ameise eine Weggabelung, muss sie sich für eine Abzweigung entscheiden. Ameisen scheiden während ihrer Suche einen Botenstoff aus, das so genannte Pheromon, das den verwendeten Weg kennzeichnet. Die Pheromonkonzentration auf einer bestimmten Strecke ist daher ein Indiz über ihre Nutzungshäufigkeit. Die Ameisen nutzen die Pheromonkonzentration aus diesem Grund als Entscheidungskriterium bei der Wegwahl an einer Gabelung, d.h. sie benutzen die zeichenbasierte Stigmergy für

die Kommunikation. Die Pheromonkonzentration auf einer Strecke ist nicht konstant. Sie wird erhöht, wenn Ameisen die Strecke benutzen, und sie wird durch Verflüchtigungseffekte mit der Zeit verringert [Grö85]. Diese Eigenschaft macht sie für den Einsatz in mobilen Ad-hoc-Netzen sehr attraktiv, da hierdurch die Dynamik von Ad-hoc-Netzen modelliert werden kann.

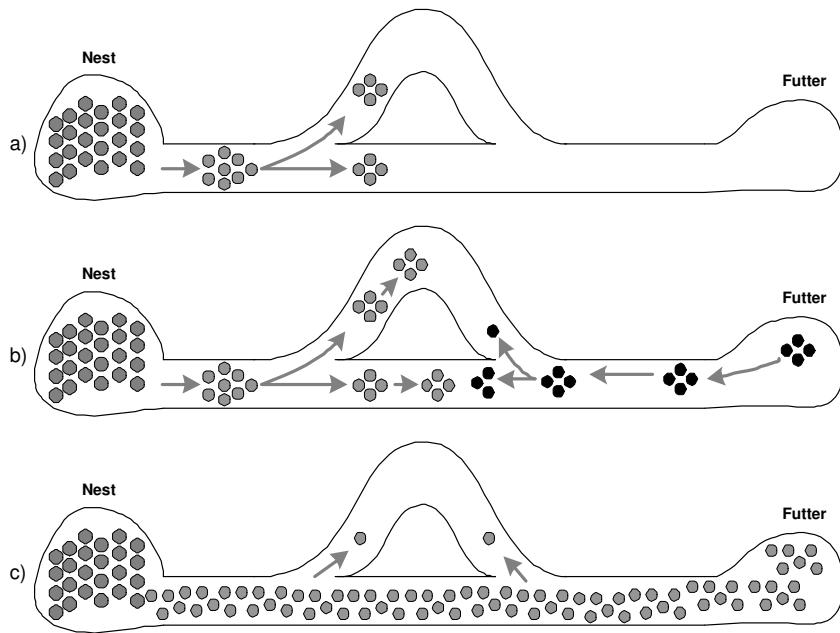


Abbildung 5.6: Versuchsaufbau, mit dem die Funktionsweise von Ameisenalgorithmen im Labor entdeckt wurde. a) Am Anfang befinden sich keine Pheromone auf den Pfaden, deshalb entscheiden sich die ersten Ameisen rein zufällig. b) Nach einer Weile unterscheiden sich die Pheromonwerte auf den beiden Pfaden, deshalb benutzen immer mehr Ameisen den kürzeren Pfad. Dadurch wird die Pheromonkonzentration auf diesem Pfad immer stärker. c) Schließlich nach einer Einschwingphase, benutzen fast alle Ameisen den kurzen Weg zwischen Nest und Futterplatz.

Abbildung 5.6 stellt ein Szenario mit zwei Pfaden vom Ameisennest zum Futterplatz dar. An den Weggabelungen müssen sich die Ameisen entscheiden, welche Strecke sie als nächstes verwenden. Die ersten Ameisen entscheiden sich zufällig, da noch keine Pheromonkonzentrationen auf den Strecken existieren (siehe Abbildung 5.6 a)). Die Ameisen, die den unteren Weg eingeschlagen haben, kommen schneller an der Futterquelle an, als die anderen Ameisen. Auf ihrem Rückweg können sich diese Ameisen schon an den Pheromonkonzentrationen orientieren, jedoch ist zu dieser Zeit die Pheromonkonzentration auf beiden Abzweigungen sehr ähnlich. Mit der Zeit unter-

scheidet sich die Pheromonkonzentration auf den möglichen Pfaden immer stärker (siehe Abbildung 5.6 b)). Nach kurzer Zeit, der Einschwingphase, ist der kürzeste Weg vom Nest zur Futterquelle eindeutig identifiziert und fast alle Ameisen verwenden diesen, um zur Futterquelle zu gelangen (siehe Abbildung 5.6 c)). Obwohl sich die meisten Ameisen an diese Vorgehensweise halten, gehen einige Ameisen manchmal andere Wege. Aus Abbildung 5.6 c) ist dies gut zu erkennen. Sowohl von den Ameisen, die vom Nest in Richtung Futterplatz gehen, als auch von den anderen, entscheiden sich einige Ameisen ab und zu für den längeren Weg.

Was passiert, wenn auf dem Weg zum Futterplatz eine Sackgasse ohne Futter existiert, und die Distanz zur Sackgasse erheblich kürzer ist als zum Futterplatz (siehe Abbildung 5.7)? Man könnte annehmen, dass die Ameisen durch diese zufällige Bewegung eigentlich den Weg zur Sackgasse stärken müssten. In der Natur ist es so, dass die Ameisen, die mit Futter zurückkehren viel mehr Pheromone auf den verwendeten Wegen ablegen, als Ameisen die ohne Futter zurückkehren. Daher probieren zwar ab und zu auch einige Ameisen die *wertlosen* Pfade, jedoch werden diese nicht so positiv verstärkt wie die eigentlichen wertvollen Pfade, die zu Futter, also Erfolg führen [Grö85].

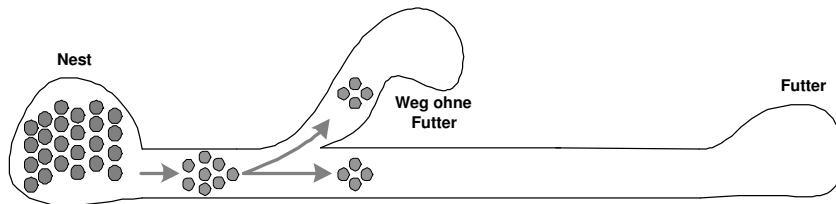


Abbildung 5.7: Futtersuchverhalten von Ameisen bei mehreren Wegen mit und ohne Futter.

Dieses Verhalten der Ameisen kann genutzt werden, um in Netzen den kürzesten Pfad zwischen zwei Knoten zu finden. Insbesondere die dynamische Komponente, die durch die Pheromonkonzentration ausgedrückt wird, bietet eine hohe Adaption des Verfahrens für dynamische mobile Ad-hoc-Netze an, da sich in diesen Netzen Verbindungen zwischen Knoten sehr oft ändern.

## Selbstorganisation

Die Anforderungen der Selbstorganisation ist bei Ameisenalgorithmen wie folgt erfüllt:

- **Positive Rückkopplung**

Die positive Rückkopplung ist in den Ameisenalgorithmen durch die Erhöhung der Pheromonkonzentration auf den verwendeten Pfaden gegeben.

- **Negative Rückkopplung**

Bei den Ameisenalgorithmen ist die negative Rückkopplung durch die Verdunstung der Pheromonkonzentration gegeben. Zum einen wird dadurch eine zu schnelle Erhöhung der Konzentration erreicht und zum anderen das Abklingen der Pheromonkonzentration sichergestellt.

- **Fluktuation des Verhaltens**

In den Ameisenalgorithmen wird durch zufälliges Probieren von Wegen die Auffindung unbekannter Futterquellen oder kürzerer Pfade ermöglicht.

- **Interaktion zwischen den Individuen**

Jede Ameise trifft ihre Entscheidungen selbstständig, sie lässt sich jedoch dabei von den Pheromonen leiten. Die Ameisen interagieren über die Pheromone miteinander und beeinflussen sich indirekt.

### 5.3.2 Ant System

Das *Ant System* (AS) wurde von Dorigo et al. in [DMC91b, DMC91a] eingeführt und für die heuristische Lösung des Problems des Handelsreisenden (Traveling Salesman Problem, TSP) benutzt. Beim TSP muss eine Route durch  $n$  Städte mit minimaler Länge bestimmt werden, wobei jede Stadt nur einmal besucht werden darf. Das Problem kann als Graph mit  $n$  Knoten dargestellt werden. Dabei wird eine Straße zwischen zwei Städten durch eine Kante zwischen den entsprechenden Knoten repräsentiert.

Das Ant System findet eine Lösung für TSP dadurch, dass sich  $m$  Ameisen auf dem Graphen bewegen, bis sie eine Route bestimmt haben. Die Autoren schlagen vor  $m = n$  zu setzen. Die Ameisen besitzen einen Speicher, der die noch nicht besuchten Knoten angibt. Bei ihren Bewegungen auf dem Graphen hinterlassen die Ameisen Pheromonspuren. Dabei bewegen sich die Ameisen probabilistisch von einem Knoten zum anderen. Bei der Wahl des nächsten Knotens lässt sich eine Ameise von den vorhandenen Pheromonen leiten.

Jede Ameise führt  $t_{\max}$  Runden durch. In Runde  $t \leq t_{\max}$  entscheidet gemäß Formel 5.1 Ameise  $k$  auf dem Knoten  $v_i$  Knoten  $v_j$  als nächstes zu besuchen

mit der Wahrscheinlichkeit  $p_{i,j}^k$ .

$$p_{i,j}^k(t) = \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}(t)]^\beta}{\sum_{l \in J_i^k} [\tau_{i,l}(t)]^\alpha \cdot [\eta_{i,l}(t)]^\beta} \quad (5.1)$$

In Formel 5.1 bezeichnet  $t$  die aktuelle Runde,  $\tau_{i,j}(t)$  den Pheromonwert der Kante  $e(i, j)$ ,  $\eta_{i,j}(t) = \frac{1}{d_{i,j}}$  bezeichnet die Sichtbarkeit einer Kante und ist durch die Distanz  $d_{i,j}$  der Knoten  $v_i$  und Knoten  $v_j$  gegeben.  $\alpha$  und  $\beta$  sind Parameter, die frei gewählt werden und die Gewichtung des Pheromonwertes und der Sichtbarkeit beeinflussen.

Kanten, die von Ameisen benutzt wurden, werden positiv verstärkt. Nach jeder Runde legt eine Ameise auf den Kanten  $e(i, j)$ , die sie benutzt hat, zusätzliche Pheromone ab. Weiterhin wirkt die negative Verstärkung auf alle Kanten, um eine Stagnation des Verfahrens zu verhindern. Daher ändern sich die Pheromonwerte  $\tau_{i,j}^k$  der Kanten gemäß der Gleichung 5.2.

$$\tau_{i,j}^k(t) := \begin{cases} (1 - \rho) \cdot \tau_{i,j}(t) + \Delta\tau_{i,j}^k(t) & \text{wenn } e(i, j) \in T^k(t) \\ (1 - \rho) \cdot \tau_{i,j}(t) & \text{wenn } e(i, j) \notin T^k(t) \end{cases} \quad (5.2)$$

In der Gleichung 5.2 bezeichnet  $\rho$ ,  $0 \leq \rho < 1$  den Verflüchtigungsfaktor,  $\Delta\tau_{i,j}^k(t)$  die Menge an Pheromon, die Ameise  $k$  auf der Kante ablegt, wenn die Kante  $e(i, j)$  auf ihrer Route  $T^k(t)$  in Runde  $t$  war.

### 5.3.3 Ant Colony System

Die Ergebnisse vom Ant System waren für TSP-Probleme mit kleinem  $n$  vergleichbar mit bekannten Algorithmen, jedoch fiel die Leistung mit wachsender Anzahl von Knoten ab. Dorigo et al. verbesserten das Verfahren und nannten es in [DG97a, DG97b] *Ant Colony System (ACS)*. Das ACS besitzt einige Modifikationen gegenüber AS, die die Wahl des nächsten Knotens, die Veränderung der Pheromonwerte und die Benutzung einer Kandidatenliste betreffen.

- **Wahl des nächsten Knoten**

Eine Ameise wählt als nächsten Knoten  $v_j$  nach der Regel in Formel 5.3.

$$j = \begin{cases} \max_{u \in J_i^k} \left\{ [\tau_{i,u}(t)] \cdot [\eta_{i,u}]^\beta \right\} & \text{wenn } q \leq q_0 \\ s & \text{wenn } q > q_0 \end{cases} \quad (5.3)$$

In Formel 5.3 bezeichnet  $q$  eine gleichverteilte Zufallsvariable über  $[0, 1]$ ,  $q_0$ ,  $0 \leq q_0 \leq 1$  ist ein wählbarer Parameter und  $J_i^k$  gibt die Menge

der Knoten an, die die Ameise  $k$ , die sich im Knoten  $i$  befindet, noch nicht besucht hat. Schließlich ist  $s$ ,  $s \in J_i^k$  ein zufällig gewählter, noch nicht besuchter Knoten. Die Wahl von  $s$  erfolgt mit der Wahrscheinlichkeit  $p_{i,s}^k$  gemäß Formel 5.4.

$$p_{i,s}^k = \frac{[\tau_{i,s}(t)] \cdot [\eta_{i,s}]^\beta}{\sum_{l \in J_i^k} [\tau_{i,l}(t)] \cdot [\eta_{i,l}]^\beta} \quad (5.4)$$

Eine Ameise entscheidet sich entweder zufällig für einen der Knoten, den sie noch nicht besucht hat ( $q > q_0$ ), oder für den Knoten mit dem maximalen Pheromonwert ( $q \leq q_0$ ).

- **Pheromonveränderung**

Bei AS waren alle Ameisen an der Veränderung der Pheromone nach jeder Runde beteiligt. Bei ACS darf nur noch die Ameise, die den besten Weg in einer Runde benutzt hat, die Pheromonwerte verändern, d.h. zusätzliche Pheromone auf den Kanten ablegen. Dadurch werden die besten Lösungen schneller verstärkt.

Diese Vorgehensweise könnte das Auffinden von anderen Lösungen verhindern. Um diesem entgegenzusteuern, wurde auch eine so genannte lokale Pheromonverstärkung integriert. Dabei verändern die Ameisen während ihrer Bewegung die Pheromonkonzentration auf den besuchten Knoten.

- **Kandidatenliste**

Die Kandidatenliste gibt für jeden Knoten eine Menge von Nachbar-knoten an, die bei der Wahl des nächsten Knotens bevorzugt zu wählen sind. Dadurch wird die Entscheidung der Ameisen stark beeinflusst.

### 5.3.4 Max-Min AS

Stützle und Hoos schlagen eine Erweiterung zum Ant System vor, mit der die Leistung des Algorithmus weiter verbessert werden konnte [SH97b, SH97a]. Diese Erweiterung nannten sie *Max-Min AS (MMAS)*. Die Veränderungen gegenüber AS sind wie folgt: i) ähnlich zu ACS darf auch bei MMAS nur die Ameise mit dem besten Ergebnis der aktuellen Runde Pheromonwerte verändern, ii) Pheromonwerte sind nach unten und oben durch ein Intervall  $[\tau_{\min}, \tau_{\max}]$  beschränkt und iii) alle Kanten werden am Anfang mit  $\tau_{\max}$  initialisiert [BDT99].

MMAS ist in seiner Leistung mit ACS vergleichbar. Interessant ist vor allem die Beschränkung der Pheromonwerte. Dadurch wird zum einen verhindert,

dass der Pheromonwert einer Route extrem steigt und das Auffinden anderer Routen unmöglich wird, und zum anderen, dass weniger optimale Routen völlig verschwinden.

### 5.3.5 AS-Rank

Eine weitere Erweiterung zum Ant System wurde von Bullnheimer et al. vorgeschlagen, die *AS-rank* genannt wird [BHS97]. Dabei werden die Ameisen nach jeder Runde nach ihrem Routen-Ergebnis sortiert und die  $m$  besten Ameisen dürfen die Pheromonwerte entsprechend ihrem Rang verändern.

### 5.3.6 Adaption an andere Fragestellungen

Ameisenalgorithmen eignen sich für die Lösung von klassischen Optimierungsproblemen. Durch die Kooperation über eine globale Informationsstruktur, die durch die Pheromonwerte gegeben ist, sind die Ameisen in der Lage gute Ergebnisse zu liefern. Ameisenalgorithmen lassen sich auf eine Vielzahl von Problemen anwenden, sofern folgende Voraussetzungen erfüllt sind [BDT99].

- Repräsentation des Problems als eine Struktur, sodass Ameisen auf ihr Lösungen bilden können, z.B. bietet sich die Darstellung als ein Graph an.
- Umsetzung des autokatalytischen Prozesses, z.B. durch positive und negative Verstärkung von Pheromonen.
- Eine Heuristik, die die Bildung von lokalen Lösungen erlaubt.
- Regeln für das Aktualisieren von Pheromonwerten und die Wahl des nächsten zu besuchenden Knotens, d.h. eine Transitionsregel.

Die Ameisenalgorithmen besitzen jedoch auch einige kritische Punkte, die bei der Anpassung von Ameisenalgorithmen für andere Probleme beachtet werden müssen [BDT99]:

- **Blockierung**

Die Blockierung tritt auf, wenn ein gefundener Weg nicht mehr vorhanden ist. Dies hat zur Folge, dass die Ameisen eine gewisse Zeit brauchen, um einen neuen Weg zu finden.

- **Stagnation**

Dieses Problem tritt auf, wenn ein neuer kurzer Weg entsteht, der nicht existierte. Da die Ameisen die vorhandenen Wege schon intensiv durchforstet haben, kann es lange dauern, bis der neue kürzere Weg gefunden wird.

- **Konvergenz**

Ameisenalgorithmen konvergieren im Allgemeinen nicht zu einem einzelnen Ergebnis, sondern sie bilden mehrere Ergebnisse, die optimal oder auch suboptimal sein können (siehe Abbildung 5.8). Diese Eigenschaft kann sich sowohl als Schwäche aber auch als Stärke erweisen. Für den Einsatz in mobilen multi-hop Ad-hoc-Netzen zeigt sie sich als Stärke. Wenn durch einen Verbindungsabbruch der beste Weg zusammenbricht, sind Ameisen noch in der Lage, die anderen Wege zu benutzen. Insbesondere können auf der Basis der anderen Wege neue optimale Wege gefunden werden.

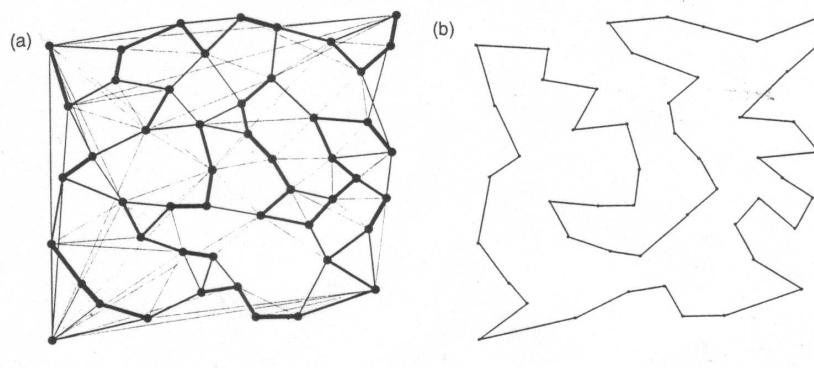


Abbildung 5.8: Ergebnis von ACS für ein TSP-Problem [BDT99]. (a) Verteilung der Pheromonwerte auf den einzelnen Kanten des Graphen. Je dicker eine Linie ist, desto höher ist die Pheromonkonzentration auf ihr. (b) Beste Lösung von ACS für dieses TSP-Problem.

- **Speicher**

Bei den Ameisenalgorithmen mit den besseren Ergebnissen besitzen die künstlichen Ameisen einen Speicher, in dem wichtige Informationen gespeichert werden, die die Entscheidungen der Ameisen sehr stark beeinflussen können. Reale Ameisen besitzen keinen solchen Speicher, deshalb sind die künstlichen Ameisen nicht mehr mit den natürlichen

Ameisen vergleichbar. Bei der Anpassung der Heuristik kann sich dieser Punkt als besonders schwierig erweisen, insbesondere wenn die Umgebung dynamisch ist und keine vollständige Information über sie existiert, wie es in mobilen multi-hop Ad-hoc-Netzen der Fall ist.

Zur Lösung der ersten beiden Probleme haben verschiedene Forscher unterschiedliche Ansätze vorgeschlagen. Generell basieren die Lösungen auf der Integration von Rauschen in das Verhalten der Ameisen [SHBR96]. Dies wird beispielsweise dadurch erreicht, dass in unregelmäßigen Abständen Ameisen in bisher nicht erkundete Bereiche geschickt werden. Ein weiterer Ansatz besteht in der Re-Initialisierung der Pheromonkonzentrationen durch bestimmte Ereignisse.

## 5.4 Der Ameisenroutingalgorithmus

In diesem Abschnitt wird die Umsetzung des Ameisenalgorithmus für mobile multi-hop Ad-hoc-Netze diskutiert und der Ameisenroutingalgorithmus (ARA) vorgestellt [GSB02, GS02, GKB03].

### 5.4.1 Adaption des Ameisenalgorithmus

Der Ameisenalgorithmus kann zum Auffinden von kürzesten Wegen in einem verbundenen Graphen eingesetzt werden. Sei  $G = (V, E)$  ein verbundener Graph mit  $n = |V|$  Knoten. Das Ziel ist das Finden des kürzesten Pfades zwischen einem Quellknoten  $v_s$  und einem Zielknoten  $v_d$  auf dem Graphen  $G$ . Die Pfadlänge ist durch die Anzahl der Knoten im Pfad gegeben.

Jeder Kante  $e(i, j) \in E$  des Graphen zwischen dem Knoten  $v_i$  und  $v_j$  sind Variablen  $\phi_{d,j}^i$ , die als künstliche Pheromone bezeichnet werden, zugeordnet, welche die Pheromonwerte auf dieser Kante beschreiben. Der Pheromonwert  $\phi_{d,j}^i$  wird durch Ameisen modifiziert, wenn sie auf dem Weg zum Zielknoten  $v_d$  den Knoten  $v_i$  besuchen und danach zum Knoten  $v_j$  gehen. Der Pheromonwert  $\phi_{d,j}^i$  ist ein Indikator für die Nutzung der Kante  $e(i, j)$  auf dem Weg zwischen  $v_s$  und  $v_d$ .

Eine Ameise auf dem Knoten  $v_i$  verwendet die vorhandenen Pheromonwerte  $\phi_{d,j}^i$ , um die Übergangswahrscheinlichkeit für den Knoten  $v_j \in N_i$  als nächsten Hop zu berechnen. Dabei ist  $N_i$  die Menge der Nachbarn des Knotens  $v_i$ . Die

Übergangswahrscheinlichkeit  $p_{d,j}^i$  berechnet sich gemäß Gleichung 5.5.

$$p_{d,j}^i = \begin{cases} \frac{\varphi_{d,j}^i}{\sum_{k \in N_i} \varphi_{d,k}^i} & \text{wenn } j \in N_i \\ 0 & \text{wenn } j \notin N_i \end{cases} \quad (5.5)$$

Die Übergangswahrscheinlichkeiten  $p_{d,j}^i$  eines Knotens  $v_i$  erfüllen die Anforderung

$$\sum_{j \in N_i} p_{d,j}^i = 1.$$

Beim Übergang von einem Knoten auf den nächsten ändern die Ameisen auch die Pheromonwerte für die gewählte Kante, wodurch die *positive Verstärkung* der Pheromone gegeben ist. Die Verstärkung der Pheromone erfolgt additiv um den Faktor  $\Delta\varphi$ , der von einer gegebenen Kostenfunktion  $\Delta\varphi = f(c)$  abhängt. Im einfachsten Fall ist die Kostenfunktion  $f(c) = \text{const}$ , d.h. die Pheromonkonzentration wird um einen konstanten Wert  $\Delta\varphi$  erhöht. Eine Ameise, die vom Knoten  $v_i$  zum Knoten  $v_j$  übergeht, modifiziert die Pheromonkonzentration für die Kante  $e(i, j)$  gemäß der Vorschrift aus Gleichung 5.6.

$$\varphi_{d,j}^i := \varphi_{d,j}^i + \Delta\varphi \quad (5.6)$$

Analog zu natürlichen Pheromonen unterliegen künstliche Pheromone auf den Kanten des Graphen  $G$  auch der Verflüchtigung, wodurch sie mit der Zeit abnehmen und die *negative Verstärkung* widerspiegeln. Der Verflüchtigungsprozess wird durch regelmäßiges Verringern der Pheromonwerte nachempfunden. Dies geschieht im einfachsten Fall multiplikativ gemäß Gleichung 5.7.

$$\varphi_{d,j}^i := (1 - q) \cdot \varphi_{d,j}^i, \quad q \in (0, 1] \quad (5.7)$$

Die Pheromonwerte für die Kanten des Knotens  $v_i$  ändern sich nach dem Weiterleiten eines Datenpakets an den Nachbarknoten  $v_j$  nach Gleichung 5.8.

$$\varphi_{d,j}^i := \begin{cases} (1 - q) \cdot \varphi_{d,j}^i + \Delta\varphi & \text{wenn } j \text{ nächster Hop} \\ (1 - q) \cdot \varphi_{d,j}^i & \text{sonst} \end{cases} \quad (5.8)$$

Die folgenden Abschnitte beschreiben die Umsetzung der hier beschriebenen Heuristik in ein Routingprotokoll für mobile multi-hop Ad-hoc-Netze. Vorher wird kurz an einigen wichtigen Stichpunkten diskutiert, warum sich Ameisenalgorithmen für den Einsatz in mobilen Ad-hoc-Netzen eignen.

- **Dynamische Netztopologie**

Gerade die dynamische Netztopologie, die durch die Knotenmobilität hervorgerufen wird, macht mobile Ad-hoc-Netze schwierig handhabbar. Die Ameisenalgorithmen basieren auf dem Verhalten einzelner Ameisen. In Abschnitt 5.3 wurde gezeigt, dass Ameisenalgorithmen nicht eine einzelne Lösung finden, sondern mehrere Lösungen bilden. Dadurch kann die Dynamik des Netzes besser aufgefangen werden.

- **Lokale Informationen**

Die Ameisenalgorithmen arbeiten nur mit lokalen Informationen, wobei jede Ameise ihre Entscheidungen unabhängig von anderen Ameisen fällt, sich jedoch von den Pheromonspuren leiten lässt. Dadurch müssen keine großen Informationsmengen im Netzwerk gesammelt und verteilt werden.

- **Quality of Service**

Wie in Abschnitt 5.3 beschrieben, berücksichtigt die Originalheuristik eine sogenannte Sichtbarkeit zwischen den Knoten, die wie der Pheromonwert gewichtet ist. Es ist möglich – in dieser Arbeit aber noch nicht berücksichtigt – die Linkqualität, Verzögerungszeit oder Energieperspektiven in die Entscheidung mit einzubeziehen.

- **Multipfadrouting**

Die Eigenschaft, dass mehrere Lösungen mit unterschiedlicher Qualität gefunden werden, ermöglicht die einfache Umsetzung des Multipfad routings. Dadurch kann die Last in Richtung des Zielknotens verteilt werden.

## 5.4.2 ARA im Detail

Jeder Knoten  $v_i$  besitzt eine Routingtabelle  $R_i$ . Die Einträge der Routingtabelle sind Wahrscheinlichkeiten  $p_{d,j}^i$  mit der ein Paket mit Zielknoten  $v_d$  als nächsten Knoten  $v_j$  wählt. Da der Ameisenroutingalgorithmus nach Bedarf arbeitet, variiert die Größe der Routingtabelle mit der Verkehrscharakteristik und der Anzahl der Verbindungen, die der Knoten weiterleitet. Die Einträge der Routingtabelle werden sowohl von normalen Datenpaketen als auch von Agenten modifiziert, die den Knoten besuchen. Der Ameisenroutingalgorithmus besteht aus drei Phasen:

- (1) **Pfadfindung:** dient der Auffindung eines neuen Pfades zwischen einem Quellknoten und einem Zielknoten.

- (2) **Pfadpflege:** dient der Pflege vorhandener Pfade.
- (3) **Fehlerbehandlung:** dient der Behandlung von Routingfehlern.

Im Vergleich zu anderen Routingalgorithmen werden bei ARA drei Phasen unterschieden. Das liegt daran, dass die Fehlerbehandlung und die Pfadpflege prinzipiell unterschiedlich behandelt werden. Für die Pfadpflege verwendet ARA keine besonderen Routingpakete, stattdessen geschieht die Pfadpflege durch die weitergeleiteten Datenpakete. Die Behandlung von Fehlern ist bei ARA aufwändiger und deswegen als eigene Phase ausgewiesen.

Wenn ein Quellknoten ein Paket an einen Zielknoten senden möchte, sucht er zuerst in seiner Routingtabelle nach einem Pfad. Findet er einen Eintrag in seiner Routingtabelle, so kann er direkt mit dem Übertragen der Pakete beginnen. Ansonsten geht der Quellknoten in die erste Phase von ARA über.

### Phase 1: Pfadfindung

In der Pfadfindungsphase werden neue Pfade gefunden, dies geschieht durch den Einsatz einer Vorwärtsameise und einer Rückwärtsameise. Eine Vorwärtsameise wird als *Forward-Ant (FANT)* und eine Rückwärtsameise als *Backward-Ant (BANT)* bezeichnet. Ein FANT ist ein Agent, der eine Pheromonspur zum Quellknoten im Netz einrichtet. Im Gegensatz hierzu legt ein BANT eine Pheromonspur zum Zielknoten aus. Der FANT und der BANT sind kleine Pakete mit einer eindeutigen Sequenznummer, Quellknotenadresse, Zielknotenadresse und weiteren Informationen, wodurch die Knoten in der Lage sind Duplikate zu erkennen.

Der Quellknoten broadcastet einen FANT, welcher von den Nachbarn weitergeleitet wird. Dadurch empfangen alle Knoten im Netz den FANT (siehe Abbildung 5.9a)). Ein Knoten, der einen FANT zum ersten Mal empfängt, erzeugt einen Eintrag in seiner Routingtabelle. Ein Routingtabelleneintrag ist ein Tripel der Form (destination address, next hop, pheromone value). Der Knoten interpretiert die Quellknotenadresse des FANTS als destination address und die Adresse des Vorgängerknotens als next hop und berechnet den Pheromonwert in Abhängigkeit der Anzahl der Knoten, die der FANT besucht hat um diesen Knoten zu erreichen. Tatsächlich wird der Pheromonwert mit dem TTL-Wert (*Time-To-Live*) der FANT initialisiert. Danach leitet der Knoten den FANT an seine Nachbarn weiter, jedoch nur einmal. Duplikate von FANTS werden durch die Sequenznummer und der Quellknotenadresse erkannt und verworfen. Wenn der FANT den Zielknoten erreicht, wird er vom Zielknoten gelöscht. Danach erzeugt der Zielknoten einen BANT, um ihn an den Quellknoten zu senden (siehe Abbildung 5.9b)). Der BANT hat eine ähnliche Aufgabe wie der FANT, nämlich den Aufbau

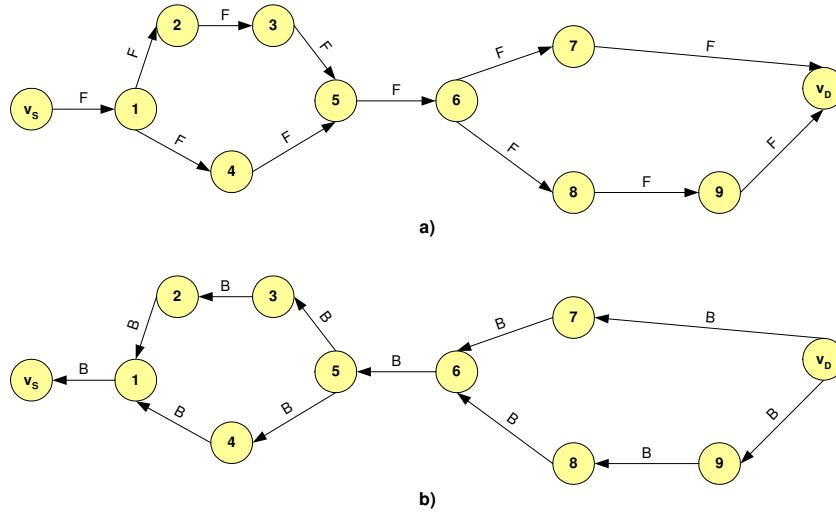


Abbildung 5.9: Phase 1: Pfadfindung von ARA. a) Ein Forward-Ant (F) wird vom Quellknoten  $v_s$  in Richtung des Zielknotens  $v_d$  gesendet. Der FANT wird von den Knoten weitergeleitet, die Knoten initialisieren dabei ihre Routingtabellen. Dadurch wird im Netz eine Pheromonspur in Richtung des Quellknotens ausgelegt. b) Der Zielknoten schickt einen BANT (B) als Antwort an den Quellknoten. Die Aufgabe des BANTS ist analog zu der Aufgabe des FANTS. Durch das Weiterleiten der BANT können die Knoten im Netz ihre Routingtabellen vollständig initialisieren, d.h. es wird auch eine Pheromonspur in Richtung des Zielknotens ausgelegt.

einer Pheromonspur zum Zielknoten. Deshalb führen alle Knoten, die einen BANT zum ersten Mal empfangen, die gleiche Prozedur aus wie bei einer FANT, d.h. es wird ein Routingtabelleneintrag in Richtung des Zielknotens angelegt. Nachdem der Quellknoten den BANT empfangen hat, kann die eigentliche Kommunikation beginnen.

Die Abbildung 5.9 stellt die Pfadfindungsphase von ARA grafisch an einem Beispiel dar. Der Quellknoten  $v_s$  broadcastet den FANT, sodass alle Knoten ihn empfangen. Im dargestellten Fall gibt es ausgehend von den Knoten 5 und 6 jeweils zwei Wege zum Quell- und Zielknoten. Der FANT erstellt zwei Pheromonspuren in Richtung des Quellknotens, je eine über Knoten 3 und Knoten 4. Analog hierzu legt der BANT zwei Pheromonspuren zum Zielknoten aus, je eine über Knoten 7 und 8.

## Phase 2: Pfadpflege

Die zweite Phase von ARA ist die Pfadpflege. In dieser Phase werden vorhandene Pfade im laufenden Betrieb gewartet und verbessert. Hierfür benötigt ARA keine speziellen *Wartungspakete*. Wurden die Pheromonspuren durch den FANT und den BANT einmal erzeugt, wird die Pfadpflege mit Hilfe nachfolgender Datenpakete durchgeführt. Hierfür spielt die Dynamik der Pheromonwerte eine wichtige Rolle. Ähnlich dem Vorbild aus der Natur, wo die Pheromonkonzentration auf einem bestimmten Weg nicht ihren Initialwert behält, sondern sich in Abhängigkeit der Weg-Nutzung ändert, ändern sich auch die künstlichen Pheromonwerte dynamisch. Wenn ein Knoten  $v_i$  ein Datenpaket in Richtung des Zielknotens  $v_d$  an seinen Nachbarn  $v_j$  weiterleitet, dann erhöht er die Pheromonkonzentration der Kante  $e(i, j)$ , also den Routingtabelleneintrag  $(v_d, v_j, \phi_{d,j}^i)$  um den Wert  $\Delta\varphi$ , d.h. der Pfad in Richtung des Zielknotens wird durch die Datenpakete verstärkt. Gleichzeitig erhöht der nächste Knoten  $v_j$  den Eintrag  $(v_s, v_i, \phi_{s,i}^j)$  um  $\Delta\varphi$ , d.h. der Pfad in Richtung des Quellknotens wird ebenfalls verstärkt (siehe Abbildung 5.10). Der Verflüchtigungsprozess der Pheromone ist ähnlich der von realen Pheromonen und wird gemäß der Gleichung 5.7 durchgeführt.

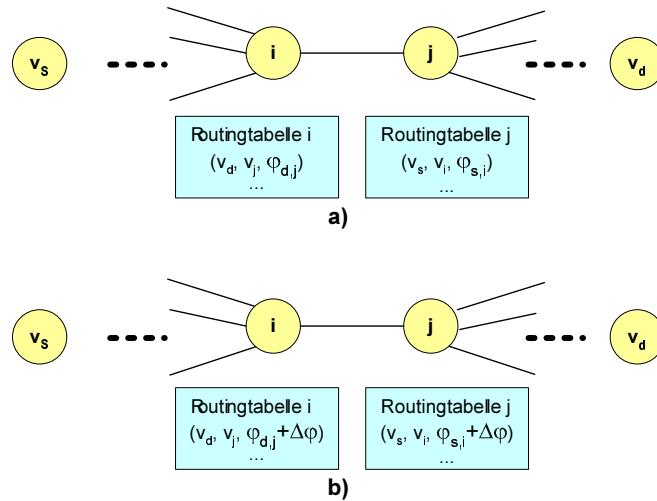


Abbildung 5.10: Funktionsweise der Pfadpflege bei ARA. a) Ein Datenpaket vom Quellknoten  $v_s$  kommt beim Knoten  $v_i$  an. b) Knoten  $v_i$  hat das Datenpaket an Knoten  $v_j$  weitergeleitet.

Im Folgenden wird an einem Beispiel, welches auf Abbildung 5.9 basiert, die Pfadpflege besprochen. Die Routingtabellen von Knoten 5 und 6 sehen zu einem bestimmten Zeitpunkt wie folgt aus:

Routingtabelle von Knoten 5

Zielknoten	Next Hop	Pheromone
$v_s$	3	$\varphi_1$
$v_s$	4	$\varphi_2$
$v_d$	6	$\varphi_3$
$\vdots$	$\vdots$	$\vdots$

Routingtabelle von Knoten 6

Zielknoten	Next Hop	Pheromone
$v_s$	5	$\varphi_4$
$v_d$	7	$\varphi_5$
$v_d$	8	$\varphi_6$
$\vdots$	$\vdots$	$\vdots$

Die Modifikationen in den Routingtabellen von Knoten 5 und 6 nach dem Weiterleiten eines Pakets von Knoten 5 in Richtung des Zielknotens ergeben sich zu:

Routingtabelle von Knoten 5

Zielknoten	Next Hop	Pheromone
$v_s$	3	$\varphi_1$
$v_s$	4	$\varphi_2$
$v_d$	6	$\varphi_3 + \Delta\varphi$
$\vdots$	$\vdots$	$\vdots$

Routingtabelle von Knoten 6

Zielknoten	Next Hop	Pheromone
$v_s$	5	$\varphi_4 + \Delta\varphi$
$v_d$	7	$\varphi_5$
$v_d$	8	$\varphi_6$
$\vdots$	$\vdots$	$\vdots$

Es hat sich nur der Eintrag für den Zielknoten  $v_d$  in der Routingtabelle von Knoten 5 geändert. Die Veränderung in der Routingtabelle von Knoten 6 ist analog. Nur der Pheromonwert für den Quellknoten  $v_s$  hat sich geändert. Die Veränderungen in beiden Routingtabellen wurden auf die gleiche Art durchgeführt.

Die Pheromonwerte bleiben nicht konstant, sondern nehmen mit der Zeit ab. Die Verflüchtigung der Pheromonwerte erfolgt multiplikativ mit Faktor  $(1 - q) = 0,1$  (siehe Gleichung 5.7). Die Routingtabellen der Knoten 5 und 6 haben nach einem vollständigen Aktualisierungsprozess folgendes Aussehen:

Routingtabelle von Knoten 5

Zielknoten	Next Hop	Pheromone
$v_s$	3	$\varphi_1 \cdot (1 - q)$
$v_s$	4	$\varphi_2 \cdot (1 - q)$
$v_d$	6	$(\varphi_3 + \Delta\varphi) \cdot (1 - q)$
$\vdots$	$\vdots$	$\vdots$

Routingtabelle von Knoten 6

Zielknoten	Next Hop	Pheromone
$v_s$	5	$(\varphi_4 + \Delta\varphi) \cdot (1 - q)$
$v_d$	7	$\varphi_5 \cdot (1 - q)$
$v_d$	8	$\varphi_6 \cdot (1 - q)$
$\vdots$	$\vdots$	$\vdots$

### Schleifenerkennung

Das hier beschriebene Verfahren kann zu unerwünschten Schleifen führen, die sich negativ auf die Leistung auswirken können. Der Ameisenroutingalgorithmus verhindert die Schleifenbildung durch eine einfache Methode.

Jedes Datenpaket kann durch seine Sequenznummer und die Quellknotenadresse von einem Knoten eindeutig identifiziert werden. Empfängt ein Knoten  $v_i$  ein Datenpaket von einem Knoten  $v_j$  und erkennt, dass er das Datenpaket vorher an einen anderen Knoten versendet hatte, muss eine Schleife entstanden sein. Der Knoten  $v_i$  setzt den `flag_loop` des Datenpakets und sendet ihn sofort an den Knoten  $v_j$  zurück. Der Knoten  $v_j$  löscht den entsprechenden Routingtabelleneintrag, sodass keine weiteren Datenpakete hierüber weitergeleitet werden. Datenpakete mit gesetztem `flag_loop` werden bevorzugt behandelt.

### Phase 3: Fehlerbehandlung

Die dritte und letzte Phase von ARA ist die Behandlung von Routingfehlern, die hauptsächlich durch die Knotenmobilität verursacht werden, und in mobilen Ad-hoc-Netzen sehr häufig sind. Die Fehlerbehandlung arbeitet mit der Pfadpflege Hand in Hand.

ARA erkennt Routingfehler aufgrund von fehlenden Bestätigungspaketen auf der MAC-Schicht. Erkennt ein Knoten einen Verbindungsfehler, der z.B. durch die Knotenmobilität verursacht sein könnte, löscht er den entsprechenden Routingtabelleneintrag. Anschließend sucht der Knoten nach einem anderen Eintrag für den Zielknoten in seiner Routingtabelle. Wenn die Suche erfolgreich ist, versucht der Knoten das Datenpaket über diesen Nachbarknoten an den Zielknoten zu transportieren. Der Pheromonwert des gefundenen Routingeintrags wird erhöht.

Wenn der Knoten keine weiteren Wege zum Zielknoten in seiner Routingtabelle findet, setzt er den `flag_rf` und informiert alle seine direkten Nachbarn darüber. Die Nachbarknoten löschen zuerst alle Einträge in ihren Routingtabellen mit dem entsprechenden Zielknoten, die über diesen Knoten führen. Danach suchen sie nach einem anderen Weg, um das Datenpaket an den Zielknoten zu transportieren. Knoten, die einen Weg kennen, leiten das Datenpaket über diesen an den Zielknoten weiter. Wenn das Datenpaket beim Zielknoten ankommt, erkennt dieser durch den gesetzten Flag `flag_rp`, dass ein Fehler aufgetreten ist. Deshalb generiert der Zielknoten einen BANT und schickt ihn an den Quellknoten. Der BANT legt auf dem Weg zum Quellknoten eine starke Pheromonspur zum Zielknoten aus, sodass nachfolgende Datenpakete diesem folgen. Abbildung 5.11 stellt die Vorgehensweise der Fehlerbehandlung an einem Beispiel dar.

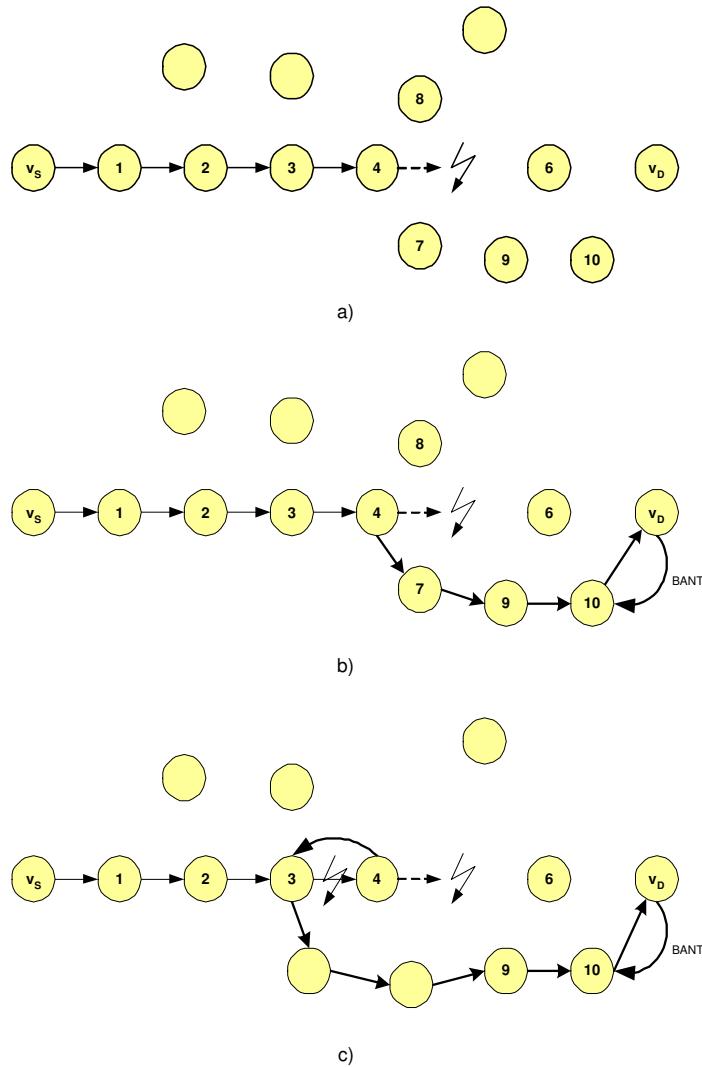


Abbildung 5.11: Fehlerbehandlung bei ARA. a) Datenpaket kommt vom Quellknoten  $v_S$  über mehrere Zwischenknoten beim Knoten 4 an, der einen Verbindungsabbruch und den dadurch bedingten Fehler erkennt. b) Knoten 4 kennt einen weiteren Weg über Knoten 7 zum Zielknoten  $v_D$ . Das Datenpaket kann über den alternativen Weg an den Zielknoten transportiert werden. Nach dem Empfang sendet der Zielknoten einen BANT aus, um die Pheromoninformationen im Netz aufzufrischen. c) Knoten 4 kennt keinen weiteren Weg, um das Datenpaket von a) weiterzuleiten. Deshalb informiert er den Vorgänger-Knoten, der zuerst seine eigene Routingtabelle korrigiert. Danach sucht er nach einem Alternativweg und kann darüber das Datenpaket an den Zielknoten weiterleiten.

Wenn das Datenpaket über die direkten Nachbarn nicht zum Zielknoten transportiert werden konnte, wird keine weitere Anstrengung unternommen. Nachfolgende Pakete gelangen jedoch jetzt nur noch bis zum vorherigen Knoten. Die oben beschriebene Prozedur wird für die nachfolgenden Knoten ebenfalls ausgeführt. Durch diese Vorgehensweise entsteht entweder ein intakter Pfad zwischen dem Quell- und Zielknoten, oder die Fehlerbehandlung gelangt zum Quellknoten. Wenn die Fehlerbehandlung beim Quellknoten ankommt, initiiert dieser eine neue Pfadfindung, wodurch im gesamten Netz neue Pheromonspuren ausgelegt werden.

Es hat sich gezeigt, dass diese Vorgehensweise bei der Fehlerbehandlung zu aufwändig ist und eine zu lange Zeit erfordert, bis der Quellknoten über den Fehler informiert ist. Dies wirkte sich negativ auf die Leistung des Routingalgorithmus aus. Im Voraus ist nicht bekannt, wie viele Knoten zwischen dem Quellknoten und dem Knoten, der den Fehler erkennt, liegen. Deshalb kann die Zeit nicht abgeschätzt werden, bis der Quellknoten informiert ist und eine neue Pfadsuche initiiert. In dieser Zeit sendet jedoch der Quellknoten weiter, wodurch sich immer mehr Datenpakete im Netz befinden. Ein Teil der Datenpakete folgen einem Pfad, der nicht bis zum Zielknoten führt.

Deshalb wurde eine zweite Variante der Fehlerbehandlung in den Ameisenroutingalgorithmus integriert (siehe Abbildung 5.12). Der Knoten, der den Fehler erkannt hat, informiert seine direkten Nachbarn über den Sachverhalt, wie oben beschrieben. Ein Nachbarknoten, der das Paket nicht weiterleiten kann, weil es z.B. keinen weiteren Weg kennt, lässt den Fehlerflag des Datenpaketes gesetzt und informiert wiederum seine direkten Nachbarn über den Fehler. Diese Knoten passen ihre Routingtabellen den Gegebenheiten an, d.h. sie löschen den Routingtabelleneintrag mit dem Zielknoten und über den informierenden Knoten. Anschließend suchen sie nach einem zweiten Weg in ihrer Routingtabelle. Bei Erfolg wird versucht das Datenpaket an den Zielknoten weiterzuleiten.

Somit wird der Fehler von der Fehlerstelle aus bis zum Quellknoten propagiert oder durch einen Knoten, der zwischen dem Quellknoten und der Fehlerstelle liegt, an den Zielknoten weitergeleitet. Wenn die Dichte der Knoten an einer Stelle hoch ist, kann es passieren, dass ein Knoten viele Einträge in seiner Routingtabelle zum Zielknoten besitzt. Um die Last gering zu halten, wird die Anzahl der Versuche, die ein Knoten hat um das Datenpaket weiterzuleiten, begrenzt. Untersuchungen haben gezeigt, dass ein Versuch tolerierbar ist.

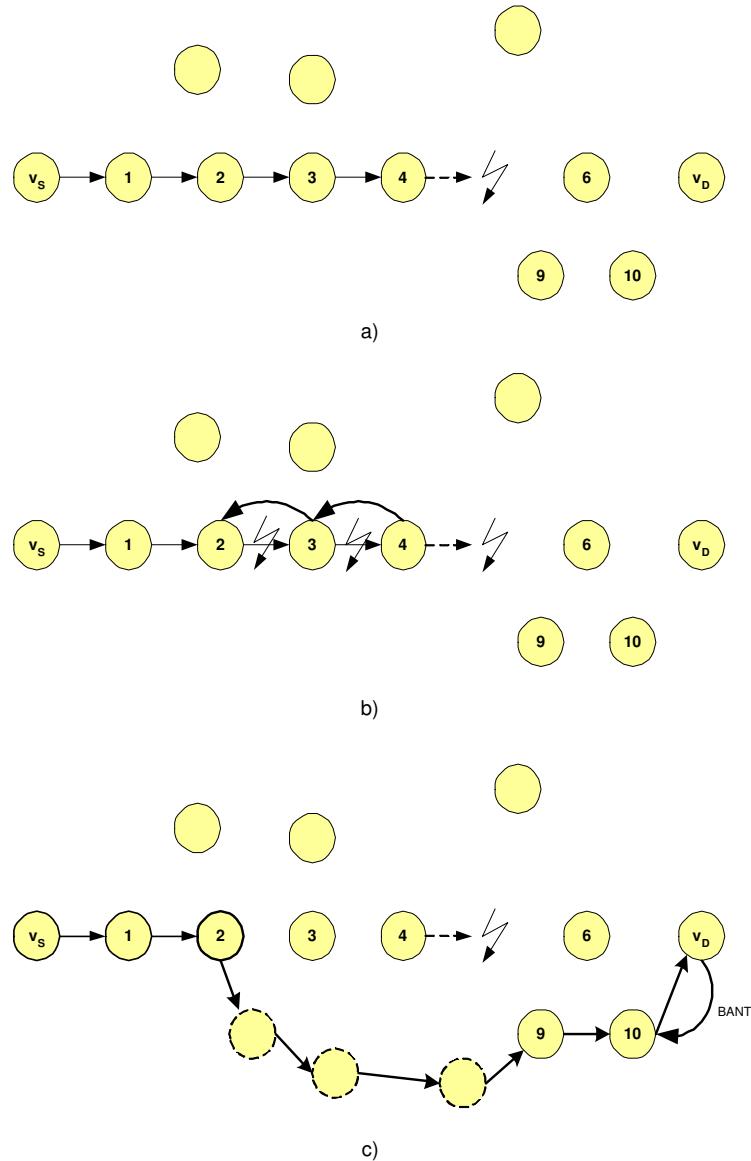


Abbildung 5.12: Fehlerbehandlung bei ARA. a) Knoten entdeckt Routingfehler. b) Direkte Nachbarn des Knotens können Datenpaket nicht weiterleiten, weil sie keinen weiteren Weg zum Zielknoten kennen. c) Der Fehler wird in Richtung des Quellknotens propagiert. Knoten 2 liegt zwischen Quellknoten und Fehlerquelle und kennt einen anderen Weg zum Zielknoten und kann das Datenpaket weiterleiten. Der Zielknoten sendet ein BANT, um die Pheromonwerte aufzufrischen.

### 5.4.3 Module von ARA

Der Ameisenroutingalgorithmus wurde um viele Module erweitert. Die Module dienen einerseits zur Verbesserung der Nachahmung von realen Ameisenkolonien, und andererseits zur Behebung von technischen Schwierigkeiten.

#### Routingentscheidung

Die grundlegende Methode der Weiterleitung von Datenpaketen wurde in Abschnitt 5.4.2 diskutiert, jedoch wurde die konkrete Umsetzung nicht beschrieben. Der Ameisenroutingalgorithmus kann in zwei unterschiedlichen Modi Datenpakete weiterleiten, die sich bei der Wahl des nächsten Nachbarn für ein Datenpaket unterscheiden:

- **Routing nach Max-Pheromonwert**

Hierbei wird versucht, Datenpakete auf einem optimalen Weg zum Zielknoten zu transportieren. Dabei ist ein optimaler Weg durch die Länge des Pfades in Anzahl der Hops, die auf dem Pfad liegen, gegeben. Deshalb wählt der Knoten  $v_i$  für die Weiterleitung eines Datenpaketes an den Zielknoten  $v_d$  den Nachbarn  $v_j$  mit dem höchsten Pheromonwert aus.

$$p_{d,j}^i = \begin{cases} 1 & \text{wenn } \varphi_{d,j}^i = \max_{k \in N_i} \{\varphi_{d,k}^i\} \\ 0 & \text{sonst} \end{cases}$$

- **Probabilistisches Routing**

Beim probabilistischen Routing werden die Datenpakete einer Datenverbindung über mehrere Pfade zum Zielknoten übertragen. Dadurch ist es möglich, die Leistung des Routingalgorithmus zu verbessern. Beim Weiterleiten wird ein Nachbarknoten mit höherem Pheromonwert mit einer größeren Wahrscheinlichkeit gewählt. Es werden jedoch auch Nachbarknoten gewählt, deren Pheromonwert kleiner ist. Der Knoten  $v_i$  wählt für die Weiterleitung eines Datenpaketes an den Zielknoten  $v_d$  seinen Nachbarn  $v_j$  mit der folgenden Wahrscheinlichkeit aus.

$$p_{d,j}^i = \begin{cases} \frac{\varphi_{d,j}^i}{\sum_{k \in N_i} \varphi_{d,k}^i} & \text{wenn } j \in N_i \\ 0 & \text{wenn } j \notin N_i \end{cases}$$

Die Version von ARA, die Routingentscheidungen nach dem maximalen Pheromonwert trifft, wird beim Leistungsvergleich in Abschnitt 5.5 als ARA<sub>max</sub>

und die Version von ARA, die Wegentscheidungen probabilistisch trifft, als  $\text{ARA}_{\text{stat}}$  bezeichnet.

### Kontinuierliche Verflüchtigung der Pheromonwerte

Die Verflüchtigung von Pheromonen wurde in der ersten Version von ARA getaktet durchgeführt. Dies hatte jedoch den Nachteil, dass zwei Pfade, die innerhalb desselben Takts verwendet wurden, jedoch unterschiedlich alt sind, gleich alternen. In der Natur verflüchtigen sich die Pheromone kontinuierlich. Der Ameisenroutingalgorithmus ahmt die kontinuierliche Verflüchtigung der Pheromonwerte ebenfalls. Hierzu merkt sich der Knoten den Zeitpunkt des letzten Zugriffs  $t_{la}$  auf die Routingtabelle. Bei einem erneuten Zugriff auf die Routingtabelle zur Zeit  $t$  werden dann die Pheromonwerte vor der Auswahl des eigentlichen Pfads gemäß der Gleichung 5.9 angepasst.

$$\varphi_{d,j}^i(t) = \varphi_{d,j}^i(t_{la}) \cdot e^{t_{la}-t} \quad (5.9)$$

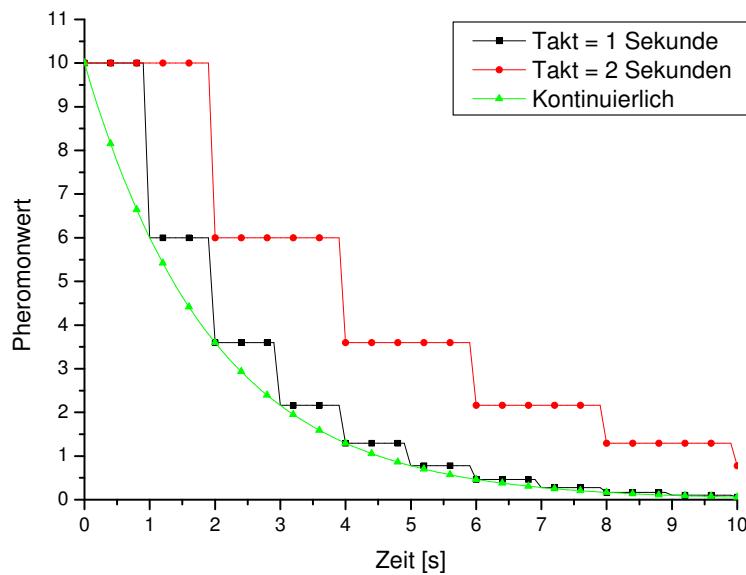


Abbildung 5.13: Verflüchtigung der Pheromonwerte.

Abbildung 5.13 zeigt die Verflüchtigung der Pheromonwerte im getakteten wie im kontinuierlichen Fall. Im getakteten Fall macht es keinen Unterschied,

zu welchem Zeitpunkt des Takts ein Pfad verwendet wurde. An einem einfachen Beispiel soll der Unterschied verdeutlicht werden. Der Knoten  $v_i$  fügt zwei neue Pfade für den Zielknoten  $v_d$  je über die Nachbarknoten  $v_j$  und  $v_k$  zu den Zeitpunkten  $t_1 = 7,1$  und  $t_2 = 7,6$  in seine Routingtabelle mit dem Initialpheromonwert  $\varphi_{d,j}^i(t_1) = 10$  bzw.  $\varphi_{d,k}^i(t_2) = 10$  ein. Sei  $q = 0,7$ , wenn der Knoten  $v_i$  zum Zeitpunkt  $t_3 = 8,5$  ein Datenpaket weiterleiten muss, haben beide Pheromone den Wert  $\varphi_{d,j}^i(t_3) = (1 - q) \cdot \varphi_{d,j}^i(t_1) = 7$  und  $\varphi_{d,k}^i(t_3) = (1 - q) \cdot \varphi_{d,k}^i(t_2) = 7$ , da nur ein Takt vergangen ist. Im kontinuierlichen Fall sind die Pheromonwerte wie folgt  $\varphi_{d,j}^i(t_3) = \varphi_{d,j}^i(t_1) \cdot e^{-1,4} = 2,5$  und  $\varphi_{d,k}^i(t_3) = \varphi_{d,k}^i(t_1) \cdot e^{-0,9} = 4,1$ . Der Pheromonwert des jüngeren Pfads ist um 60% höher und würde bei der Weiterleitung daher bevorzugt.

### Priorisierte Pakete

In der Grundversion des Ameisenroutingalgorithmus werden alle Pakete im Netz gleich behandelt. Dadurch kann es passieren, dass Pakete mit wichtigen Informationen verzögert am Ziel ankommen. Der Ameisenroutingalgorithmus wurde so erweitert, dass FANTS, BANTS, Pakete mit gesetztem Schleifenflag und Pakete mit gesetztem Pfadfehlerflag gegenüber anderen Paketen bevorzugt behandelt werden. Dadurch wird sichergestellt, dass wichtige Informationen schnellmöglichst im Netz transportiert werden.

### Fluten der BANT und Multipfadrouting

Ursprünglich wurde in ARA nur der FANT geflutet. Der BANT hingegen wurde vom Zielknoten per unicast an den Quellknoten gesendet. Der Vorteil dieser Vorgehensweise war, die Last durch das Fluten zu reduzieren. Es hat sich jedoch gezeigt, dass dadurch das Multipfadrouting nicht richtig ausgenutzt werden kann, da die Pheromonwerte für beide Richtungen nicht gleichartig positiv verstärkt werden.

Durch das Fluten der BANT vom Zielknoten zum Quellknoten werden alle möglichen Pfade zwischen Quell- und Zielknoten positiv verstärkt werden. Nach einer kurzen Zeit verschwinden die weniger guten Pfade durch die negative Verstärkung. Jedoch kann der Ameisenroutingalgorithmus nun seine Fähigkeit, die Last auf mehrere Pfade zu verteilen, besser ausnutzen.

### Sendepuffer und minimal FANT

Eine Schwierigkeit bei der Grundversion des Ameisenroutingalgorithmus war, dass als FANT das erste Paket einer Verbindung ausgesendet wurde. Dies

hatte den Nachteil, dass bei großen Paketen auch ein großer FANT im Netz geflutet werden musste. Dies erhöhte zusätzlich den Overhead des Algorithmus. Ein weiteres Problem trat auf, wenn weitere Pakete von der Transportschicht kamen, während der FANT oder der BANT unterwegs waren. Die nachrückenden Pakete wurden aufgrund des Fehlens eines Pfades ebenfalls als FANTS ausgesendet, wodurch der Overhead weiter erhöht wurde.

Ein Knoten kann in der aktuellen Version nachrückende Pakete im Sendepuffer ablegen, bis ein Pfad zwischen Quell- und Zielknoten gefunden ist. Als weitere Verbesserung wird nicht mehr das erste Datenpaket als FANT verwendet, sondern es wird ein kleines Paket, bestehend aus Adressen, Sequenznummer und Flags, generiert und als FANT bzw. BANT im Netz geflutet. Dadurch wird zum einen die Übertragung der FANT bzw. BANT beschleunigt und zum anderen der Overhead reduziert.

Das Simulationstool ns-2 hat in seiner aktuellen Version eine sehr beschränkte Implementierung des *Address Resolution Protocols* (ARP). Das Simulationstool ist nicht in der Lage, mehrere Anfragen parallel zu bearbeiten. Wenn ein Paket von der Transportschicht auf der Netzwerkschicht ankommt, wird zuerst versucht, die IP-Adresse auf eine MAC-Adresse abzubilden. Für diesen Zweck wird das Address Resolution Protocol eingesetzt. Sollte jedoch bevor die Bearbeitung fertig ist, ein zweites Paket von der Transportschicht ankommen, für das wiederum eine Adressanfrage durchgeführt werden muss, dann verwirft ns-2 das erste Paket und startet die Anfrage für das neue Paket. Diese Einschränkung hat in der Basisversion des Ameisenroutingalgorithmus dazu geführt, dass bei Datenströmen mit kurzem Paketabstand eine erfolgreiche Adressauflösung nicht durchgeführt werden konnte.

Um diese Probleme zu beheben, wurde für den Ameisenroutingalgorithmus ein eigenes ARP-Modul implementiert, welches parallel mehrere Anfragen bedienen kann. Das neue ARP-Modul von ARA speichert die aufgelösten Adresspaare in einem Puffer, sodass die Netzwerkschicht darauf zugreifen kann. Das ARP-Modul arbeitet mit dem MAC-Tap zusammen, um den Bedarf an Adressauflösungen zu verringern.

## MAC-Tap

Mobile multi-hop Ad-hoc-Netze kommunizieren über Funk. Hierdurch können prinzipiell alle Knoten, die in der Reichweite eines Senders sind, die Pakete empfangen. Ein Nachteil der Funkkommunikation ist, dass nur ein Knoten gleichzeitig senden kann und die Gefahr von Paketkollisionen sehr hoch ist. Um die Paketkollisionen zu reduzieren und dadurch die Leistung des Netzes zu steigern, wurden unterschiedliche MAC-Verfahren entwickelt, von denen einige schon in Abschnitt 2.4 besprochen wurden. Die meisten

weiterentwickelten Protokolle für drahtlose Netze verwenden Kontrollnachrichten auf der MAC-Schicht, um den Zugriff auf das Medium zu steuern. Diese Kontrollnachrichten bestehen aus Adress- und Zeitinformationen. Die Kontrollnachrichten werden genau wie Datenpakete von allen Knoten in der Nachbarschaft des Senders empfangen. Jedoch werden die Pakete auf der MAC-Schicht gefiltert und nur die Pakete an die höheren Schichten weitergeleitet, die für diesen Knoten bestimmt sind. Es ist offensichtlich, dass die Kontrollnachrichten sehr wichtige Informationen über die Nachbarschaft eines Knotens enthalten. Um diese Informationen aus der Nachbarschaft für das Routing zu verwenden, wurde das MAC-Tap konzipiert und implementiert.

Da die Ergebnisse dieser Arbeit auf Simulationen mit mobilen Knoten gemäß IEEE 802.11 basieren, und diese auf der MAC-Schicht als MAC-Protokoll CSMA/CA (siehe Abschnitt 2.4.1) einsetzen, ist der MAC-Tap auf dieses Verfahren zugeschnitten. Es ist jedoch möglich den MAC-Tap mit jedem anderen MAC-Protokoll einzusetzen.

Die Aufgabe des MAC-Tap ist, alle empfangenen Pakete auf interessante Informationen zu analysieren und sie ggf. an die Routingschicht weiterzugeben. Die Routingfunktion profitiert davon, da die Routingtabelle optimiert werden kann. Weiterhin kann hierdurch die Anzahl der Adressauflösungen durch ARP verringert werden.

Das CSMA/CA verwendet für die Zugriffsteuerung drei Kontrollnachrichten: Ready-To-Send (RTS), Clear-To-Send (CTS) und Acknowledgement (ACK). Ein RTS-Paket enthält die MAC-Adressen des Senders und des Empfängers (siehe Abbildung 5.14).

- **Empfang einer RTS-Nachricht**

Durch den Empfang einer RTS-Nachricht weiß ein Knoten, dass er in der Nähe des Senders ist. Eine RTS-Nachricht enthält die MAC-Adressen des Senders und des Empfängers. Der Knoten versucht mit Hilfe des MAC-Cache die IP-Adresse des Senders aufzulösen. Ist er erfolgreich, so werden alle Routingtabelleneinträge, die als Zielknoten den Sender haben, gelöscht. Nur der Eintrag mit direktem Sprung verbleibt in der Routingtabelle. Wenn die IP-Adresse des Senders nicht durch den MAC-Cache aufgelöst werden konnte, sind die Informationen der RTS-Nachricht wertlos.

Der Knoten versucht auch die IP-Adresse des Empfängers durch den MAC-Cache aufzulösen. Bei Erfolg merkt er sich die Adressinformationen des Empfängers, um sie später zu verwenden.

- **Empfang einer CTS-Nachricht**

Wie in Abbildung 5.14 dargestellt, enthält eine CTS-Nachricht nur die MAC-Adresse des Knotens vom dem die RTS-Nachricht stammt, also

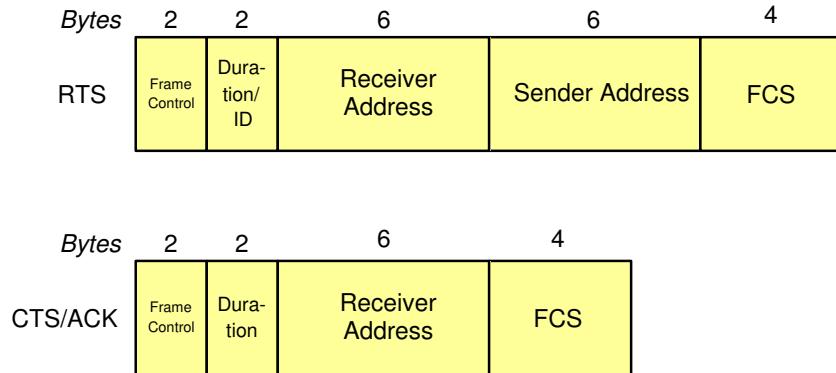


Abbildung 5.14: Kontrollnachrichten von IEEE 802.11.

dem eigentlichen Sender. Wenn die IP-Adresse des Senders durch den MAC-Cache aufgelöst werden kann, überprüft der Knoten, ob er vorher die zugehörige RTS-Nachricht erhalten hatte. Ist die Überprüfung erfolgreich, weiß der Knoten, dass der Empfänger auch in seiner Reichweite ist. Daraufhin löscht der Knoten die Routingtabelleneinträge, die als Ziel den Empfänger haben. Es bleibt wiederum nur der Eintrag mit dem direkten Sprung in der Routingtabelle übrig.

- **Empfang eines Datenpakets**

Datenpakete enthalten die meisten Informationen, da auf der IP-Ebene sowohl die Quellknotenadresse als auch die Zielknotenadresse enthalten ist. Auf der IP-Ebene wird ein erweiterter Header eingesetzt. In diese schreibt ein Knoten, der ein Datenpaket weiterleitet, seine eigene IP-Adresse und die IP-Adresse des Knotens, an den er das Paket weiterleitet. Auf der MAC-Schicht sind die MAC-Adressen des Senders und des Empfängers enthalten. Dadurch ist es einem Knoten möglich, die Adresspaare sowohl für den Sender als auch für den Empfänger zu extrahieren.

Die Vorteile des MAC-Tap werden insbesondere nach einem Neustart eines Knotens, wenn keine Pfadinformationen bekannt sind, deutlich. Durch den Einsatz der FANT bzw. der BANT werden im ganzen Netz Informationen verbreitet. Sowohl der FANT als auch der BANT stellen Datenpakete ohne Nutzlast dar. Dadurch kann jeder Knoten im Netz Informationen über seine Nachbarn sammeln, die er für spätere Rou-tingaufgaben benutzen kann.

- **Empfang einer ACK-Nachricht**

Die ACK-Nachrichten enthalten keine zusätzlichen Informationen, deshalb werden sie vom MAC-Tap nicht verarbeitet.

#### 5.4.4 Die Eigenschaften von ARA

Ein Routingalgorithmus für mobile Ad-hoc-Netze sollte gewissen Anforderungen genügen [MC98]. Im folgenden werden die Eigenschaften von ARA diesen Anforderungen gegenübergestellt und diskutiert:

- **Verteilter Algorithmus**

In ARA besitzt jeder Knoten eine Menge von künstlichen Pheromonen  $\varphi_{d,j}^i$  für eine Kante zwischen dem Knoten  $v_i$  und seinen Nachbarn  $v_j$ . Jeder Knoten kontrolliert seine Routingtabelle unabhängig von den anderen Knoten. Die Routingtabelle eines Knotens wird modifiziert, wenn Pakete von diesem Knoten weitergeleitet werden.

- **Schleifenfreiheit**

Die Knoten registrieren die Quellknotenadresse und die Sequenznummern von Paketen. Hierdurch können sie Paketduplikate erkennen und Schleifen auflösen.

- **On-Demand Operation**

Die Pfadfindung wird durch den Quellknoten gestartet, wenn er keinen Pfad zum Zielknoten kennt. Sofern keine Pakete versendet werden müssen, werden auch keine Pfade zwischen den Knoten aufgebaut.

- **Lokalität**

Die Routingtabelle und die statistischen Daten eines Knotens werden aufgrund von lokalen Informationen aufgebaut und gepflegt. Es werden weder Routingtabellen noch andere Informationen an andere Knoten übertragen.

- **Multipfadrouting und probabilistisches Routing**

Ein Knoten kann mehrere Pfade für ein Kommunikationspaar aus Quell- und Zielknoten verwalten. Die Wahl des aktuellen Pfades kann von der Umgebung, z.B. der Link-Qualität, abhängig gefällt werden. Auf Wunsch kann die Wahl des nächsten Knotens probabilistisch gefällt werden, wodurch die Last auf mehrere Pfade in Richtung des Zielknotens verteilt wird.

## 5.5 Ergebnisse

Der *Ameisenroutingalgorithmus* wurde im Simulationstool ns-2 [FV00] implementiert. Für die Ergebnisse wurden Simulationen mit 50 mobilen Knoten gemäß IEEE 802.11 durchgeführt. Auf einer Simulationsfläche von 1500 m × 300 m Größe bewegen sich die Knoten mit einer maximalen Geschwindigkeit von 10 m/s nach dem Random-Waypoint-Mobility Modell [BMJ<sup>+</sup>98] über eine Simulationszeit von 900 Sekunden. Die Mobilität des Netzes wurde durch 7 unterschiedliche Pausenzeiten ausgedrückt. Diese waren 0, 30, 60, 120, 300, 600 und 900 Sekunden. Die Pausenzeiten spiegeln den Anteil der Zeit wider, in der sich ein Knoten auf der Simulationsfläche nicht bewegt, d.h. je kleiner die Pausenzzeit desto größer ist die Mobilität eines Knotens. Für jede Kombination der Parameter wurden 10 verschiedene Mobilitätsmuster herangezogen, um die Ergebnisse zu berechnen, d.h. es wurden 10 unabhängige Simulationsdurchläufe für einen Messpunkt durchgeführt.

### 5.5.1 Bewertungskriterien

In [CM99] werden einige qualitative wie auch quantitative Kriterien für die Evaluierung und Leistungsbewertung von Routingalgorithmen für mobile multi-hop Ad-hoc-Netze vorgeschlagen. Im Folgenden werden die in dieser Arbeit benutzten Bewertungskriterien beschrieben.

#### Zustellrate

Der Anteil der vom Zielknoten empfangenen Datenpakete ist eine der wichtigsten Bewertungskriterien für Routingalgorithmen. Andere Dienste und Protokolle, die in dem Protokollstapel über dem Routingalgorithmus sind, erfahren die Qualität des Netzes durch diese Größe. Je höher der Anteil der empfangenen Datenpakete ist, desto mehr Daten wurden vom Zielknoten empfangen. Der Anteil der vom Zielknoten empfangenen Datenpakete wird mit Zustellrate bezeichnet und berechnet sich wie folgt:

$$\text{Zustellrate} = \frac{\sum \text{Empfangene Datenpakete}}{\sum \text{Gesendete Datenpakete}}$$

#### Pfadoptimalität

Mit der Pfadoptimalität kann ein Routingalgorithmus auf seine Fähigkeit, den kürzesten Pfad zwischen Quell- und Zielknoten zu finden, bewertet werden.

Da in mobilen multi-hop Ad-hoc-Netzen die Netztopologie ständigen Veränderungen unterworfen ist, und die Routingalgorithmen nur auf ihren eigenen Informationen basierend einen Pfad zwischen Quell- und Zielknoten berechnen müssen, gibt die Pfadoptimalität Auskunft über die Anpassbarkeit eines Routingalgorithmus.

Für dieses Bewertungskriterium wird für jedes Datenpaket die Länge des tatsächlich verwendeten Pfades mit der Länge des kürzesten Pfades zwischen Quell- und Zielknoten verglichen. Die Information über den kürzesten Pfad wird vom Simulationstool zur Verfügung gestellt. Die Pfadabweichung für Paket  $i$  wird wie folgt berechnet:

$$\text{Pfadabweichung}(i) = \text{Länge}(\text{Pfad}_{\text{tatsächlich}}(i)) - \text{Länge}(\text{Pfad}_{\text{optimal}}(i))$$

Ein Histogramm über die Pfadabweichung gibt Aufschluss über den Anteil der Datenpakete, die mit einer bestimmten Pfadabweichung zum Zielknoten übertragen wurden.

## Routingaufwand

Der Routingaufwand gibt Aufschluss über die Kosten eines bestimmten Routingalgorithmus. Dieses Bewertungskriterium ist wichtig, da die zur Verfügung stehende Bandbreite in mobilen multi-hop Ad-hoc-Netzen sehr beschränkt ist und deshalb diese wichtige Ressource schonend genutzt werden muss. Ein hoher Routingaufwand schlägt sich in einer verminderten Übertragung von Datenpaketen, d.h. der Zustellrate, nieder. Der Routingaufwand ergibt sich aus dem Verhältnis der benutzen Routingbytes und den übertragenen Datenbytes:

$$\text{Routingaufwand} = \frac{\sum \text{Routingbytes}}{\sum \text{Datenbytes}}$$

Zur Berechnung des Routingaufwandes werden alle Bytes gezählt, die für die Verfügungstellung der Routingfunktion benötigt werden. Hierzu gehören Routingpakete, die keine Nutzdaten enthalten und Routinginformationen, die in Datenpaketen mitgeführt werden.

## Routingfehler

Ein Pfadfehler tritt ein, wenn während der Datenkommunikation die Verbindung zwischen zwei benachbarten Knoten, die auf dem Pfad liegen, abbricht.

Die Ursache für Pfadfehler ist die Knotenmobilität in mobilen multi-hop Ad-hoc-Netzen. Die Anzahl der Pfadfehler hängt sehr stark vom Mobilitätsverhalten der Knoten ab. Die Anzahl der Pfadfehler von unterschiedlichen Routingalgorithmen bei gleichen Mobilitätszenarien hängt wiederum von den verwendeten Pfaden ab. Deshalb eignet sich dieses Kriterium weniger für den Vergleich von unterschiedlichen Routingalgorithmen. In dieser Arbeit wird dieses Bewertungskriterium nur für den Vergleich der unterschiedlichen Varianten des Ameisenroutingalgorithmus benutzt.

### Ende-zu-Ende-Verzögerung

Bei der Übertragung von Multimediadaten spielt die Verzögerung eine wichtige Rolle. Bei der Audioübertragung, z.B. bei einem Telefongespräch, ist die Akzeptanz maßgeblich von der Verzögerung abhängig, da nach Überschreiten einer Grenze die Verständlichkeit der Sprache nicht mehr möglich ist. Die Ende-zu-Ende-Verzögerung des Datenpaketes  $i$  berechnet sich wie folgt:

$$\text{Verzögerung}(i) = \text{Empfangszeit}(i) - \text{Sendezzeit}(i)$$

### Jitter

Der Jitter gibt die Abweichung von Datenpaketen von der Ende-zu-Ende-Verzögerung an. Diese Größe ist bei der Wiedergabe von Multimediadaten wie Musik und Video wichtig, da bei einer Fehleinschätzung die Puffer leerlaufen können, und dadurch die Wiedergabe abbricht. Der Jitter wird durch unterschiedlich lange Pfade und die Auslastung von Warteschlangen bei den Zwischenknoten verursacht. Der Jitter des Pakets  $i$  wird gemäß der geglätteten Formel aus [SCFJ01] wie folgt berechnet:

$$\text{Jitter}(i) = \text{Jitter}(i-1) + \frac{|\text{Verzögerung}(i) - \text{Verzögerung}(i-1)| - \text{Jitter}(i-1)}{16}$$

#### 5.5.2 Vergleich verschiedener Varianten von ARA

Um die Entwicklung von ARA darzustellen und den Einfluss der einzelnen Module zu beschreiben, wird nachfolgend die Leistung des Ameisenroutingalgorithmus mit seinen unterschiedlichen Varianten und Modulen diskutiert. Insbesondere wird die Leistung des Algorithmus über seine Entwicklungsstadien verfolgt. In den Graphen in diesem Abschnitt ist die Basisversion des Ameisenroutingalgorithmus mit ANT bezeichnet. Die Bezeichnung ARA

wird für die Versionen mit bestimmten Erweiterungen verwendet, die jeweils kurz beschrieben werden.

### ANT vs. ARA

In Abbildung 5.15 ist die Leistung von ANT und  $ARA_{max}$  gegenübergestellt.  $ARA_{max}$  ist die erweiterte Version von ANT mit der besseren Nachahmung von realen Ameisenkolonien. Jedoch wird die Entscheidung für den nächsten Knoten wie bei ANT gefällt, d.h. ein Datenpaket wird an den Nachbarknoten weitergeleitet, der den höchsten Pheromonwert für den Zielknoten besitzt.

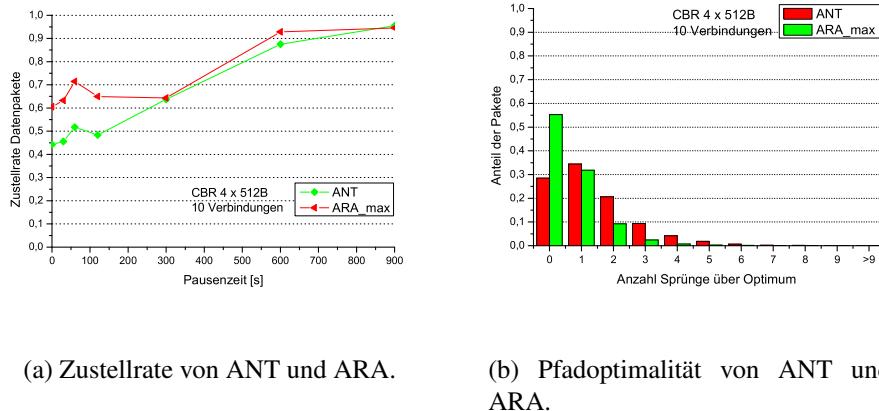
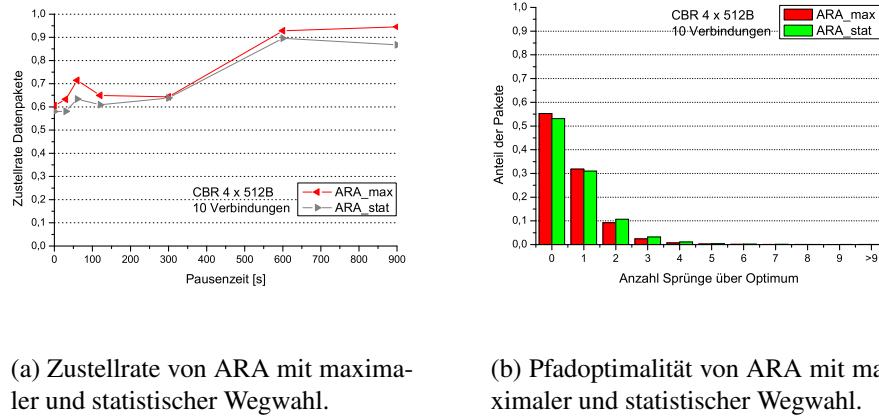


Abbildung 5.15: Leistung von ANT und ARA als Funktion der Pausenzeit bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s.

Abbildung 5.15(a) zeigt die Zustellrate von ANT und ARA als Funktion der Pausenzeit. Die Kurven stellen den Anteil der vom Zielknoten empfangenen Datenpakete im Verhältnis zu den gesendeten Datenpaketen während der gesamten Simulationszeit von 900 Sekunden dar. In den Simulationsszenarien mit wenig Mobilität ist die Zustellrate von ANT und ARA sehr ähnlich. Dies ist auch nicht weiter verwunderlich, da bei statischen Szenarien der Verlust von Paketen nur durch Überlauf von Warteschlangen in den Knoten verursacht wird. Dies zeigt sich auch im Ergebnis der Szenarien mit 900 Sekunden Pausenzeit, wo nicht immer eine Zustellrate von 100% erreicht wird. In Abbildung 5.15(b) ist die Pfadoptimalität für die gleichen Szenarien von beiden Varianten des Ameisenroutingalgorithmus dargestellt. Der Vorsprung von  $ARA_{max}$  wird durch die Pfadoptimalität bestätigt. ANT transportiert die meisten Pakete über einen Pfad, der um einen Sprung länger ist als der optimale Weg zwischen Quell- und Zielknoten. Im Gegensatz dazu ist  $ARA_{max}$  durch die bessere Nachahmung von Ameisenkolonien in der Lage, die meisten Datenpakete über einen Pfad mit optimaler Länge zu transportieren.

## Einfluss des Multipfad routings

Abbildung 5.16 stellt die Leistung von  $\text{ARA}_{\max}$  und  $\text{ARA}_{\text{stat}}$  gegenüber. In  $\text{ARA}_{\text{stat}}$  werden Datenpakete nicht mehr an den Nachbarknoten mit dem höchsten Pheromonwert weitergeleitet. Die Wahl des Nachbarknotens hängt vielmehr von der Verteilung der Wahrscheinlichkeiten  $p_{d,j}^i$  ab. Die Zustellrate von  $\text{ARA}_{\text{stat}}$  liegt zwischen den Zustellraten von ANT und  $\text{ARA}_{\max}$ , jedoch näher bei  $\text{ARA}_{\max}$  (siehe Abbildung 5.16(a)). In Abbildung 5.16(b), welche die Pfadoptimalität darstellt, ist leicht ersichtlich, woran das liegt.  $\text{ARA}_{\text{stat}}$  verteilt die Datenpakete auf mehrere optimale und suboptimale Wege in Richtung des Zielknotens.



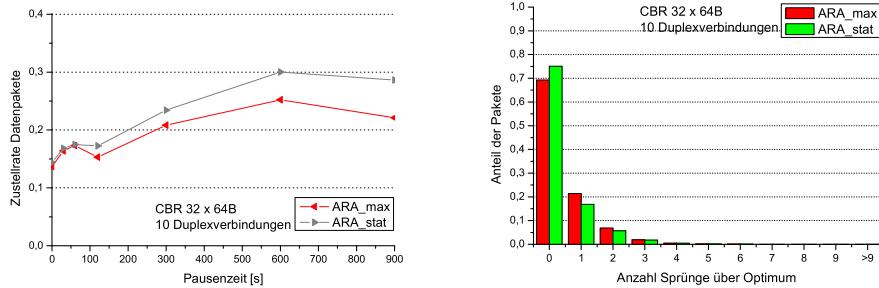
(a) Zustellrate von ARA mit maximaler und statistischer Wegwahl.

(b) Pfadoptimalität von ARA mit maximaler und statistischer Wegwahl.

Abbildung 5.16: Leistung von ARA mit statistischer und maximaler Wegwahl als Funktion der Pausenzeit bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s.

Man erwartet durch die Lastverteilung auf mehrere Pfade eine bessere Leistung von  $\text{ARA}_{\text{stat}}$ . Dies trifft tatsächlich auch zu. In Abbildung 5.17 wird die Leistung von  $\text{ARA}_{\max}$  und  $\text{ARA}_{\text{stat}}$  unter höherer Last dargestellt. Die Ergebnisse zeigen die Leistung bei einem Duplex-Verkehr von 32 Paketen pro Sekunde mit je 64 Byte, d.h. die Gesamtanzahl der gesendeten Daten eines Quellknotens ist zwar gleich, jedoch ist die Senderate nun viel höher. Da hier Duplex-Verkehr betrachtet wird, bedeuten 10 parallele Verbindungen in beide Richtungen eine Gesamtanzahl von 20 Verbindungen im Netz.

Aus Abbildung 5.17(a) ist ersichtlich, dass durch die Verteilung der Datenpakete auf mehrere Wege die Zustellrate von  $\text{ARA}_{\text{stat}}$  in den meisten Szenarien höher ist als die von  $\text{ARA}_{\max}$ . Das Ergebnis wird durch Abbildung 5.17(b) bestätigt. Die meisten Datenpakete werden von beiden Varianten über einen optimalen Pfad transportiert, jedoch fällt der Anteil bei  $\text{ARA}_{\text{stat}}$  höher aus.



(a) Zustellrate von ARA mit maximaler und statistischer Wegwahl.

(b) Pfadoptimalität von ARA mit maximaler und statistischer Wegwahl.

Abbildung 5.17: Leistung von ARA mit statistischer und maximaler Wegwahl als Funktion der Pausenzeit bei 10 CBR-Duplex-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s.

### Der Sendepuffer von ARA

Die bisher diskutierte Leistung des Ameisenroutingalgorithmus ist für Szenarien mit wenig Mobilität akzeptabel. Die Leistung in hoch-mobilen Szenarien fällt jedoch stark ab. Untersuchungen haben gezeigt, dass die schwache Leistung des Ameisenroutingalgorithmus in hoch-mobilen Szenarien nicht von den Eigenschaften des Algorithmus selbst herührt, sondern von der Umsetzung. Ein großes Problem war die Verwendung des ersten Datenpakets einer Verbindung als FANT. Da Datenpakete die maximal erlaubte Größe haben können, erfordert die Übertragung einer großen FANT im Netz entsprechend viel Zeit.

Ein weiteres Problem war, dass, wenn während der FANT und BANT unterwegs waren, weitere Datenpakete zur Übertragung von der Transportschicht ankamen, durch Mangel eines Pfades diese auch als FANT im Netz geflutet wurden. Dieser Effekt zeigte sich besonders in hoch-mobilen Szenarien, da durch die Knotenmobilität die Anzahl der Pfadfindungsphasen ansteigt. Die Umsetzung des Ameisenroutingalgorithmus wurde so verändert, dass nur noch ein minimales Paket als FANT verwendet wird. Als weitere Verbesserung bekam der Ameisenroutingalgorihmus einen Sendepuffer, in dem anstehende Datenpakete abgelegt werden.

In Abbildung 5.18 wird die Leistung von ARA<sub>stat</sub> und ARA<sub>stat\_sb</sub>, d.h. statistische Wegewahl mit eingeschaltetem Sendepuffer, gegenübergestellt. Es ist offensichtlich, dass durch die beschriebenen Verbesserungen bei der Umsetzung, die Leistung des Ameisenroutingalgoritmus auch in hoch-mobilen

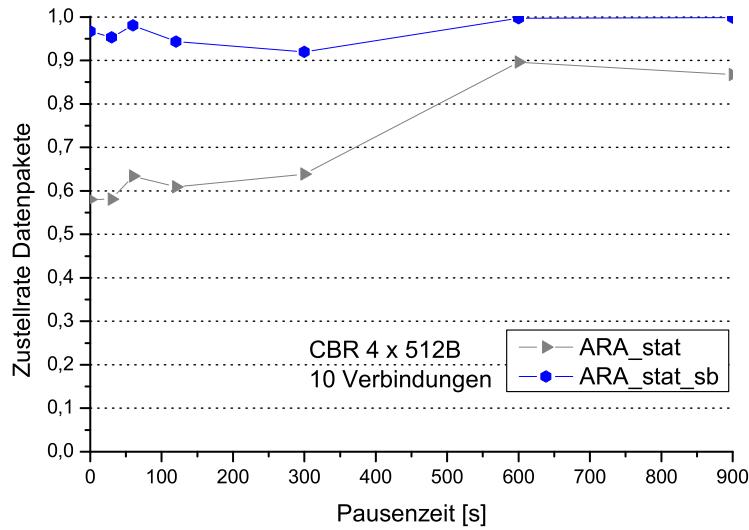


Abbildung 5.18: Zustellrate von ARA mit Sendepuffer bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s.

Szenarien in einen sehr guten Bereich gestiegen ist. Das Netz wird nun nicht mehr durch die nachrückenden Datenpakete, die als FANT geflutet werden, überlastet. In Szenarien mit Pausenzeit von 900 Sekunden ist die Zustellrate fast immer 100%. In den hoch-mobilen Szenarien fluktuiert die Zustellrate zwar ein wenig, jedoch liegt sie in fast allen Fällen über 95%.

### Einfluss des MAC-Tap

Die Zustellrate und der Routingaufwand von ANT, ARA<sub>stat\_sb</sub> und ARA als Funktion der Pausenzeit ist in Abbildung 5.19 dargestellt. Hier steht ARA nun für die *Vollversion des Ameisenroutingalgorithmus* mit allen vorgestellten Erweiterungen. In ARA ist zusätzlich zum Sendepuffer nun auch der MAC-Tap aktiv. Mit dem MAC-Tap versucht ein Knoten durch mithören der Kommunikation seiner Nachbarn, Informationen für das Routing zu extrahieren und diese zu optimieren.

Abbildung 5.19(a) stellt die Zustellrate von ANT und den beiden Varianten von ARA dar. Durch den Einsatz des MAC-Taps steigt die Leistung des Ameisenroutingalgorithmus in hoch-mobilen Szenarien weiter an und entspricht etwa der Zustellrate der Szenarien mit 900 s Pausenzeit.

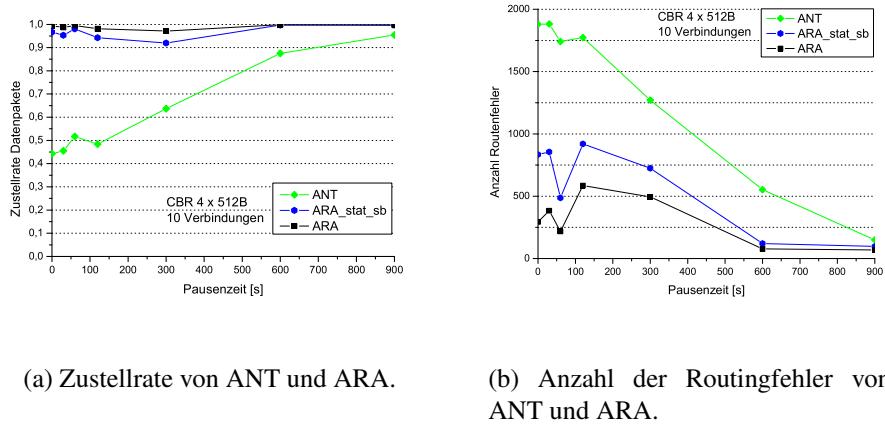


Abbildung 5.19: Leistung von ANT und ARA als Funktion der Pausenzeit bei 10 CBR-Verbindungen und einer Maximalgeschwindigkeit von 10 m/s.

Die Abbildung 5.19(b) zeigt für alle drei Varianten des Ameisenroutingalgorithmus die Anzahl der Routingfehler als Funktion der Pausenzeit. Das Ergebnis ist nicht weiter verwunderlich. Die Anzahl der Routingfehler nimmt mit Abnahme der Mobilität in allen Varianten stark ab. Die Anzahl der Routingfehler ist bei der Basisversion in den hoch-mobilen Szenarien etwa 4 mal höher als bei der Vollversion. Der Unterschied zwischen ARA<sub>stat\_sb</sub> und ARA hinsichtlich der Anzahl der Routingfehler ist etwa ein Faktor von 2, welcher sich mit Abnahme der Mobilität verringert.

In Abbildung 5.20 ist die Pfadoptimalität von ANT und ARA in der Vollversion gegenübergestellt. ANT transportiert die Mehrzahl der Datenpakete über einen Pfad, der um einen Sprung länger war als der kürzeste Weg. ARA transportiert etwa 85% der Datenpakete über einen kürzesten Weg und etwa 10% der Datenpakete über einen Pfad mit einem Sprung mehr. ARA kann die Fähigkeit der Ameisenkolonien sehr gut ausnutzen, um den kürzesten Pfad zwischen dem Quell- und Zielknoten zu finden. Dabei wird der Datenverkehr auf mehrere Pfade verteilt, wodurch eine weitere Verbesserung der Leistung erzielt wird.

Abbildung 5.21 stellt die Zustellrate von ANT und ARA mit  $\alpha = 0,05$ -Konfidenzintervall dar. Aus den Graphen ist gut zu erkennen, dass die Ergebnisse beider Varianten in einem schmalen Bereich liegen und daher auf zuverlässige Ergebnisse schließen lassen.

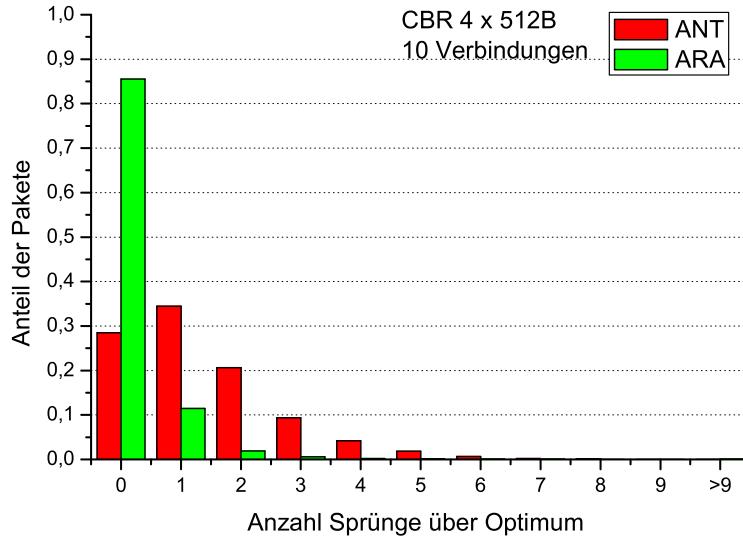


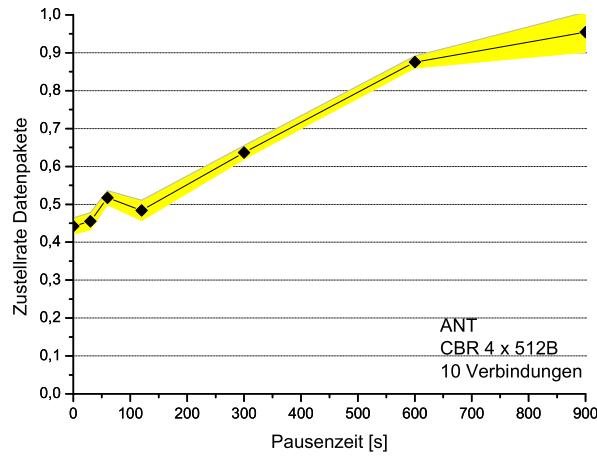
Abbildung 5.20: Routingoptimalität von ANT und ARA.

### 5.5.3 Übertragung von Strömen mit konstanter Datenrate

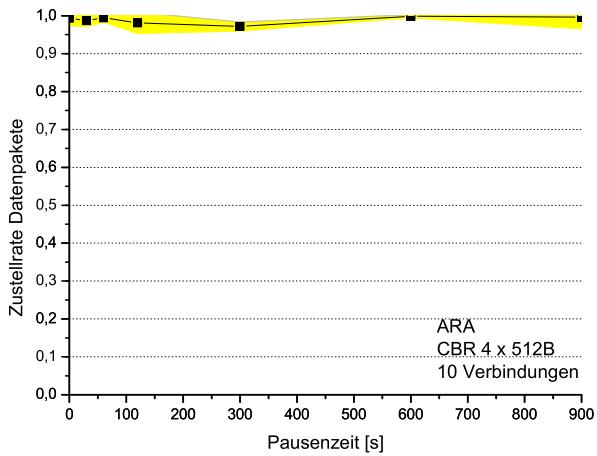
Bisher wurde die Leistungsfähigkeit des Ameisenroutingalgorithmus betrachtet. Ferner wurde auf die Leistungsverbesserung durch den Einsatz von Erweiterungen, bis hin zur aktuellen Vollversion, eingegangen. Die bisherige Betrachtung erlaubt jedoch keinen Einblick, wie die Leistung des Ameisenroutingalgorithmus im Vergleich zu existierenden Routingalgorithmen für mobile Ad-hoc-Netze ist. Deshalb wird in diesem Abschnitt die Leistung des Ameisenroutingalgorithmus zwei bekannten Vertretern – AODV und DSR – gegenübergestellt. Der Ameisenroutingalgorithmus wird ab jetzt in der Vollversion mit allen vorgestellten Erweiterungen verwendet und mit ARA bezeichnet.

#### Zustellrate

Die Abbildung 5.22 zeigt die Zustellrate von AODV, DSR und ARA als Funktion der Pausenzeit. Die Graphen sind die Ergebnisse aus Simulationen mit 10 parallelen Verbindungen mit vier Paketen pro Sekunde, wobei ein Paket eine Größe von 512 Byte besitzt. Die Ergebnisse aller drei Verfahren sind in den Szenarien mit wenig Mobilität – bei einer Pausenzeit von 600 und 900



(a) Zustellrate von ANT.



(b) Zustellrate von ARA.

Abbildung 5.21: Zustellrate von ANT und ARA mit  $\alpha = 0,05$ -Konfidenzintervall.

Sekunden – sehr ähnlich und liegen zwischen 99% und 100%. Die Unterschiede zeigen sich erst in den Szenarien mit mehr Mobilität und insbesondere in hoch-mobilen Szenarien. Mit Zunahme der Mobilität nimmt die Zustellrate aller Verfahren ab. Jedoch sind AODV und ARA in der Lage, ihre Leistungen auch in den Szenarien mit mehr Mobilität aufrechtzuhalten. Die Leistung

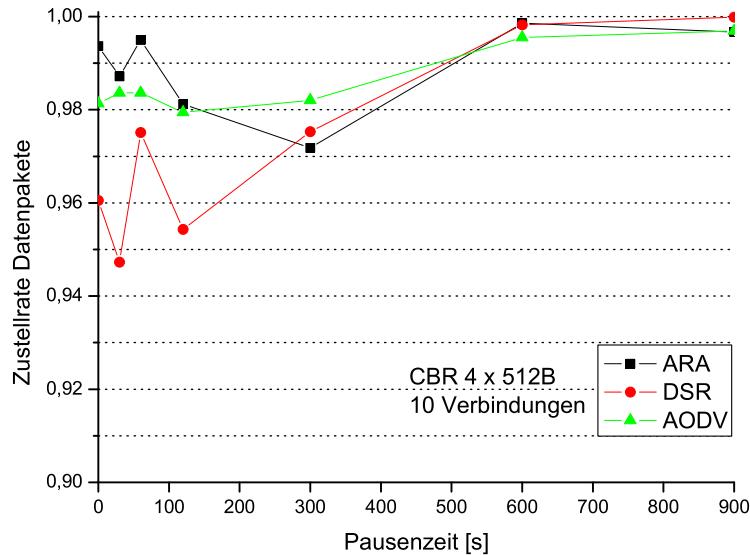
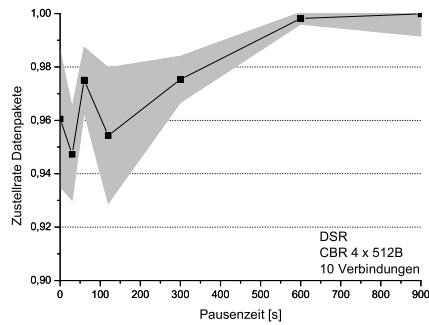


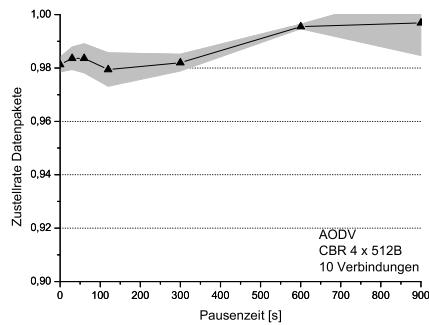
Abbildung 5.22: Zustellrate von AODV, DSR und ARA als Funktion der Pausenzeit bei 10 CBR-Verbindungen.

von DSR fällt mit Zunahme der Mobilität immer weiter ab. In den Szenarien mit einer Pausenzeit von 0 Sekunden zeigen sich die Unterschiede am stärksten. Woran liegt die schlechtere Leistung von AODV und DSR in den hochmobilen Szenarien? Als erstes ist anzumerken, dass in beiden Verfahren nicht versucht wird das Auftreten von Routingfehlern, lokal zu lösen, sondern den Quellknoten zu informieren. Wenn die Information beim Quellknoten angekommen ist, müssen beide Verfahren versuchen, einen Pfad zum Zielknoten zu finden. Bei DSR resultiert dies im Fluten des Netzes. Bei AODV müssen unter Umständen zuerst die Routingtabellen aktualisiert werden. Der Ameisenroutingalgorithmus kann hier durch seine Multipfadrouting-Fähigkeit ein wenig an Leistung herausholen, da ein Knoten beim Auftreten eines Routingfehlers versucht, den Zielknoten über einen zweiten Pfad zu erreichen. Dieser zweite Versuch schlägt sich in den hoch-mobilen Szenarien positiv nieder.

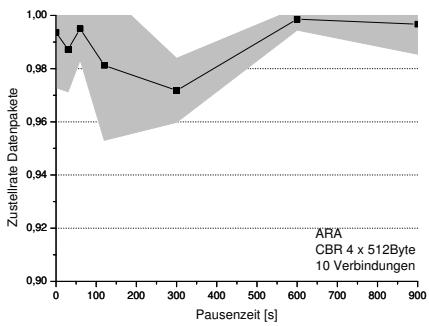
Abbildung 5.23 stellt die Zustellraten aller drei Verfahren mit  $\alpha = 0,05$ -Konfidenzintervall dar. Als erstes fällt auf, dass die Ergebnisse von AODV (siehe Abbildung 5.23(b)) in einem kleineren Bereich liegen, als die Ergebnisse von DSR und ARA. Das Intervall, in dem die Ergebnisse von DSR liegen, ist in den Szenarien mit wenig Mobilität klein und wächst mit Zunahme der Mobilität. In den Szenarien mit einer Pausenzeit von 0 Sekunden schwanken die



(a) Zustellrate von DSR.



(b) Zustellrate von AODV.



(c) Zustellrate von ARA.

Abbildung 5.23: Zustellrate von AODV, DSR und ARA mit  $\alpha = 0,05$ -Konfidenzintervall.

Ergebnisse zwischen 93% und 99% (siehe Abbildung 5.23(a)). Die Ergebnisse des Ameisenroutingalgorithmus sind in Abbildung 5.23(c) dargestellt, die identisch mit der Abbildung 5.21(b) ist. Nur die Skalierung ist anders gewählt, um die Vergleichbarkeit mit AODV und DSR zu gewährleisten. Die Ergebnisse des Ameisenroutingalgorithmus sind verteilter als die von AODV, liegen jedoch immer über 95% und somit sehr oft über der Leistung von DSR.

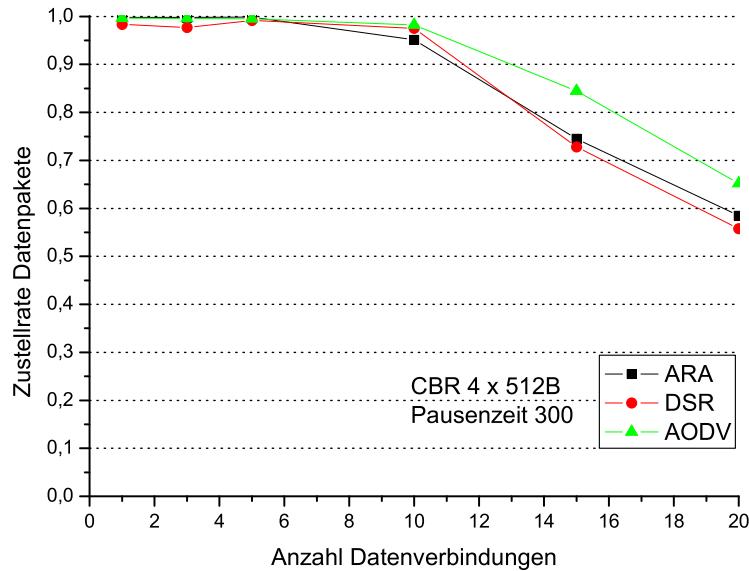


Abbildung 5.24: Zustellrate von AODV, DSR und ARA als Funktion der Anzahl paralleler Verbindungen bei einer Pausenzeit von 300 Sekunden.

## Skalierbarkeit

Um die Skalierbarkeit der drei Verfahren zu untersuchen, wurden Simulationen mit wachsender Anzahl von Verbindungen durchgeführt. Abbildung 5.24 stellt die Zustellrate der drei Verfahren als Funktion der parallelen Verbindungen von 1 bis 20 dar. Die Simulationen wurden bei mittlerer Mobilität der Knoten, d.h. 300 Sekunden Pausenzeit, durchgeführt. Für alle Verfahren gilt, dass mit Zunahme der parallelen Verbindungen die Zustellrate abnimmt. Es gibt jedoch einen kritischen Wert bei der Anzahl der parallelen Verbindungen, der bei 10 liegt. Ab 10 parallelen Verbindungen zeigen sich Unterschiede zwischen den drei Verfahren. AODV zeigt durchgehend eine höhere Zustellrate als DSR und ARA. Die Leistung von ARA liegt zumeist über der von DSR.

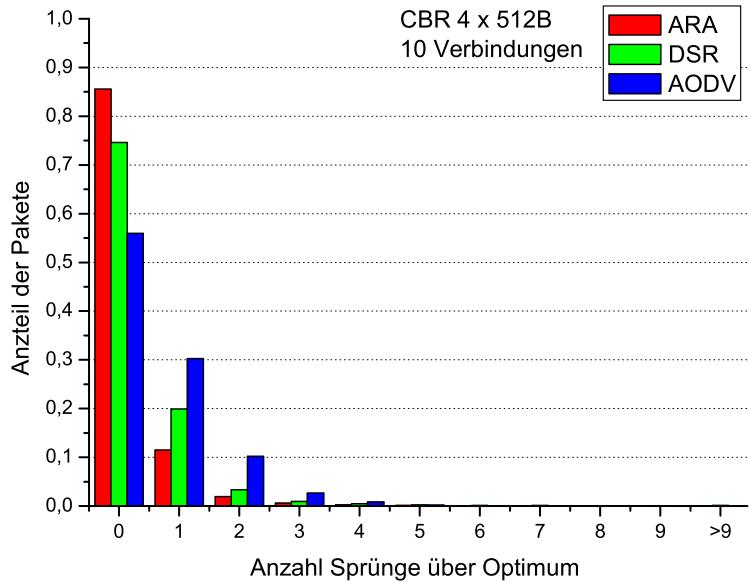
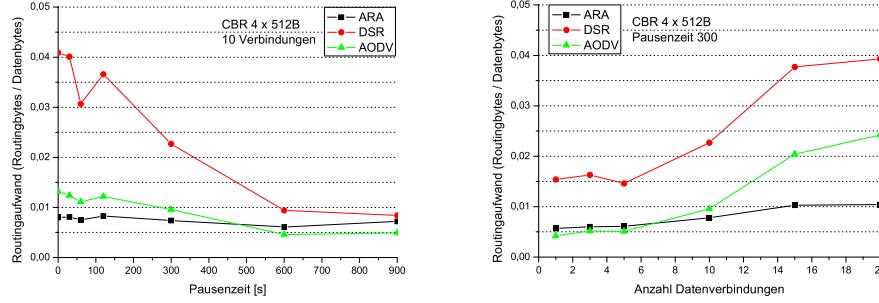


Abbildung 5.25: Pfadoptimalität von AODV, DSR und ARA bei 10 CBR-Verbindungen.

## Pfadoptimalität

Als nächstes werden die drei Verfahren hinsichtlich der Pfadoptimalität gegenübergestellt. Abbildung 5.25 zeigt die Pfadoptimalität für alle drei Verfahren. Alle drei Routingverfahren transportieren die meisten Datenpakete über einen optimalen Pfad, jedoch ist der Anteil unterschiedlich. AODV transportiert etwa 55%, DSR 75% und ARA 85% aller Datenpakete über einen optimalen Pfad. Die Ursache dieses Ergebnisses ist, dass DSR und ARA bemüht sind, die schon bekannten Pfade zu verbessern. Im Gegensatz dazu verwendet AODV einen Pfad, bis dieser nicht mehr verwendet werden kann. Obwohl diese Vorgehensweise sich in der Benutzung von längeren Pfaden zeigt, wird die Zustellrate dadurch nicht negativ beeinflusst. Beim Ameisenroutingalgorithmus zeigt sich die sehr gute Pfadoptimalität bei der Zustellrate in hoch-mobilen Szenarien, wo er am besten abschneidet. Dies ist aber auch auf die Multipfadrouting-Fähigkeit zurückzuführen, da hierdurch die Knoten sehr oft mehrere Pfade mit optimaler Länge zum Zielknoten besitzen. Bei einem Routingfehler wirkt sich zusätzlich der Versuch, das Datenpaket über einen zweiten Pfad zu transportieren, sehr positiv aus.

## Routingaufwand



(a) Routingaufwand als Funktion der Pausenzeit.

(b) Routingaufwand als Funktion der Anzahl paralleler Verbindungen.

Abbildung 5.26: Routingaufwand von AODV, DSR und ARA als Funktion der Pausenzeit und der Anzahl paralleler Verbindungen.

In Abbildung 5.26 werden die drei Routingverfahren hinsichtlich des Routingaufwands gegenübergestellt. In Abbildung 5.26(a) ist der Routingaufwand als Funktion der Pausenzeit dargestellt. Der Routingaufwand ergibt sich als Verhältnis von benötigten Routingbytes zu transportierten Datenbytes. Aus den Ergebnissen ist zu erkennen, dass DSR den höchsten Unterschied zwischen DSR und den beiden anderen Routingverfahren ist, dass DSR in jedes Datenpaket den vollständigen Pfad vom Quellknoten zum Zielknoten einfügen muss. Durch die erneute Pfadsuche wird auch der Routingaufwand erhöht. Die Ergebnisse von AODV und ARA liegen nahe beieinander, wobei in den hochmobilen Szenarien AODV einen höheren Routingaufwand besitzt als ARA. Der Routingaufwand von ARA ist über alle Szenarien nahezu konstant. Dies liegt daran, dass ARA bei der Pfadsuche immer einen Forward-Ant und einen Backward-Ant einsetzt, wodurch der Routingaufwand bestimmt ist.

Die Abbildung 5.26(b) zeigt den Routingaufwand der drei Verfahren als Funktion der Anzahl paralleler Verbindungen. Die Ergebnisse wurden durch Simulationen mit einer mittleren Mobilität der Knoten, d.h. bei einer Pausenzeit von 300 Sekunden, erhalten. Erwartungsgemäß wächst der Routingaufwand mit Zunahme der parallelen Verbindungen. Die Entwicklung zeigt sich jedoch für die drei Verfahren sehr unterschiedlich. Der Routingaufwand von DSR ist wieder höher als der Routingaufwand von AODV und ARA, und wächst drastisch mit zunehmender Anzahl der Verbindungen. Der Routingaufwand von AODV und ARA ist bis 10 Verbindungen in etwa gleich. Ab 10 parallelen Verbindungen steigt der Routingaufwand von AODV und ist etwa 2,4 mal höher.

her als der Routingaufwand von ARA. Der Routingaufwand von ARA wächst mit zunehmender Anzahl der Verbindungen, ist jedoch im Vergleich zu den beiden anderen Verfahren sehr moderat.

#### 5.5.4 Übertragung von Echtzeitdaten

Ein Echtzeitdatenstrom hat andere Anforderungen an das Netz, als der bisher betrachtete Datenstrom mit konstanter Datenrate. Der Unterschied liegt in der zeitlichen Abhängigkeit von Echtzeitdaten. Beispielsweise wird bei der Übertragung von Sprache eine Verzögerung von mehr als 200 ms von den Kommunikationsteilnehmern als störend empfunden. Bei der Übertragung von Musik und Video kann durch Puffern die Verzögerung teilweise kompensiert werden. Dieser Abschnitt betrachtet die Leistung von AODV, DSR und ARA bei der Übertragung von Echtzeitdaten.

Als Referenzdatenstrom wurde ein typischer Vertreter ausgewählt. Mit einem Datenstrom von 13 kBit/s wird die Sprachübertragung von GSM nachempfunden. Der Schwerpunkt der Untersuchung liegt auf der Skalierbarkeit der Routingalgorithmen, d.h. wie sich ihre Leistung bei steigender Netzlast ändert. Auf der Simulationsfläche bewegen sich 50 Knoten, die jedoch keine Pausen zwischen ihren Bewegungen einlegen. Somit ist ein hoch mobiles Szenario gegeben. Als Vergleichskriterien werden die Zustellrate, die Ende-zu-Ende-Verzögerung und der Jitter herangezogen, die nachfolgend diskutiert werden.

#### Zustellrate

In Abbildung 5.27 ist die Zustellrate von AODV, DSR und ARA bei der Übertragung von Audioströmen dargestellt. In der Abbildung ist die Zustellrate der drei Verfahren als Funktion der Anzahl paralleler Duplex-Verbindungen dargestellt. Dabei übertragen die Knoten 26 Pakete zu je 64 Byte pro Sekunde. Dieser Datenstrom entspricht dem Datenaufkommen eines GSM-Audiostroms mit 13 kBit/s. Um beide Gesprächspartner zu simulieren, wurden Duplex-Verbindungen verwendet, d.h. die Anzahl von 5 Duplex-Verbindungen bedeutet 10 Verbindungen im Netzwerk. Ein wichtiger Unterschied gegenüber den bisher betrachteten Datenströmen ist, dass nun viele kleine Pakete transportiert werden müssen.

Die Graphen aller drei Verfahren zeigen einen sehr ähnlichen Verlauf. Mit zunehmender Anzahl an Verbindungen im Netz nimmt die Zustellrate schnell ab. Die Zustellrate aller drei Verfahren für eine Duplex-Verbindung liegt im Bereich von 98% - 100%. Doch schon bei 3 Duplex-Verbindungen sinkt die Zustellrate unter 90% wobei AODV die beste und DSR die schlechteste Lei-

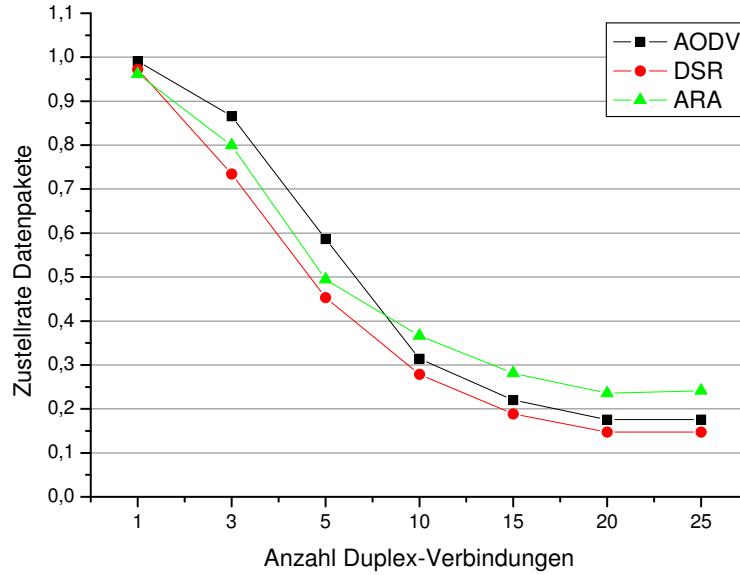
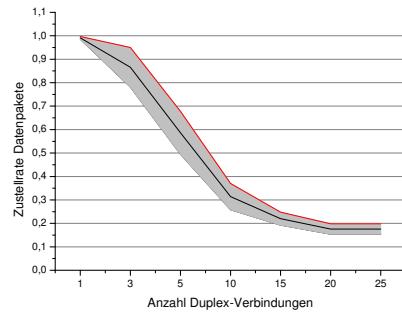


Abbildung 5.27: Zustellrate von AODV, DSR und ARA als Funktion der Anzahl paralleler Audio-Duplex-Verbindungen mit 13 kBit/s in beide Richtungen.

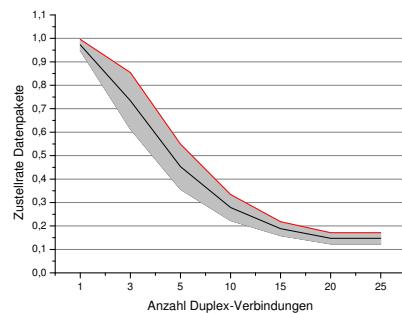
stung zeigt. Die Zustellrate von ARA befindet sich zwischen DSR und AODV. Ab 15 parallelen Duplex-Verbindungen sinkt die Zustellrate aller drei Verfahren bis unter 30% und stabilisiert sich danach. Eine interessante Beobachtung ist, dass ARA ab 10 Duplex-Verbindungen die beste Leistung zeigt und diese auch im weiteren Verlauf beibehält. Bei 25 Duplex-Verbindungen besitzt ARA eine um etwa 10% höhere Zustellrate als DSR und AODV.

Die Abbildung 5.28 zeigt die Zustellraten von AODV, DSR und ARA zusammen mit ihrem  $\alpha = 0,05$ -Konfidenzintervall. Die Ergebnisse liegen in einem kleinen Intervall, was auf zuverlässige Ergebnisse deutet.

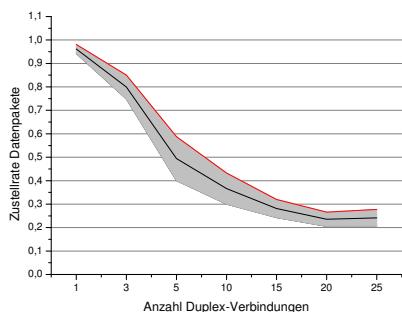
Interessant ist die Tatsache wie sich die Zustellrate bei den drei Routingverfahren, verglichen mit den vorherigen Ergebnissen, verschlechtert hat. Die Ursachen dieser Ergebnisse sind vielfältig. Zum einen scheinen die Knoten mit Duplex-Verbindungen schlechter umgehen zu können. Eine andere Ursache liegt in der vermehrten Anzahl an Verbindungen, die im Netz zu transportieren sind. Wie in Abschnitt 5.5.3 diskutiert, sinkt die Zustellrate aller drei Routingverfahren mit zunehmender Anzahl an Verbindungen. Die doppelte Rolle der Knoten scheint diesen Effekt zu verstärken. Die Beobachtung,



(a) AODV



(b) DSR



(c) ARA

Abbildung 5.28: Zustellrate von AODV, DSR und ARA bei Audioübertragung als Funktion der Anzahl paralleler Duplex-Verbindungen mit  $\alpha = 0,05$ -Konfidenzintervall.

dass ARA mit zunehmender Anzahl an Duplex-Verbindungen seine Leistung gegenüber AODV und DSR verbessern kann, ist auf seine Multipfadrouting-Fähigkeit zurückzuführen. Der Ameisenroutingalgorithmus kann auch in einem sehr stark belasteten Netzwerk durch die Benutzung mehrerer Pfade seine Leistung ein wenig verbessern.

### Verzögerung

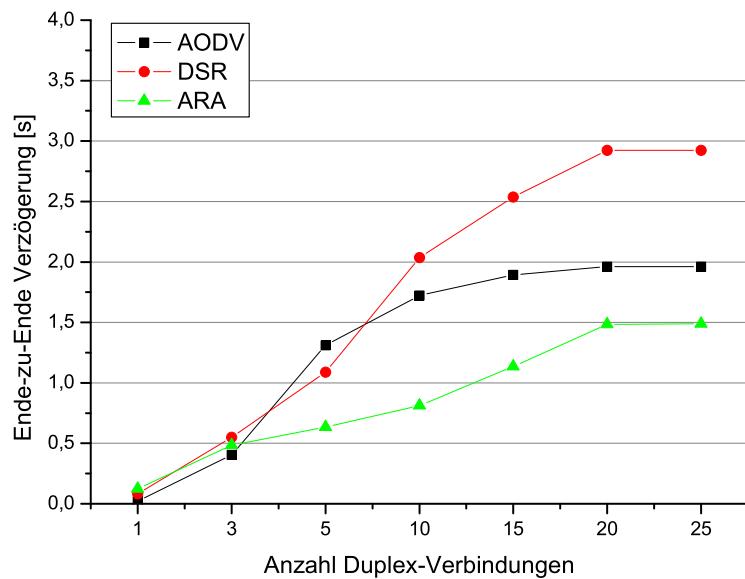
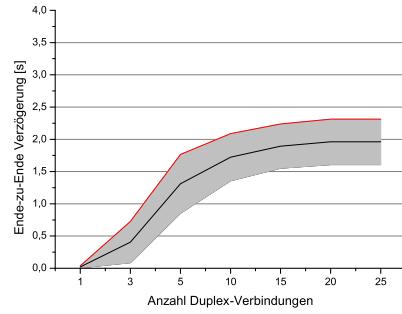


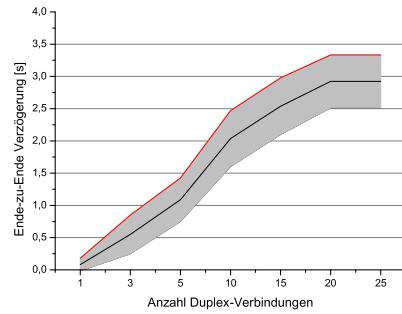
Abbildung 5.29: Ende-zu-Ende Verzögerung von AODV, DSR und ARA als Funktion der Anzahl paralleler Audio-Duplex-Verbindungen mit 13 kBit/s in beide Richtungen.

In Abbildung 5.29 ist die Ende-zu-Ende-Verzögerung bei der Übertragung von Audioströmen mit AODV, DSR und ARA dargestellt. Analog zum letzten Abschnitt ist die Ende-zu-Ende-Verzögerung als Funktion der Anzahl paralleler Duplex-Verbindungen dargestellt. Abbildung 5.30 stellt die Ende-zu-Ende-Verzögerung von AODV, DSR und ARA zusammen mit  $\alpha = 0,05$ -Konfidenzintervall dar. Die Ergebnisse liegen in einem kleinen Intervall, was auf zuverlässige Ergebnisse deutet.

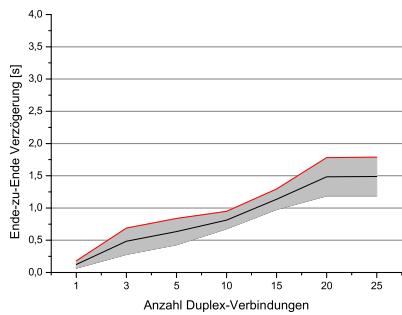
Mit zunehmender Anzahl an Duplex-Verbindungen nimmt die Ende-zu-Ende-Verzögerung der Audioverbindungen deutlich zu. Für eine Duplex-Verbin-



(a) AODV



(b) DSR



(c) ARA

Abbildung 5.30: Ende-zu-Ende Verzögerung von AODV, DSR und ARA bei Audioübertragung als Funktion der Anzahl paralleler Duplex-Verbindungen mit  $\alpha = 0,05$ -Konfidenzintervall.

dung befindet sich die Ende-zu-Ende-Verzögerung unter 200 ms, bei 3 Duplex-Verbindungen bewegt sie sich um 0,5 s. Ab 5 Duplex-Verbindungen liegt die Ende-zu-Ende-Verzögerung von AODV und DSR über 1 s und der von ARA bei 0,6 s. Im weiteren Verlauf steigt die Ende-zu-Ende-Verzögerung bei DSR bis 3 s, bei AODV bis 2 s und bei ARA bis 1,5 s. Die bessere Leistung von ARA ist wiederum auf seine Multipfadrouting-Fähigkeit zurückzuführen, die sich auch bei der Zustellrate gezeigt hat.

### Jitter

Der Jitter gibt die Abweichung der Zwischenankunftszeiten einer Verbindung an und ist ein wichtiger Messwert, da hierdurch die Größe von Puffern beeinflusst wird. Um ein kontinuierliches Abspielen von Echtzeitdaten zu gewährleisten muss der Zielknoten die Daten so puffern, dass durch den Jitter der Puffer nicht leer- oder überläuft.

In Abbildung 5.31 ist der Jitter für AODV, DSR und ARA als Funktion der Anzahl paralleler Duplex-Verbindungen dargestellt. Abbildung 5.32 stellt den Jitter von AODV, DSR und ARA zusammen mit  $\alpha = 0,05$ -Konfidenzintervall dar. Die Ergebnisse liegen in einem kleinen Intervall, was auf zuverlässige Ergebnisse deutet.

Die Ergebnisse der drei Verfahren liegen sehr nah beieinander, wobei DSR mit zunehmender Anzahl der Duplex-Verbindungen sich von AODV und ARA absetzt und einen höheren Jitter aufzeigt. Der Ameisenroutingalgorithmus kann ähnlich der Zustellrate und der Ende-zu-Ende-Verzögerung durch seine Multipfadrouting-Fähigkeit eine etwas bessere Leistung als AODV und DSR erzielen.

### 5.5.5 Übertragung von TCP-Strömen

Bisher wurde für den Leistungsvergleich nur Verbindungen mit konstanter Senderate verwendet. Jedoch zeigen Untersuchungen, dass der Hauptteil des Kommunikationsverkehrs im Internet das verbindungsorientierte und zuverlässige Transportkontrollprotokoll (TCP) verwendet. Deshalb ist zu erwarten, dass der Kommunikationsverkehr in mobilen multi-hop Ad-hoc-Netzen ähnlich sein wird. In diesem Abschnitt wird die Leistung des Ameisenroutingalgoritmus unter einem typischen TCP-Verkehr betrachtet. Als Vergleichsverfahren werden wiederum AODV und DSR herangezogen.

Die Ergebnisse dieses Abschnitts stammen aus Simulationen mit 5 parallelen TCP-Verbindungen. Die Knoten bewegen sich auf einer Fläche von 1500 m  $\times$  300 m mit einer Maximalgeschwindigkeit von 10 m/s und einer Pausenzeit

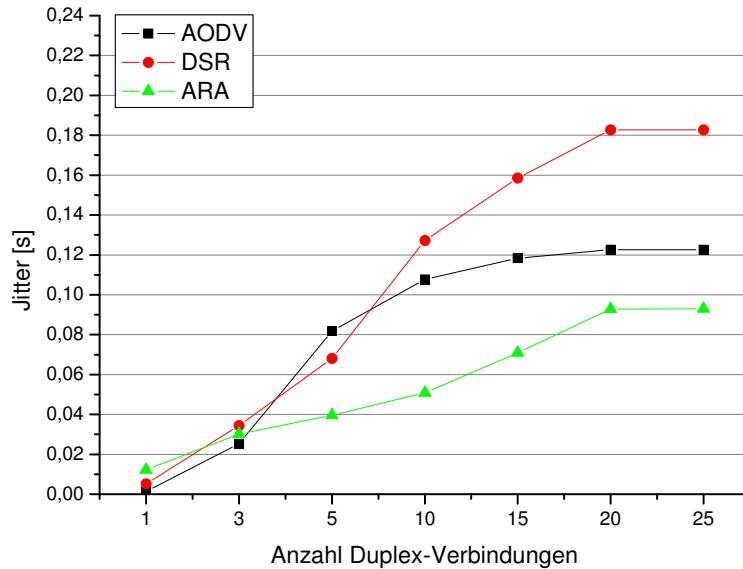
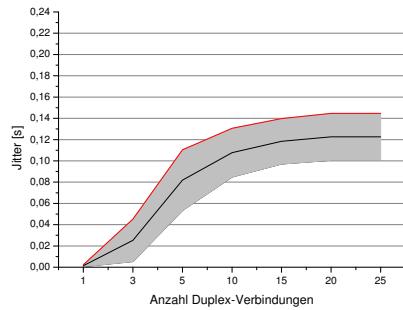


Abbildung 5.31: Jitter von AODV, DSR und ARA als Funktion der Anzahl paralleler Audio-Duplex-Verbindungen mit 13 kBit/s in beide Richtungen.

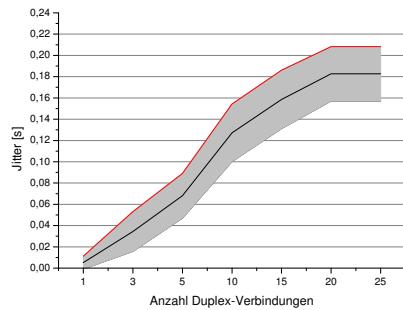
von 1 Sekunde. Für die Ergebnisse wurden 10 unabhängige Simulationsdurchläufe herangezogen.

In Abbildung 5.33 ist der Durchsatz von AODV, DSR und ARA als Funktion der Simulationszeit dargestellt. Die fünf Verbindungen starten gleichverteilt über die ersten 200 Sekunden. Am Anfang der Simulationszeit starten alle Routingverfahren mit einem hohen Durchsatz, welches durch die Slow-Start-Phase von TCP bedingt ist und passen ihren Durchsatz nach dieser Einschwingphase an. Bemerkenswert ist, dass der Durchsatz von DSR und ARA in der Einschwingphase etwa 15% höher ist als der von AODV. AODV kann seinen Durchsatz über die ganze Simulationszeit in etwa gleicher Höhe halten, wobei er zwischen 100 kBit/s und 120 kBit/s schwankt. Der Durchsatz von DSR liegt am Anfang der Simulationszeit gleichauf mit der von ARA, kann den hohen Durchsatz aber nicht halten und nähert sich immer mehr an AODV an. ARA startet wie DSR mit einem hohen Durchsatz, muss jedoch in der Einschwingphase den Durchsatz durch das Hinzukommen aller fünf Verbindungen anpassen. ARA kann jedoch den hohen Durchsatz über die ganze Simulationszeit halten.

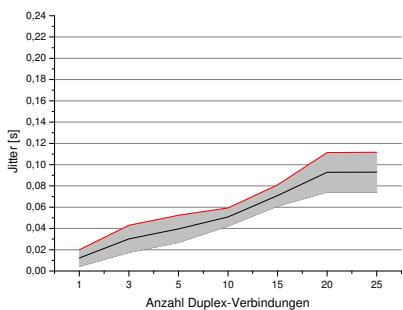
Der Durchsatz von AODV, DSR und ARA als Funktion der Simulationszeit



(a) AODV



(b) DSR



(c) ARA

Abbildung 5.32: Jitter von AODV, DSR und ARA bei Audioübertragung als Funktion der Anzahl paralleler Duplex-Verbindungen mit  $\alpha = 0,05$ -Konfidenzintervall.

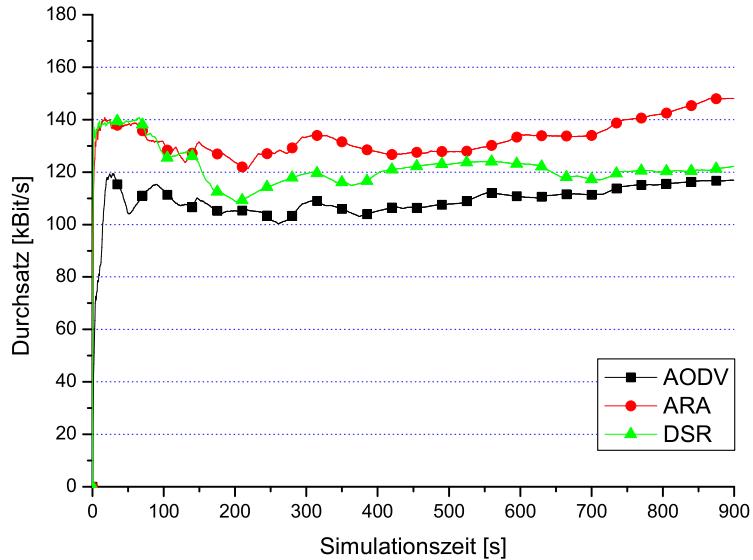


Abbildung 5.33: Durchsatz von AODV, DSR und ARA bei 5 TCP-Verbindungen.

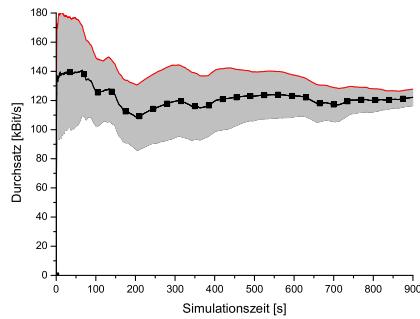
mit  $\alpha = 0,05$ -Konfidenzintervall ist in Abbildung 5.34 dargestellt. Für alle drei Routingverfahren gilt, dass der Durchsatz am Anfang der Simulationszeit eine größere Schwankung zeigt, welche mit zunehmender Zeit immer kleiner wird.

## 5.6 Verwandte Ansätze

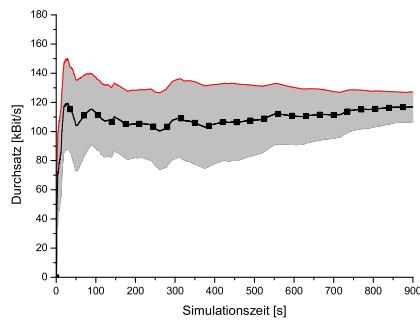
Es gibt eine Vielzahl von Ansätzen, die auf Ameisenalgorithmen basieren. Die Unterschiede zum Ameisenroutingalgorithmus werden nachfolgend diskutiert und aufgezeigt. Am Ende des Abschnitts folgt eine Zusammenfassung der Verfahren.

### 5.6.1 ABC Routing

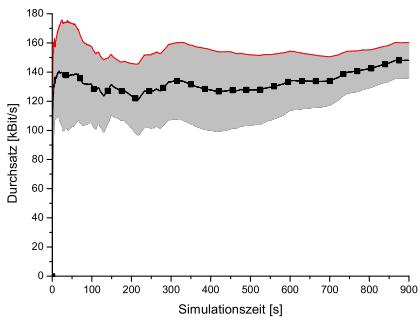
In [SHBR96] wird ein Verfahren – *Ant-Based Control (ABC) for network management* – für die Lastverteilung in leitungsvermittelten Telekommunikationsnetzen vorgestellt, das auf Ameisenalgorithmen basiert. Das Ziel ist, die



(a) Durchsatz von DSR.



(b) Durchsatz von AODV.



(c) Durchsatz von ARA.

Abbildung 5.34: Durchsatz von AODV, DSR und ARA mit  $\alpha = 0,05$ -Konfidenzintervall bei 5 TCP-Verbindungen.

Leistung des Netzes durch Lastverteilung zu steigern. Die Leistungsgröße des Telekommunikationsnetzes ist durch den Anteil der Anrufe gegeben, die nicht durchgestellt werden können.

Das ABC Routing basiert auf der Annahme symmetrischer Leitungen zwischen Knoten. Jeder Knoten sendet in regelmäßigen Intervallen Ameisen mit zufälligen Zielen ins Netz aus. Dies erfordert, dass den Knoten alle anderen Knoten im Netz bekannt sind. Die Ameisen werden entsprechend den Routingtabellen im Netz weitergeleitet und aktualisieren unterwegs die Informationen in den Routingtabellen. Ameisen werden bei Ankunft am Ziel verworfen, d.h. die Ameisen werden jeweils nur in eine Richtung geschickt. Um die Erstarrung der Routinginformationen zu verhindern, schlagen die Autoren vor, ein gewisses Rauschen einzubauen. Dadurch soll auch die Auffindung von neuen Wegen ermöglicht werden.

Das Verfahren ist für mobile Ad-hoc-Netze nicht geeignet, da es regelmäßig Ameisen aussendet, wodurch eine sehr große Last erzeugt wird, die nur von Festnetzen mit großer Kapazität tragbar ist. Weiterhin ist in mobilen Ad-hoc-Netzen nicht immer bekannt welche Knoten sich im Netz befinden. Deshalb ist das zufällige Aussenden von Ameisen sehr problematisch. Ein anderes Problem welches damit zusammenhängt ist, dass in mobilen Ad-hoc-Netzen die Last nicht auf alle Bereiche gleichverteilt ist. Es wird Bereiche im Netz geben, die mehr Verkehr haben als andere Bereiche.

### 5.6.2 AntNET

In [DMC96, CD97a, CD97b] wird ein Routingalgorithmus für paketvermittelte Netze vorgestellt, der als eine Erweiterung vom ABC Routing aus Abschnitt 5.6.1 angesehen werden kann. AntNET nutzt zwei Arten von Ameisen, die als *forward-ant* und *backward-ant* bezeichnet werden.

Allen Knoten im Netz senden in regelmäßigen Intervallen je einen forward-ant an einen zufällig gewählten Zielknoten. Auf dem Weg zum Ziel sammeln die forward-ants Daten, die später für die Aktualisierung der Routinginformationen verwendet werden. Zu den gesammelten Daten gehört die Identität der besuchten Knoten und die Zeit zwischen dem Aussenden beim Sender und der Ankunft beim aktuellen Knoten. Wenn der forward-ant beim Zielknoten ankommt, extrahiert dieser die gesammelten Daten und verwirft den forward-ant. Danach erstellt der Zielknoten einen backward-ant, der die Daten vom forward-ant enthält. Der backward-ant wird an den Quellknoten zurückgeschickt. Der backward-ant wird dabei auf dem gleichen Weg wie der forward-ant, nur in umgekehrter Richtung, weitergeleitet. Auf seinem Weg zum Quellknoten werden die Routinginformationen der vom forward-ant besuchten Knoten aktualisiert.

In AntNET werden forward-ants und backward-ants unterschiedlich behandelt. Die forward-ants werden wie Datenpakete bedient, d.h. sie verwenden die gleichen Warteschlangen und erfahren deshalb auch die gleichen Verzögerungen. Die backward-ants werden bevorzugt behandelt. Die Autoren begründen diese Vorgehensweise damit, dass die Aufgabe der forward-ants das Auskundschaften des Netzzustandes ist. Daher ist es erforderlich, dass die forward-ants wie Datenpakete behandelt werden. Die Aufgabe der backward-ants ist die Aktualisierung der Routinginformationen, wodurch die Leistung des Netzes gesteigert werden kann. Deshalb werden sie bevorzugt behandelt.

### 5.6.3 Verfahren von White

In [Whi97a, Whi97b, WBP98] wird ein weiteres Routingverfahren für leitungsvermittelte Netze beschrieben, das auf Ameisenalgorithmen basiert. Soll eine neue Verbindung aufgebaut werden, startet das Verfahren mit dem Erstellen einer neuen Ameisenkolonie, die im Netz verteilt wird. Dabei werden der Quell- und der Zielknoten jeweils als ein Nest betrachtet. Daraufhin starten die Ameisen von den Nestern. Auf der Basis der benutzten Wege der Ameisen wird der kürzeste Weg zwischen Quellknoten und Zielknoten gefunden. Dabei unterscheidet das Verfahren im Gegensatz zu den anderen vorgestellten Verfahren drei Klassen von Ameisen:

- **Explorer-ants:** suchen einen Weg zwischen Quellknoten und Zielknoten.
- **Allocater-ants:** reservieren die Ressourcen auf dem Pfad zwischen Quell- und Zielknoten.
- **Deallocater-ants:** geben reservierte Ressourcen auf dem Pfad zwischen Quell- und Zielknoten frei.

Da sowohl der Quellknoten als auch der Zielknoten als Nest angesehen werden, starten die Explorer-ants von beiden Knoten. Die Ameisen vom Quellknoten starten in Richtung des Zielknotens, und die Ameisen vom Zielknoten starten in Richtung des Quellknotens. Die Ameisen besitzen einen Speicher in der sie die besuchten Knoten vermerken – die so genannte Tabuliste – um Schleifen zu verhindern. Ameisen, die an ihrem Ziel angekommen sind, kehren sofort zurück und legen dabei eine Pheromonspur aus, die sich mit der Zeit verflüchtigt. Für den Rückweg wird die Tabuliste verwendet. Die Knoten erstellen eine Statistik aus den gesammelten Informationen. Diese Prozedur wird solange wiederholt, bis der Quellknoten der Meinung ist, dass ein „guter Pfad“ gefunden ist. Ein guter Pfad ist dadurch gekennzeichnet, dass in einem bestimmten Zeitintervall ein bestimmter Anteil –  $k\%$  der Ameisen

– diesen verwendet haben. Danach schickt der Quellknoten einen Allocater-ant los, um die Ressourcen, die auf dem Pfad liegen, zu reservieren. Wenn die Verbindung beendet ist und der Pfad nicht mehr benötigt wird, wird ein Deallocator-ant losgeschickt, um die reservierten Ressourcen frei zu geben.

#### 5.6.4 GPSAL

In [CL00b, CL00a, CL01] wird das *GPS/Ant-Like Routing Algorithm* für Ad-hoc-Netze vorgestellt. Das GPSAL basiert auf GPS und nutzt Ameisen, um Informationen im Netz zu sammeln und zu verteilen. Das Verfahren setzt voraus, dass alle Knoten mit GPS ausgestattet sind und ihre Positionen kennen. Weiterhin besitzen die Knoten detaillierte Informationen über Knoten im Netz, z.B. ihre aktuelle und vorherige Position mit Verweildauer, die Bewegungsgeschwindigkeit von mobilen Knoten, und können zwischen mobilen und nicht-mobilen Knoten unterscheiden.

Das Verfahren setzt voraus, dass ein bestimmter Anteil der Knoten im Netz nicht-mobil ist, und Pfade, die über diese Knoten laufen, sehr wenig Last verursachen. Der Einsatz von Ameisen wird nicht detailliert beschrieben. Die Autoren erwähnen, dass sie das gleiche Verfahren verwenden, wie die Verfahren aus den Abschnitten 5.6.2 und 5.6.3. Weiterhin wird vermerkt, dass die Auswahl der Ziele für Ameisen wichtig ist. Als mögliche Ziele werden Knoten mit den ältesten Informationen in der Routingtabelle und der größten Distanz vom Quellknoten vorgeschlagen.

#### 5.6.5 Mobile Agenten

Mobile Agenten unterscheiden sich von den hier vorgestellten Ameisenalgorithmen dahingehend, dass bei mobilen Agenten sowohl Daten als auch der Programmcode, der auf einem Knoten zur Ausführung kommt, übertragen werden [Lip02]. Wenn ein mobiler Agent von einem Knoten auf einen anderen migriert, dann trägt er auch die Ergebnisse mit, die er auf diesem Knoten gesammelt hat.

Von dieser Perspektive lassen sich die Verfahren aus Abschnitt 5.6.2 und 5.6.3 als mobile Agenten auffassen, jedoch führen die Ameisen hier keine eigenständigen Berechnungen durch, sondern sie sammeln lediglich Informationen. Der Knoten nutzt jedoch die Informationen der Ameisen – zumindest der Rückwärtsameisen – um seine Routinginformationen zu aktualisieren. Im Ameisenroutingalgorithmus (ARA) geschieht jedoch weder ein Transport von gesammelten Informationen bei den Ameisen noch bei den Datenpaketen. Weiterhin ist die Vorwärtsameise der Rückwärtsameise gleichgestellt, die Aufgaben unterscheiden sich prinzipiell nicht. Bei mobilen Agenten wird

erwartet, dass der mobile Agent irgendwann an seinen Heimatknoten zurückkehrt, und dass der Heimatknoten mit dem Ergebnis des mobilen Agenten einen Geschäftsprozess startet oder eine Entscheidung fällt.

### 5.6.6 Diskussion der unterschiedlichen Ansätzen

Zwischen ARA und den vorgestellten Verfahren aus den letzten Abschnitten gibt es einige Unterschiede, die die Eignung der Verfahren für mobile multi-hop Ad-hoc-Netze beeinflussen. Diese Unterschiede werden im Folgenden diskutiert.

- **Regelmäßiges Aussenden von Ameisen**

Der wichtigste Unterschied zwischen ARA und den anderen Verfahren ist, dass ARA kein regelmäßiges Aussenden von Ameisen vor sieht. Dies hat mehrere Gründe. Ein Grund ist, dass ein regelmäßiges Aussenden von Ameisen eine sehr hohe Last erzeugen würde, die für mobile multi-hop Ad-hoc-Netze nicht zu tragen ist. Weiterhin ist es unwahrscheinlich, dass der Kommunikationsverkehr in einem mobilen Ad-hoc-Netz gleichmäßig über alle Knoten verteilt ist. Deshalb würde das Aussenden von Agenten von allen Knoten wenig Sinn machen.

Ein anderer Grund, der gegen ein regelmäßiges Aussenden von Ameisen spricht, ist die Auswahl der Zielknoten. Einige der diskutierten Verfahren versuchen die Wahl der Zielknoten in Abhängigkeit des Netzwerkverkehrs zu treffen, andere fordern eine Gleichverteilung der Zielknoten im gesamten Netz. Die Netztopologie und die Anzahl der Knoten in einem mobilen multi-hop Ad-hoc-Netz variieren und sind schwer zu prognostizieren. Eine zufällige Wahl der Zielknoten, ohne zu wissen, ob der Knoten existiert, verursacht im Netz nur zusätzliche Kosten.

- **Vorwärtsameise**

Die diskutierten Verfahren benutzen die Vorwärtsameise, um Informationen über den Zustand des Netzes zu sammeln. Insbesondere besitzt die Vorwärtsameise einen Speicher in dem sie die gesammelten Informationen ablegt. Zu den gesammelten Informationen der Vorwärtsameise gehören die Adressen der besuchten Knoten, die Zeiten zwischen dem Aussenden beim Quellknoten und dem Ankommen bei den Knoten und die Verweildauer in den Knoten selbst. Aus diesem Grund werden Vorwärtsameisen wie Datenpakete behandelt, d.h. Vorwärtsameisen benutzen die gleichen Warteschlangen wie Datenpakete. Diese gesammelten Informationen werden beim Zielknoten der Rückwärtsameise übergeben, die dann auf dem Rückweg auf diesen Informationen aufbauend die Routingtabellen der Knoten anpasst.

Die Größe der Vorwärtsameise ist in diesem Fall von der Länge des Pfades abhängig und verursacht zusätzliche Last. In mobilen multi-hop Ad-hoc-Netzen ist es wünschenswert, die Last die durch das Routing entsteht, zu reduzieren. Daher sammelt die Vorwärtsameise im Ameisenroutingalgorithmus keine Daten.

Die Behandlung der Vorwärtsameise wie ein Datenpaket hat den Vorteil, dass der Zustand des Netzes genauer untersucht wird. Für mobile multi-hop Ad-hoc-Netze, in denen sich durch die Topologieänderungen auch die Kommunikationsstruktur hinsichtlich der verwendeten Pfade ändert, ist diese Vorgehensweise nicht von großem Vorteil, weil die gesammelten Informationen nach kurzer Zeit nicht mehr zutreffen müssen.

- **Rückwärtsameise**

In den diskutierten Verfahren nutzt die Rückwärtsameise die Informationen der Vorwärtsameise und wandert auf einem bestimmten Pfad zurück zum Quellknoten. Die Rückwärtsameise wird bevorzugt behandelt, um das Netz schnellstmöglich mit den gesammelten Informationen zu aktualisieren. Die Benutzung der Vorwärtsameise und der Rückwärtsameise ähnelt in diesem Fall sehr stark der Pfadsuche von DSR. Diese Vorgehensweise ist sinnvoll, da alle Knoten gleichzeitig Ameisen aussenden, um im ganzen Netz Informationen zu sammeln.

In ARA wird der BANT wie der FANT im Netz geflutet, um die Informationsaktualisierung im gesamten Netz durchzuführen. Jedoch kommt der FANT bzw. der BANT jeweils nur von einem Knoten, da hier nur die Kommunikation dieser beiden Knoten betrachtet wird. Diese Vorgehensweise ist für mobile multi-hop Ad-hoc-Netze besser geeignet, da hier die Kommunikationsverteilung nicht über das gesamte Netz gleich ist.

## 5.7 Fazit

In diesem Kapitel wurde ein neuer Routingalgorithmus, der *Ameisenroutingalgorithmus*, für mobile multi-hop Ad-hoc-Netze vorgestellt und mit bekannten Verfahren verglichen. Ameisenalgorithmen, die ein Teilbereich der Schwarmintelligenz sind, wurden in den letzten Jahren erfolgreich für unterschiedliche Optimierungsprobleme angewendet. Ausgehend vom einfachen Ameisenalgorithmus wurde das Modell für mobile multi-hop Ad-hoc-Netze adaptiert und zu einem Routingprotokoll entwickelt. Die Untersuchungen haben gezeigt, dass der vorgestellte Ansatz eine gute Leistung bei geringem Overhead für die untersuchten Szenarien bietet.

---

## KAPITEL 6

---

### Zusammenfassung und Ausblick

In dieser Arbeit wurde die automatische Adresskonfiguration und das Routing in mobilen multi-hop Ad-hoc-Netzen behandelt, die aktuell Gegenstand der Forschung sind. Ad-hoc-Netze erlauben den flexiblen Aufbau von Netzwerken überall und ohne jegliche Infrastruktur.

Zu Beginn der Arbeit wurden Ad-hoc-Netze allgemein charakterisiert und klassifiziert, da in der Literatur unterschiedliche Ansätze mit verschiedenen Annahmen zu finden sind. Weiterhin wurden zwei unterschiedliche Funktechniken vorgestellt, die für Simulationen bzw. für die Realisierung von Ad-hoc-Netzen in Frage kommen.

In Zusammenhang mit Ad-hoc-Netzen wird im Allgemeinen auch von Netzen gesprochen, die selbstkonfigurierend sind und wenig bis keinen Eingriff vom Benutzer erfordern. Die erste Fragestellung, mit der sich diese Arbeit beschäftigt, ist deshalb die Behandlung der automatischen Adresskonfiguration eines Ad-hoc-Netzwerks. Im Mittelpunkt steht die Benutzung von offenen Internet-Protokollen wie TCP und IP. Die Benutzung von IP für die Adressierung ist erforderlich, um existierende Protokolle und Anwendungen aus dem Internet und den lokalen Netzen auch in Ad-hoc-Netzen einsetzen zu können, da diese Protokolle und Anwendungen von einem bereits konfigurierten TCP/IP-Protokollstack ausgehen. Die gestellten Anforderungen an ein Verfahren legen den Schwerpunkt auf eine Vielzahl von Ad-hoc-Netzen, die dynamisch ihre Zusammensetzung ändern.

Im Rahmen dieser Arbeit wurde ein dynamisches Adressierungsverfahren für mobile multi-hop Ad-hoc-Netze vorgestellt, das die gestellten Anforderungen erfüllt. Insbesondere die Erkennung der Aufteilung eines Ad-hoc-Netzes in mehrere Ad-hoc-Netze und die Vereinigung von mehreren Ad-hoc-Netzen zu einem neuen Ad-hoc-Netz stand im Mittelpunkt des Entwurfs.

Die zweite Fragestellung, die diese Arbeit behandelt, ist das Routing in mobi-

len multi-hop Ad-hoc-Netzen. Seit drei Jahrzehnten wird an diesem Problem gearbeitet und es wurden unterschiedliche Ansätze vorgeschlagen, von denen keine für alle Szenarien geeignet ist. Das Routing spielt in mobilen multi-hop Ad-hoc-Netzen eine noch wichtigere Rolle als in Festnetzen, da aufgrund der Knotenmobilität die Netzwerktopologie Änderungen unterworfen ist, die berücksichtigt werden müssen. Der für die Bereitstellung des Routing-Dienstes erzeugte Overhead ist sehr hoch, wodurch Verfahren benötigt werden, die möglichst wenig Overhead erzeugen.

Im Rahmen dieser Arbeit wurde ein neuartiger Routingalgorithmus auf der Basis von Ameisenalgorithmen vorgestellt. Ameisenalgorithmen ahmen das Verhalten von Ameisenkolonien bei der Lösung einer Aufgabenstellung nach. Sie wurden in den letzten Jahren für unterschiedliche mathematische Optimierungsprobleme erfolgreich angewandt. Ausgehend von der Grundidee wurde der Ameisenalgorithmus für den Bedarf von mobilen multi-hop Ad-hoc-Netzen angepasst. Das Ergebnis ist ein reaktiver und effizienter Routingalgorithmus für mobile multi-hop Ad-hoc-Netze. Durch Simulationsergebnisse wurde die Leistung des Ameisenroutingalgorithmus zwei bekannten Vertretern, AODV und DSR, gegenübergestellt. Die Leistung des Ameisenroutingalgorithmus lässt sich in den Standardszenarien mit den Leistungen von AODV und DSR vergleichen. Außerdem wurde untersucht in wieweit sich die Routingalgorithmen für die Übertragung von Multimediadaten eignen, wobei der Schwerpunkt auf die Übertragungsqualität von Sprache gelegt wurde. Die Ergebnisse der drei Routingalgorithmen sind ähnlich. Sie erlauben, mit der in den Simulationen eingesetzten Technik IEEE 802.11 mit 2 Mbit/s, keine akzeptable Übertragung von Sprachverbindungen. Das letzte untersuchte Kriterium für die Routingalgorithmen war ihre Auswirkung auf die Leistung von TCP. Die Ergebnisse des Ameisenroutingalgorithmus sind vergleichbar mit denen von AODV bzw. DSR und liegen teilweise über diesen.

## Ausblick

Nicht alle Aspekte der vorgestellten Probleme konnten in dieser Arbeit behandelt werden. Es bleiben noch eine ganze Reihe von Fragen offen, von denen einige im Folgenden kurz angesprochen werden sollen.

Die Adresskonfiguration und das Routing wurden voneinander unabhängig betrachtet. In der Realität hängen beide Fragestellungen jedoch stark zusammen. Deshalb ist es von Bedeutung, die Adresskonfiguration und das Routing zu verzahnen. Insbesondere könnte dadurch die für die Adresskonfiguration benötigte Last durch Zusammenlegen von Funktionen mit dem Routing verringert werden.

Sicherheitsaspekte wurden im Rahmen dieser Arbeit nicht berücksichtigt. Dies betrifft sowohl die Sicherheit für den Routingalgorithmus, als auch die Sicherheit bezüglich der Adresskonfiguration. Beide Ansätze sind gegen Angriffe jeglicher Art offen.



---

## Literaturverzeichnis

- [ACJ<sup>+</sup>03] ADJIH, CEDRIC, THOMAS CLAUSEN, PHILIPPE JACQUET, ANIS LAOUITI, PASCALE MINET, PAUL MUHLETHALER, AMIR QAYYUM und LAURENT VIENNOT: *Optimized Link State Routing Protocol*, INTERNET-DRAFT, IETF MANET Working Group, *draft-ietf-manet-olsr-08.txt*, 2003.
- [BDT99] BONABEAU, ERIC, MARCO DORIGO und GUY THERAULAZ: *Swarm intelligence: from natural to artificial intelligence*. Oxford University Press, 1999. ISBN: 0-19-513159-2.
- [BG02] BOUAZIZI, IMED und MESUT GÜNES: *A Framework for Transmitting Video over Mobile multi-hop Ad-Hoc Networks*. In: *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, Band XV, Orlando, USA, July 2002.
- [BG03] BOUAZIZI, IMED und MESUT GÜNES: *A video streaming framework for transporting MPEG-4 Video over mobile ad-hoc networks*. In: BELGHITH, ABDELFETTAH, SAMI TABBANE, NAOUEL BEN ALI und ACHRAF GAZDAR (Herausgeber): *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003)*, Seiten 211–218, Mahdia, Tunisia, 25-27 June 2003.
- [BHS97] BULLNHEIMER, B., R. HARTL und C. STRAUSS: *A New Rank Based Version of the Ant System — A Computational Study*. Swarm intelligence, University of Vienna, Institute of Management Science, 1997.
- [BMJ<sup>+</sup>98] BROCH, JOSH, DAVID A. MALTZ, DAVID B. JOHNSON, YIHCUN HU und JORJETA JETCHEVA: *A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols*. Proceedings of the Fourth Annual ACM/IEEE Internatio-

nal Conference on Mobile Computing and Networking (MobiCom'98), Seiten 85–97, 1998.

- [BS01] BRAY, JENNIFER und CHARLES STURMAN: *Bluetooth: connect without cables*. Prentice Hall, Upper Saddle River, USA, 2001. ISBN: 0-13-089840-6.
- [CA01] CHESHIRE, STUART und BERNARD ABOBA: *Dynamic Configuration of IPv4 Link-Local Addresses*. <http://www.ietf.org/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-04.txt>, July 2001.
- [CBD02] CAMP, T., J. BOLENG und V. DAVIES: *A Survey of Mobility Models for Ad Hoc Network Research*. Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2(5):483–502, 2002.
- [CD97a] CARO, GIANNI DI und MARCO DORIGO: *AntNet: a mobile agents approach to adaptive routing*. Technical IRIDIA/97-12, Universite Libre de Bruxelles, Belgium, 1997.
- [CD97b] CARO, GIANNI DI und MARCO DORIGO: *A Study of Distributed Stigmergetic Control for Packet-Switched Communications Networks*. Technical IRIDIA/97-20, Universite Libre de Bruxelles, Belgium, 1997.
- [CDR98] CRUSE, HOLK, JEFFREY DEAN und HELGE RITTER: *Die Entdeckung der Intelligenz oder Können Ameisen denken? - Intelligenz bei Tieren und Maschinen*. Verlag C.H. Beck, München, 1998. ISBN: 3-406-44073-8.
- [CG85] CROFT, W.J. und J. GILMORE: *Bootstrap Protocol, RFC 951*. <http://www.ietf.org/rfc/rfc951.txt>, September 1985.
- [CJL<sup>+</sup>01] CLAUSEN, T., P. JACQUET, A. LAOUITI, P. MUHLETHALER, A. QAYYUM und L. VIENNOT: *Optimized Link State Routing Protocol*. In: *Proceedings of INMIC 2001*, Seiten 177–184, Pakistan, December, 28-30 2001. IEEE.
- [CL00a] CÂMARA, DANIEL und ANTONIO A.F. LOUREIRO: *A GPS/Ant-Like Routing Algorithm for Ad Hoc Networks*. In: *IEEE Wireless Communications and Networking Conference*, Seiten 1232–1237, Chicago, IL, USA, September 2000.

- [CL00b] CÂMARA, DANIEL und ANTONIO A.F. LOUREIRO: *A Novel Routing Algorithm for Ad Hoc Networks*. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Band 8, Seiten 8022–8028, Maui, January 2000. IEEE Computer Society.
- [CL01] CÂMARA, DANIEL und ANTONIO A.F. LOUREIRO: *A GPS/Ant-Like Routing Algorithm for Ad Hoc Networks*. Journal of Telecommunications Systems, 18(1-3):85–100, September 2001.
- [CM99] CORSON, S. und J. MACKER: *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, RFC 2501. <http://www.ietf.org/rfc/rfc2501.txt>, Jan 1999.
- [CMC99] CORSON, M. SCOTT, JOSEPH P. MACKER und GREGORY H. CIRINCIONE: *Internet-Based Mobile Ad Hoc Networking*. IEEE Internet Computing, 3(4):63–70, July, August 1999.
- [CTAG01] CATRINA, OCTAVIAN, DAVE THALER, BERNARD ABOBA und ERIK GUTTMAN: *Zeroconf Multicast Address Allocation Protocol (ZMAAP)*. <http://www.ietf.org/internet-drafts/draft-ietf-zeroconf-zmaap-02.txt>, Oct 2001.
- [DCY00] DAS, SAMIR R., ROBERT CASTANEDA und JIANGTAO YAN: *Simulation-based performance evaluation of routing protocols for mobile ad hoc networks*. Mobile Networks and Applications, 5:179–189, 2000.
- [DD99] DORIGO, MARCO und GIANNI DI CARO: *The Ant Colony Optimization Meta-Heuristic*. In: CORNE, DAVID, MARCO DORIGO und FRED GLOVER (Herausgeber): *New Ideas in Optimization*, Seiten 11–32. McGraw-Hill, London, 1999.
- [DG97a] DORIGO, MARCO und LUCA MARIA GAMBARDELLA: *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. IEEE Transactions on Evolutionary Computation, 1(1):53–66, April 1997. <http://citeseer.nj.nec.com/article/dorigo96ant.html>.
- [DG97b] DORIGO, MARCO und L. M. GAMBARDELLA: *Ant Colonies for the Travelling Salesman Problem*. Bio Systems, 1:73–81, 1997.

- [DMC91a] DORIGO, MARCO, VITTORIO MANIEZZO und ALBERTO CORLONI: *Ant System: An Autocatalytic Optimizing Process*. Swarm intelligence 91-016 Revised, Politecnico di Milano, Milano, Italy, 1991.
- [DMC91b] DORIGO, MARCO, VITTORIO MANIEZZO und ALBERTO CORLONI: *Positive Feedback as a Search Strategy*. Swarm intelligence 91-016, Politecnico di Milano, Milano,, 1991.
- [DMC96] DORIGO, MARCO, VITTORIO MANIEZZO und ALBERTO CORLONI: *Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Trans. on Systems, Man, and Cybernetics–Part B, 26(1):29–41, 1996.
- [Dro97] DROMS, R.: *Dynamic Host Configuration Protocol, RFC 2131*. <http://www.ietf.org/rfc/rfc2131.txt>, March 1997.
- [Dud00] *Duden – Fremdwörterbuch*, Band 5. Dudenverlag, Mannheim, Leipzig, Wien, Zürich, 2000.
- [FL01] FREEBERSYSER, JAMES A. und BARRY LEINER: *A DoD Perspective on Mobile Ad Hoc Networks*. In: PERKINS, CHARLES E. (Herausgeber): *Ad Hoc Networking*, Ad-hoc chapter 2, Seiten 29–51. Addison-Wesley, 2001.
- [FMMT84] FINLAYSON, ROSS, TIMOTHY MANN, JEFFREY MOGUL und MARVIN THEIMER: *A Reverse Address Resolution Protocol, RFC 903*. <http://www.ietf.org/rfc/rfc903.txt>, June 1984.
- [FV00] FALL, KEVIN und KANNAN VARADHAN: *The ns Manual*, Nov 2000.
- [GHB03] GÜNES, MESUT, MARCEL HECKER und IMED BOUAZIZI: *Influence of adaptive RTS/CTS retransmissions on TCP in wireless and ad-hoc networks*. In: *Proceedings of the 8th IEEE Symposium on Computer and Communications, (ISCC 2003)*, Band II, Seiten 855–860, Antalya, Turkey, June, July 2003. IEEE.
- [GKB03] GÜNES, MESUT, MARTIN KÄHMER und IMED BOUAZIZI: *Ant-Routing-Algorithm (ARA) For Mobile Multi-Hop Ad-Hoc Networks – New Features And Results*. In: BELGHITH, ABDELFETTAH, SAMI TABBANE, NAOUEL BEN ALI und ACHRAF GAZDAR (Herausgeber): *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003)*, Seiten 9–20, Mahdia, Tunesia, 25-27, June 2003.

- [GR02a] GÜNES, MESUT und JÖRG REIBEL: *An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-Hoc Networks*. In: *Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services*, Sophia Antipolis, France, September 2002. ETSI.
- [GR02b] GÜNES, MESUT und JÖRG REIBEL: *Ein dynamisches Adressierungsverfahren für Mobile Ad-Hoc Netze*. In: *Mobile Ad-Hoc Netzwerke*, Band Lecture Notes in Informatics (LNI) - Proceedings, Seiten 59–78. Michael Weber, Frank Kargl, März 2002.
- [Grö85] GRÖSSWALD, KARL: *Organisation und Leben der Ameisen*. Wissenschaftliche Verlagsgesellschaft, 1985. ISBN: 3-8047-0691-6.
- [GS02] GÜNES, MESUT und OTTO SPANIOL: *Routing Algorithms for Mobile Multi-Hop Ad-Hoc Networks*. In: TURLAKOV, HRISTO und LUBEN BOYANOV (Herausgeber): *Proceedings of Next Generation Network Technologies International Workshop*, Seiten 10–24. The Central Laboratory of Parallel Processing, Department "Distributed Computing Systems and Networks", October 2002.
- [GSB02] GÜNES, MESUT, UDO SORGES und IMED BOUAZIZI: *ARA - The Ant-Colony Based Routing Algorithm for MANETs*. In: OLARIU, STEPHAN (Herausgeber): *Proceedings of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN 2002)*, Seiten 79–85. IEEE, August 2002.
- [Gut01a] GUTTMAN, ERIK: *An API for the Zeroconf Multicast Address Allocation Protocol (ZMAAP)*. <http://www.ietf.org/internet-drafts/draft-ietf-zeroconf-zmaap-api-00.txt>, Jun 2001.
- [Gut01b] GUTTMAN, ERIK: *Autoconfiguration for IP Networking: Enabling Local Communication*. Internet Computing, 5(3):81–86, May, June 2001.
- [Gut01c] GUTTMAN, ERIK: *Zeroconf Host Profile Applicability Statement*. <http://www.ietf.org/internet-drafts/draft-ietf-zeroconf-host-prof-01.txt>, July 2001.
- [GV02] GÜNES, MESUT und DONALD VLAHOVIC: *The Performance of the TCP/RCWE Enhancement for Ad-Hoc Networks*. In: ANTONIO CORRADI, MAHMOUD DANESMAND (Herausgeber): *Pro-*

*ceedings of the Seventh IEEE Symposium on Computers and Communications, ISCC 2002*, Seiten 43–48. IEEE, July 2002.

- [HD98] HINDEN, R. und S. DEERING: *IP Version 6 Addressing Architecture*, RFC 2373. <http://www.ietf.org/rfc/rfc2373.txt>, July 1998.
- [HGPC99] HONG, X., M. GERLA, G. PEI und C.-C. CHIANG: *A Group Mobility Model for Ad Hoc Wireless Networks*. In: *Proceedings of ACM/IEEE MSWiM'99*, Seiten 53–60, August 1999.
- [HJ00] HU, YIH-CHUN und DAVID B. JOHNSON: *Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks*. In: *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Seiten 231–242. ACM, August 2000. <http://citeseer.nj.nec.com/hu00caching.html>.
- [HP99] HAAS, Z. und M. PEARLMAN: *The Zone Routing Protocol (ZRP) for Ad Hoc Networks (Internet-Draft)*. Mobile Ad Hoc Networking Group (MANET), IETF, 1999.
- [Iee97] *Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE, Institute of Electrical and Electronics Engineers, Inc., 1997.
- [Iee99] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*. IEEE 802.11b, Institute of Electrical and Electronics Engineers, Inc., The Institute of Electrical and Electronics Engineers, Inc., 1999.
- [Iet03] *Ad hoc Network Scaling Research Discussion*, 2003. <http://www1.ietf.org/mail-archive/working-groups/ans-research/current/index.html>.
- [Iet] *IETF Mobile Ad-hoc Networks (Manet) Working Group*. <http://www.ietf.org/html.charters/manet-charter.html>.
- [Joh94] JOHNSON, DAVID B.: *Routing in Ad Hoc Networks of Mobile Hosts*. In: *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, Seiten 158–163, Santa Cruz, USA, December 1994. <http://citeseer.nj.nec.com/johnson94routing.html>.

- [Lip02] LIPPERTS, STEFFEN RICHARD GOSWIN: *Mobile Agent Support Services*. Mobile agents, University of Aachen, Department of Computer Science, Aachen, Germany, February 2002.
- [LL68] LARSON, PEGGY PICKERING und MERVIN W. LARSON: *Insekten Staaten - Aus dem Leben der Wespen, Bienen, Ameisen und Termiten*. Verlag Paul, Hamburg, Berlin, Deutschland, 1968. ISBN: 3-490-03618-2.
- [MC98] MACKER, JOSEPH P. und M. SCOTT CORSON: *Mobile Ad Hoc Networking and the IETF*. Mobile Computing and Communications Review, 2(1):9–14, 1998.
- [MP02] MOHSIN, MANSOOR und RAVI PRAKASH: *IP Address Assignment In A Mobile Ad Hoc Network*. In: *Military Communications Conference (MILCOM 2002)*, Anaheim, USA, October 2002.
- [NP02] NESARGI, SANKET und RAVI PRAKASH: *MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network*. In: *Proceedings of The 21st Annual Joint Conference of the IEEE Computer and Communications Societies, (InfoCom 2002)*, New York, NY USA, June 2002. IEEE. <http://www.ieee-infocom.org/2002/papers/756.pdf>.
- [Oci01] OUI, IEEE STANDARDS TUTORIALS: und COMPANY ID: *Guidelines for 64-bit Global Identifier (EU-64) Registration Authority*. <http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>, May 2001.
- [PB94] PERKINS, CHARLES E. und P. BHAGVAT: *Hihgly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*. Computer Communications Rev., Seiten 234–244, October 1994.
- [Per98] PERKINS, CHARLES E.: *Mobile IP: design and principles and practices*. Prentice Hall PTR, 1 Auflage, 1998. ISBN: 0201634694.
- [Per01] PERKINS, CHARLES E. (Herausgeber): *Ad Hoc Networking*. Addison-Wesley, 1 Auflage, 2001. ISBN: 0-201-30976-9.
- [PGC00] PEI, GUANGYU, MARIO GERLA und TSU-WEI CHEN: *Fisheye State Routing in Mobile Ad Hoc Networks*. In: *ICDCS Workshop on Wireless Networks and Mobile Computing*, Seiten D71–D78, 2000.

- [PKP01] PARK, JUNG-SOO, YONG-JIN KIM und SUNG-WOO PARK: *Stateless address autoconfiguration in Mobile Ad Hoc Networks using site-local address, draft-park-zeroconf-manet-ipv6-00.txt.* <http://www.ietf.org/internet-drafts/draft-park-zeroconf-manet-ipv6-00.txt>, July 2001.
- [Plu82] PLUMMER, DAVID C.: *An Ethernet Address Resolution Protocol, RFC 826.* <http://www.ietf.org/rfc/rfc826.txt>, 1982.
- [PMW<sup>+01</sup>] PERKINS, CHARLES E., JARI T. MALINEN, RYUJI WAKI-KAWA, ELIZABETH M. BELDING-ROYER und YUAN SUN: *IP Address Autoconfiguration for Ad Hoc Networks, draft-perkins-manet-autoconf-01.txt.* <http://www.ietf.org/internet-drafts/draft-perkins-manet-autoconf-01.txt>, November 2001.
- [PRD02] PERKINS, CHARLES E., E. M. ROYER und S. R. DAS: *Ad Hoc On Demand Distance Vector (AODV) Routing.* IETF Internet draft, [draft-ietf-manet-aodv-12.txt](http://www.cs.ucs.edu/ebelding/txt/aodvid.txt), November 2002. <http://www.cs.ucs.edu/ebelding/txt/aodvid.txt>.
- [Rap96] RAPPAPORT, THEODORE. S.: *Wireless communications, principles and practice.* Communications Engineering and Emerging Technologies. Prentice-Hall, Upper Saddle River, New Jersey, USA, 1996. ISBN: 0-13-375536-3.
- [RR02] RAMANATHAN, RAM und JASON REDI: *A brief overview of ad hoc networks: challenges and directions.* IEEE Communications Magazine, 40(5):20–22, May 2002.
- [RT99] ROYER, ELIZABETH M. und CHAI-KEONG TOH: *A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks.* IEEE Personal Communications, Seiten 46–55, April 1999.
- [Sas99] SASS, PAUL: *Communications networks for the Force XXI Digitized Battlefield.* Mobile Networks and Applications (MONET), Mobile Ad Hoc Networking(4):139–155, October 1999.
- [SCFJ01] SCHULZRINNE, CASNER, FREDERICK und JACOBSON: *RTP: A Transport Protocol for Real-Time Applications, Internet-Draft.* <http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-new-11.txt>, November 2001.

- [Sch00] SCHILLER, JOCHEN: *Mobile Communications*. Addison-Wesley, Bonn, Paris, Massachusetts, 2000. ISBN: 0-201-39836-2.
- [SH97a] STÜTZLE, THOMAS und HOLGER HOOS: *Improvements on the Ant-System: Introducing the MAX-MIN Ant System*. In: *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms*, Norwich, England, 1997. Springer-Verlag.
- [SH97b] STÜTZLE, T. und H. HOOS: *The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem*. In: *Proceedings of the Fourth International Conference on Evolutionary Computation (ICEC'97)*, Seiten 308–313. IEEE Press, 1997.
- [SHBR96] SCHOONDERWOERD, RUUD, OWEN E. HOLLAND, JANET L. BRUTEN und LEON J. M. ROTHKRANTZ: *Ant-Based Load Balancing in Telecommunications Networks*. Adaptive Behaviour, 2:169–207, 1996.
- [Sta02] STALLINGS, WILLIAM: *Wireless Communications and Networking*. Prentice-Hall, Upper Saddle River, New Jersey, USA, 2002. ISBN: 0-13-040864-6.
- [Tan96] TANENBAUM, ANDREW S.: *Computernetzwerke*. Prentice Hall, 3 Auflage, 1996. ISBN: 3-8272-9536-x.
- [Toh02] TOH, CHAI-KEONG: *Ad hoc mobile wireless networks: protocols and systems*. Prentice Hall PTR, 1 Auflage, 2002. ISBN: 0130078174.
- [WBP98] WHITE, TONY, A. BIESZCZAD und B. PAGUREK: *Distributed Fault Location in Networks Using Mobile Agents*. In: ALBAYRAK, S. und F. J. GARIJO (Herausgeber): *Intelligent Agents for Telecommunication Applications — Proceedings of the Second International Workshop on Intelligent Agents for Telecommunication (IATA '98)*, Band 1437. Springer-Verlag: Heidelberg, Germany, 1998.
- [Whi97a] WHITE, TONY: *Routing with Swarm Intelligence*. Swarm intelligence SCE-97-15, Systems and Computer Engineering Department, Carleton University, Canada, September 1997.
- [Whi97b] WHITE, TONY: *Swarm intelligence and problem solving in telecommunications*. Canadian Artificial Intelligence Magazine, Spring 1997.

- [ZNM03] ZHOU, HONGB0, LIONEL NI und MATT MUTKA: *Prophet Address: Allocation for Large Scale MANETs.* In: *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom 2003)*, San Francisco, March, April 2003. IEEE. [http://www.ieee-infocom.org/2003/papers/32\\_02.PDF](http://www.ieee-infocom.org/2003/papers/32_02.PDF).

---

## Lebenslauf

8. April 1973 Geburt in Gaziantep, Türkei
- 1979 - 1981 Besuch der Grundschule in Gaziantep
- 1981 - 1984 Besuch der Grundschule in der Andernacher Straße in der Hansestadt Bremen
- 1984 - 1990 Besuch der Gesamtschule am Schulzentrum an der Waliserstraße in der Hansestadt Bremen
- 1990 - 1993 Besuch des Gymnasiums an der Waliserstraße in der Hansestadt Bremen  
Abitur im Mai 1993
- 1993 - 1998 Studium der Informatik an der Rheinisch-Westfälischen Technischen Hochschule Aachen  
Diplom im Juni 1998
- 10/1998 - wissenschaftlicher Mitarbeiter am Lehrstuhl für Informatik IV der Rheinisch-Westfälischen Technischen Hochschule Aachen

