Routingalgorithmen für infrastrukturlose Paketfunknetze mit kooperativen mobilen Stationen

Vom Fachbereich Elektrotechnik und Informatik
der Universität Siegen
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

genehmigte

Dissertation

von
Diplom-Ingenieur Ralph Jansen
geboren in Ochsenfurt

1. Gutachter: Prof. Dr. Bernd Freisleben
2. Gutachter: Prof. Dr. Peter Kamerer

Tag der mündlichen Prüfung: 11.4.2002

urn:nbn:de:hbz:467-366

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die auf unterschiedliche Arten das Zustandekommen der vorliegenden Arbeit unterstützt haben.

Mein besonderer Dank gilt zunächst meinen Doktorvater Prof. Bernd Freisleben, dessen beständiger Einsatz das Entstehen vieler Teile der Arbeit erst ermöglicht hat und dem ich wesentliche inhaltliche Anregungen bei der Erstellung der Arbeit verdanke. Ebenso gilt mein Dank auch meinem Korreferenten Prof. Peter Kammerer, der mir den Beginn der Arbeit durch die Aufnahme in sein Institut ermöglichte und der mir während der ersten drei Jahre die nötige Unterstützung zukommen ließ. Mein Dank gilt dabei auch dem ISIA-Graduiertenkolleg, und insbesondere seinem Sprecher Prof. Manfred Glesner, ohne dessen Förderung diese Arbeit nie hätte beginnen können. Ausserdem möchte ich mich bei dem Vorsitzenden der Prüfungskommission, Prof. Wolfgang Merzenich, bedanken, der es ermöglichte, dass Verfahren reibungslos und zügig durchzuführen.

Weiterhin möchte ich den ehemaligen Mitgliedern des Fachgebiets Betriebssysteme der TU Darmstadt für ihren jahrelangen Beistand danken. Dazu gehören Prof. Henning Pagnia, dem ich die Anleitung für erste Schritte im wissenschaftlichen Arbeiten verdanke und Jörg Baumgart, sowie Prof. Oliver Theel, die für alle Probleme gute Ansprechpartner waren. Ganz besonders möchte ich mich auch bei der guten Seele des Fachgebiets, Gudrun Jörs, bedanken, die mich unermüdlich trotz aller Höhen und Tiefen immer wieder auf den richtigen Weg gebracht hat. Nicht vergessen möchte ich auch den Rest der Darmstädter Kaffeerunde, Thomas Kunkelmann, Christoph Liebig, Renato Vinga-Martins, Christian Hochberger, Ralf Schneider und Peter Hartmann, die alle auf ihre besondere Art und Weise zu einer produktiven Arbeitsatmosphäre beitrugen.

Den Kollegen an der Universität Siegen gebührt ebenfalls mein besonderer Dank. Hier sind insbesondere Dr. Kurt Sieber und Dr. Wolfgang Golubski zu nennen, deren Rat mir bei vielen Problemen weitergeholfen hat. Den Netzwerkspezialisten der Informatik, Frank Schuh und Christoph Schlechtingen, danke ich für die Bereitstellung der Technik, die für die Experimente von Nöten war und den Systemspezialisten Thomas Unger, Jürgen Schöw und Michael Engel danke ich für die viele Zeit, die sie für meine Versuchsprojekte geopfert haben. Ausserdem möchte ich noch meinen Kollegen Markus Borschbach, Peter Merz, Guido Rössling, Ralf Greb,

Frank Thilo, Thomas Barth, Marc Staiger und den Sekretärinnen Marion Kielmann, Esther te Vaarwerk und Birgit Berger-Bedarff für ihre Unterstützung bei der Arbeit danken. Aber auch Studenten haben zu dieser Arbeit beigetragen. Sie enthält Ideen, die in Seminaren und Diskussionen entwickelt wurden und eine Reihe von Analysen und Simulationen aus Diplomarbeiten. Besonders zu nennen sind hierbei Sascha Demetrio, Frank Türke und Jörg Kühn, deren umfangreiche Untersuchungen sich in dieser Arbeit wiederfinden.

Mein tiefster Dank gilt meiner Familie, die mit Abstand den größten Anteil daran hatte, dass diese Arbeit verwirklicht werden konnte. Ich danke meinen Eltern, die mir alle Voraussetzungen für das Erreichen dieses Zieles mitgegeben haben, die aber leider den Abschluss der Dissertation nicht mehr erleben durften, sowie meinem Bruder Marcus und meiner Grossmutter Frieda Moder für ihren unerschütterlichen Beistand in allen Lebenssituationen, die mir die Kraft gaben, dieses Projekt bis zum Ende durchzuhalten.

Siegen, im April 2002

Ralph Jansen

Inhaltsverzeichnis

Kapitel 1

Einleitung

Rechnernetze sind ein universelles Kommunikationsmedium. Sie integrieren zunehmend andere Kommunikationstechnik wie Telefon, Fernsehen oder den Briefverkehr, da durch die fortschreitende Digitalisierung immer mehr Informationen in ein für Rechnernetze geeignetes Format umgewandelt werden können. Diese Vielseitigkeit der digitalen Übertragung bewirkt aber nicht nur, dass herkömmliche Technik ersetzt wird. Vielmehr entstehen fortwährend neue Möglichkeiten ihrer Nutzung, und damit etablieren sich neue Medien. Als Beispiel hierfür sei das World Wide Web genannt. Zudem sind auch diese neuen Kommunikationsdienste einem andauernden Wandel ausgesetzt; ein gutes Beispiel hierfür ist die elektronische Post (Email), die im privaten Sektor zunehmend Konkurrenz durch den Short Message Sevice (SMS) bekommt. Das hohe Potenzial dieser Technologie bewirkt ein starkes Wachstum in fast allen Anwendungsgebieten der digitalen Kommunikationstechnik. Die Anzahl der an das Internet angeschlossenen Rechner wächst exponenziell, ebenso wächst die Anzahl der über das Internet erreichbaren Personen. Die Mobilkommunikation kann ähnliche Wachstumsraten vorweisen. Alle modernen Mobilkommunikationssysteme sind speziell angepasste Varianten moderner Rechnernetze, sie verwenden eine digitale Übertragung und können so eine ganze Palette verschiedener Dienste anbieten.

Neue Technologien, insbesondere die Miniaturisierung aller Elektronikkomponenten, treiben diese Entwicklungen weiter voran. Es ist absehbar, dass die Miniaturisierung immer kleinere Geräte mit immer größerer Leistungsfähigkeit hervorbringen wird. In naher Zukunft werden auch einfache Geräte Mikroprozessoren besitzen, die intelligent genug sind, um Netzwerkkommunikation zu unterstützen. Die kostengünstige Verfügbarkeit zusätzlicher Dienste wird voraussichtlich auch Waschmaschinen und Heizungssteuerungen mit Internetanschluss hervorbringen. Die dazu notwendige Basistechnologie wird weiter standardisiert. Das derzeit bekannteste Projekt im Bereich Funknetzwerke ist Bluetooth, ein Zusammenschluss mehrerer großer Firmen, die eine Schnittstelle definieren, mit der sich Geräte wie PCs, Tastaturen, Handys, Drucker oder auch Haushaltsgeräte per Funk verbinden lassen. Ein wesentliches Entwicklungsziel ist die Herstellung besonders günstiger und massenmarktfähiger Module zur Funkkommunikation.

Eine neue Kommunikationstechnik kommt nicht ohne entsprechende Software und Protokolle

aus, und hier existiert noch ein großer Entwicklungsbedarf. Es ist beispielsweise absehbar, dass in naher Zukunft im derzeit verwendeten Internetprotokoll IPv4 die Rechneradressen knapp werden. An einer Lösung dieses Problems wird eifrig gearbeitet. Die Neuentwicklung IPv6 wird 1500 IP-Adressen pro Quadratmeter Erdoberfläche unterstützen. Die Netze, die damit möglich werden, sind nur noch durch selbstkonfigurierende Netzwerksysteme zu verwalten. Die Voraussetzungen für Supernetzwerke mit Milliarden von Kleinststationen sind im Entstehen - die passende Software dazu fehlt noch.

Die vorliegende Arbeit adressiert das Problem der selbstkonfigurierenden Netzwerke und behandelt dabei speziell die Netzwerke, die ohne feste Infrastruktur auskommen müssen. Solche Netzwerke können unabhängig von irgendwelchen Koordinierungsstellen eine eigenständige Kommunikationsstruktur aufbauen. Die dazu notwendigen Aufgaben müssen im Netzwerk unter den Stationen verteilt werden. So entstehen Ad-hoc Netzwerke aus kooperativen Stationen. Da die Verbindungen zwischen den Stationen drahtlos hergestellt werden, sind Positionswechsel der Stationen leicht möglich. Ein wesentlicher Teil dieser Arbeit beschäftigt sich deshalb mit den Auswirkungen von Positionswechseln auf die Berechnung der Routen zu den Teilnehmern. Dazu werden entsprechende Netzwerkmodelle vorgestellt und die Kommunikationsmöglichkeiten in den Netzwerken durch Berechnungen wie auch durch Simulationen untersucht.

Die Arbeit beginnt mit einem Überblick über die historische Entwicklung der Funknetzwerke in Kapitel 2. Ausgehend von der analogen Funktechnik wird die Entwicklung der Zellularstruktur und der Wandel zur digitalen Funktechnik beschrieben. Dabei werden die Unterschiede von Zellularnetzen zu den in dieser Arbeit untersuchten Paketfunknetzwerken erläutert und die Besonderheiten der Ad-hoc Netzwerke aufgezeigt. Anschließend gibt Kapitel 2 eine kurze Einführung in die Grundlagen moderner Rechnernetzwerke. Dazu werden die gängigen Protokolle und der schichtweise Aufbau gebräuchlicher Netzwerke erklärt. Schließlich wird eine Einführung in die verwendeten Begriffe der Graphentheorie gegeben.

Kapitel 3 widmet sich allein dem Routing. Es werden aktuelle und in der Entwicklung befindliche Routingverfahren beschrieben. Der erste Teil des Kapitels gibt eine Übersicht zu den in der Literatur beschriebenen Standardalgorithmen und erläutert auch, welche besonderen Routingprobleme in Ad-hoc Netzwerken zu lösen sind. Im zweiten Teil des Kapitels werden die unterschiedlichen Möglichkeiten vorgestellt, die zur Lösung der Routingprobleme in Ad-hoc Netzen vorgeschlagen wurden. Da die Lösungsansätze sehr verschiedenartig sind, werden zur besseren Verständlichkeit aus der Literatur bekannte Routingalgorithmen in verwandte Gruppen eingeteilt. Anschließend wird die Arbeitsweise der Algorithmen erläutert, wobei die Vor- und die Nachteile der jeweiligen Lösungsansätze analysiert werden.

Kapitel 4 beschreibt, welche Bewertungsverfahren zur Untersuchung von Routingalgorithmen eingesetzt werden. Sowohl die theoretische Analyse als auch praktische Testverfahren werden dabei betrachtet, zu denen auch die Simulationen zählen. Die Grenzen einer rein theoretischen Analyse von Routingalgorithmen werden beschrieben. Ebenso wird auf die Probleme von Implementierung und Messungen im Feld eingegangen. Simulationen sind ein sehr flexibles Testwerkzeug, allerdings erfordert ihr Einsatz eine umfangreiche Modellbildung und statistische Auswertungen. Kernpunkt des Kapitels ist die Vorstellung eines selbst entwickelten Simulators. Es werden Testsituationen für Ad-hoc Routingalgorithmen vorgestellt und Kenndaten für

die Bewertung der Algorithmen diskutiert.

Kapitel 5 fasst die Anforderungen aus Kapitel 4 zusammen und stellt sie den bekannten Lösungsmöglichkeiten gegenüber. Daraus wird im Anschluss ein neuer Routingalgorithmus entwickelt, der die Grundlage für alle anderen in dieser Arbeit vorgestellten Optimierungen bildet. Der Algorithmus ist universell entworfen, um ihn für ein breites Einsatzgebiet offenzuhalten. Sein Aufbau ist daher noch an vielen Stellen erweiterbar, damit er an spezielle Anforderungen angepasst werden kann.

Einige mögliche Erweiterungen zur besseren Ausnutzung der verfügbaren Netzwerkkapazität werden in Kapitel 6 vorgestellt. Im ersten Schritt wird die Topologieerkennung durch ein adaptives Verfahren verbessert, wodurch sich die Bandbreitenausnutzung und die Qualität der Nachbardetektion erhöht. Im Anschluß daran wird eine Algorithmuserweiterung vorgeschlagen, die Routenberechnungen möglichst auf lokale Gruppen begrenzt. Durch dieses Verfahren wird die Bandbreitenausnutzung weiter verbessert und so der Betrieb auch größerer Ad-hoc Netzwerke ermöglicht.

Das Kapitel 7 stellt zusätzliche Erweiterungen des Routingalgorithmus vor, die paralleles Routing über mehrere Wege erlauben. Anhand von Simulationen wird gezeigt, wie sich damit die Last im Netzwerk besser verteilt und auch die Erreichbarkeit im Netzwerk gesteigert wird. Zusätzlich werden die durch die Mobilität der Stationen verursachten Verbindungsausfälle untersucht. Anschließend wird eine Möglichkeit vorgestellt, wie durch die Nutzung paralleler Routen die Auswirkungen der Verbindungsausfälle gering zu halten sind.

Kapitel 8 beschäftigt sich mit der Ankopplung von Ad-hoc Netzwerken an das Internet. Zuerst wird die im Internetstandard beschriebene Technik vorgestellt, die den Betrieb mobiler Stationen im Internet erlaubt. Danach wird ein Konzept entwickelt, das diese Technik erweitert, um auch mobile Netzwerke, insbesondere Ad-hoc Netze, anzubinden. Als weitere Aspekte werden die Sicherheit und Abrechnung der Internetanbindung untersucht.

Den Abschluss bildet Kapitel 9 mit einer Schlussbetrachtung sowie einem Ausblick auf zukünftige Entwicklungen.

Kapitel 2

Grundlagen drahtloser Kommunikation

2.1 Einleitung

Dieses Kapitel beschreibt, wie sich die Funktechnik von den Anfängen der drahtlosen Übertragung bis hin zu den Ad-hoc Netzwerken entwickelt hat. Moderne Paketfunksysteme beruhen auf Erfindungen aus mehreren Forschungsgebieten, wie beispielsweise der Funktechnik und besonders auch der Digitaltechnik. Die Abfolge der Erfindungen erklärt, wie sich aus den einfachen Funkgeräten mit der Zeit die zellularen Systeme entwickelt haben. Die Paketfunksysteme haben mit modernen zellularen Systemen sehr viel gemeinsam, beide nutzen ähnliche Verfahren zur digitalen Übertragung ihrer Daten, dennoch nimmt ihre Entwicklung einen anderen Weg. Die in dieser Arbeit vorgestellten kooperativen Paketfunknetzwerke sind ein weiterer Entwicklungsschritt weg von den zellularen Systemen.

Im zweiten Teil des Kapitels wird genauer auf die für drahtlose Datenübertragung notwendige Technik eingegangen. Die Beschreibung umfasst eine kurze Einführung in die für das Verständnis der Arbeit notwendigen physikalischen Grundlagen und erklärt das Funktionsprinzip digitaler Datenübermittlung über Funk- oder Infrarotverbindungen.

2.2 Drahtlose Kommunikationsnetzwerke

Die wissenschaftlichen Grundlagen der Funktechnik wurden 1865 von Clerk Maxwell postuliert, der die Gleichungen aufstellte, die eine elektromagnetische Wellenausbreitung vorhersagen. Diese Wellen wurden 1887 vom Physiker Heinrich Hertz dann in Versuchen nachgewiesen. Um 1897 fand der Italiener Guglielmo Marconi als erster eine Möglichkeit zur praktischen Nutzung, er baute eine Maschine, die eine drahtlose Nachrichtenübertragung ermöglichte.

Die ersten Geräte waren breitbandige Funksender, die im Bereich von 20 kHz bis 1500 kHz arbeiteten. Höhere Frequenzen wurden damals noch als unbrauchbar angesehen.

Das erste praktisch genutzte Funksystem wurde 1921 bei der Polizei in Detroit eingesetzt. Es hatte noch keine Verbindung zum Telefon, zeigte aber schon die Vorteile mobiler Kommunikation. Auch die Probleme durch die begrenzt verfügbaren Frequenzen traten bereits auf, denn zur damaligen Zeit konnten Frequenzen bis maximal 2 MHz genutzt werden. Funkamateure zeigten in den folgenden Jahren dann Möglichkeiten auf, Frequenzen oberhalb 3 MHz zu verwenden.

Das erste Telefonsystem für die Nutzung in Autos wurde 1946 in St. Louis aufgebaut. Es verwendete drei Kanäle zum Senden und Empfangen und funktionierte nur in einem begrenzten Radius um einen einzigen Funkturm. Die Technik ähnelte dem bekannten CB-Funksystem. Das System wurde nur so erweitert, dass eine Handvermittlung ins Telefonnetz möglich war.

Solange die Signale analog übertragen werden, belegt jedes Gespräch einen Frequenzbereich, und die Anzahl der gleichzeitig vermittelbaren Gespräche ist durch den nutzbaren Frequenzbereich limitiert. Obwohl eine verbesserte Technik inzwischen die Nutzung höherer Frequenzen (in diesem Fall 150 MHz) erlaubte, war der Bedarf wesentlich größer als die zur Verfügung stehenden Frequenzen. Bereits eine kleine Zahl von Nutzern konnte diese Grenze leicht erreichen. Deswegen war die Nutzung relativ teuer und besonderen Organisationen wie z.B. der Polizei vorbehalten.

2.2.1 Zellulare Netzwerke

Um 1947 wurde vorgeschlagen, das Versorgungsgebiet in Zonen (Zellen) aufzuteilen und einen Frequenzbereich mehrfach in weit voneinander entfernten Zonen wiederzuverwenden. Die Zonen müssen allerdings so weit auseinanderliegen, dass auch bei Überreichweiten, die durch besondere Wetterverhältnisse entstehen können, die Signale nicht von einer Zone in die nächste hinüberstrahlen und dort Störungen verursachen. Die Planung der Zonen muss die Landschaft und die Funkausbreitung bei unterschiedlichen Wetterlagen berücksichtigen. Da hier viele Probleme zu lösen waren, dauerte es bis 1978, bis das erste Zellenfunksystem in der Praxis realisiert werden konnte.

Das Zellenkonzept [Meh94] vervielfachte die verfügbaren Kanäle, da durch die nun mögliche Wiederverwendung eine mehrfache Ausnutzung einer Frequenz an unterschiedlichen Orten möglich war. Abbildung 2.1 zeigt idealisiert, wie eine Mehrfachnutzung der Frequenzen erfolgen kann. In der Abbildung ist der verfügbare Frequenzbereich in drei Gruppen aufgeteilt und wird in voneinander entfernten Zellen immer wieder verwendet. In Zellen mit besonders hoher Auslastung konnte das Verfahren durch Einführung neuer Gruppen nochmals angewendet werden. So entstanden Mikrozellen, die abermals die Gesamtzahl der nutzbaren Kanäle erhöhten [Ben96]. Wurde ein Versorgungsgebiet in Zellen aufgeteilt, entstanden Probleme beim Überqueren einer Zellgrenze. Ein Nutzer konnte nur innerhalb einer Zelle telefonieren, beim Verlassen des Gebietes brach die Verbindung ab. Es war auch schwierig einen Nutzer anzurufen, da sein Aufenthaltsort (seine Zelle) bekannt sein musste.

Die frühen zellularen Systeme wie das MJ (1964) oder MK (1969) System in den USA oder das A- und das B-Netz in Deutschland boten den Nutzern keine Hilfe bei diesen Problemen.

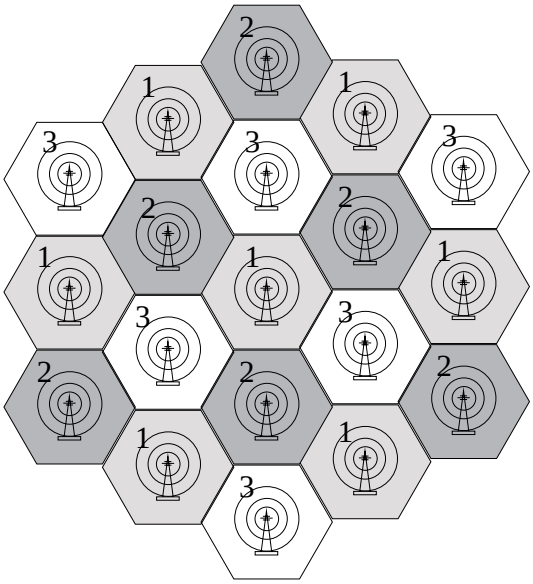


Abbildung 2.1: Grundstruktur eines Zellularnetzwerks

Die Firma Bell entwickelte ab 1960 ein Verfahren für die Weiterleitung von Gesprächen über Zellgrenzen hinweg (handoff) und das automatische Auffinden von Nutzern (locating). Daraus entstanden die Mobilfunknetze AMPS (Advanced Mobile Telphone System) in Chicago und ARTS (American Radio Telephone Service) in Washington, die ab 1983 auch für die Öffentlichkeit verfügbar waren.

Die skandinavischen Länder waren in der Weiterentwicklung der Mobilfunknetze besonders aktiv, denn die Technik ist besonders geeignet, um ein dünnbesiedeltes Land mit einer Kommunikationsinfrastruktur zu versorgen. Skandinavien begann mit dem Betrieb des NMT (Nordic Mobile Telephony) Systems 1981. Das System wurde von Ericsson entwickelt, hieß anfangs NMT450 und arbeitete im Frequenzbereich zwischen 450 und 470 MHz. Das System war sehr erfolgreich, und daher waren die Zellen oft überlastet. Deswegen wurde bald ein weiterer Frequenzbereich für die Nutzung durch ein Telefonsystem freigegeben. So entstand das NMT900, das die gleichen Eigenschaften besaß, aber zwischen 860 und 960 MHz arbeitete. Dieses System bot auch erstmals eine systemübergreifende Nutzbarkeit der Handgeräte (roaming). Damit war mit einem Gerät das Führen von Gesprächen im gesamten Gebiet der nordischen Länder möglich.

In Deutschland wurde in den 80er Jahren das C-Netz aufgebaut. Es war das fortschrittlichste System zu dieser Zeit, die Signalisierung erfolgte bereits digital, nur die Gespräche selbst wurden noch analog übertragen. Aber das C-Netz bot im Gegensatz zur Konkurrenz eine Verschlüsselung an, um das Mithören zu erschweren. Im C- Netz wurde auch erstmals eine Technik eingesetzt, die die Sendeleistung der Stationen bei Bedarf verstellt. Damit konnten die mobilen Stationen Energie sparen und die Basisstationen die Störungen verhindern, die durch Überreichweiten entstehen. Das C-Netz wurde zuerst für einen Einsatz in 450 MHz Band entwickelt. Als die Leistungsgrenze erreicht war, wurde das System erweitert, um auch im 900 MHz Band wei-

tere Kapazitäten zu nutzen. Aber trotz aller Erweiterungen erreichte das C-Netz 1995 mit einer Million angemeldeter Teilnehmer sein endgültiges Kapazitätslimit.

Am skandinavischen System waren die Vorteile der systemübergreifenden Nutzbarkeit von Mobiltelefonen bereits erprobt worden. Die Entstehung eines internationalen Systems scheiterte bis dahin an der Frequenzvergabepolitik der einzelnen Länder. Es gelang aber, innerhalb der Europäischen Gemeinschaft eine Einigung zu erzielen.

Die Frequenzen für die nächste Generation der Mobiltelephonie in Europa wurden 1979 auf der World Administrative Radio Conference festgelegt. Auf der Konferenz wurde auch beschlossen, eine Forschungsgruppe einzurichten, die einen gemeinsamen Standard für die Zukunft entwickelt. Diese Gruppe entstand dann 1982 unter dem Dach des Council of European Posts and Telecommunications (CEPT). Sie bekam den Namen Group Speciale Mobile oder einfach GSM.

Die Entwickler von GSM hatten die schwierige Aufgabe zu lösen, ein System zu entwickeln, das so exakt spezifiziert ist, dass unterschiedliche Hersteller die nötigen Komponenten produzieren können, das aber gleichzeitig offen für eine Weiterentwicklung ist. Die Spezifikation war offensichtlich ein Erfolg, denn GSM entwickelte sich in den folgenden Jahren dann zum größten Mobilfunksystem weltweit. Im Jahr 2001 hat das System weltweit 500 Millionen Nutzer erreicht und ist in über 100 Ländern installiert.

Die bedeutendste Verbesserung des GSM-Systems war der Wechsel auf eine vollständige digitale Übertragung. Der Einsatz der Digitaltechnik eröffnete eine ganze Reihe von Möglichkeiten. Bei GSM werden die digitalen Daten komprimiert, wodurch sich mehrere Gespräche gleichzeitig über einen Kanal übertragen lassen. Zusätzlich wird eine Fehlerkorrektur durchgeführt, die kleinere Übertragungsstörungen ausblendet. Alle Daten sind verschlüsselt, das System ist so sicher, dass einige Länder, unter anderem auch die Bundesrepublik, einen speziellen Zugang für die Strafverfolgungsbehörden einsetzen ließen.

Obwohl das GSM-System seinen Erfolg weiter fortsetzen wird, findet eine neue Übertragungstechnik immer mehr Beachtung, die Code Division Multiple Access (CDMA) genannt wird. Sie ist allerdings erst in der nächsten Mobilfunkgeneration (UMTS) einsetzbar, da sie den verfügbaren Frequenzbereich vollständig anders nutzt. GSM teilt den verfügbaren Frequenzbereich in feste Kanäle auf, dagegen nutzt CDMA Sendungen mit unterschiedlichen Codierungen um die Kanäle voneinander zu trennen. Ein anschauliches Beispiel für die Funktionsweise von CDMA ist ein Saal voller Personen, die gleichzeitig miteinander reden und sich trotz der Störungen gegenseitig verstehen. Die Besonderheit des Verfahrens liegt in der dynamisch aufteilbaren Bandbreite. Die insgesamt verfügbare Kapazität kann damit, je nach Bedarf, auf wenige Hochleistungskanäle oder viele Kanäle geringer Kapazität verteilt werden.

2.2.2 Paketfunk

Parallel zur Entwicklung der Funktelephonie wurde auch die Datenübertragung per Funk weiterentwickelt. Die Anforderungen beim Datenfunk sind mit den Anforderungen in den draht-

losen Telefonsystemen nicht vergleichbar. In einem Telefonsystem müssen die Sprachinformationen in sehr kurzer Zeit übertragen werden, die Qualität der übertragenen Information spielt nur eine nebensächliche Rolle. Beim Datenfunk ist die Dauer der Übertragung nicht besonders kritisch, dagegen haben verfälschte oder unvollständige Daten oft katastrophale Auswirkungen. Bei der Entwicklung des Datenfunks entstanden deshalb viele neuartige Verfahren, um den besonderen Anforderungen zu entsprechen.

Der Urahn aller Paketfunksysteme entstand an der Universität von Hawaii [Tan92]. Das System wurde aufgebaut, um Rechner, die auf vier Inseln verstreut waren, zu verbinden. Jede Station wurde mit einem Sender/Empfänger ausgerüstet, der eine ausreichende Reichweite besaß, um sich mit dem maximal 30 Kilometer entfernten Sender/Empfänger des Rechenzentrums zu verständigen.

Es wurden zwei Frequenzbänder benutzt, eines für Sendungen an das Rechenzentrum und eines für die Sendungen vom Rechenzentrum zu den Stationen. Das System erlaubt aus diesem Grund nur die Kommunikation von den Außenstellen zum Rechenzentrum, nicht aber zwischen den Stationen. Die beiden getrennten Kanäle wurden eingeführt, da ein grundsätzlicher Unterschied zwischen dem beim Rechenzentrum ankommenden und dem abgehenden Verkehr bestand. Beim eingehenden Verkehr konkurrieren mehrere Sender um den Kanal, es kann folglich zu Kollisionen kommen. Da so Pakete verloren gehen, wird jedes korrekt eingegangene Paket vom Rechenzentrum bestätigt. Das Rechenzentrum hat jedoch einen exklusiven Sender und kann damit die Bestätigungen kollisionsfrei übermitteln. Dies war die Geburt des ALOHA Protokolls. Routing war in diesem System noch kein Thema, da eine Weitervermittlung zu anderen Stationen gar nicht gebraucht wurde.

Später entwickelte sich ein Paketfunknetzwerk zwischen Amateurfunkstationen, dessen Protokoll unter dem Namen AX.25 bekannt wurde [KPD85]. Dieses Protokoll beschreibt ein Paketformat, das unter anderem ein Adressfeld, ein Steuerfeld und ein Datenfeld enthält. Das Netzwerk vermittelt Pakete zwischen den Amateurfunkstationen, die als Repeater (Wiederholer) arbeiten. Bei dem auch „Store and Forward“ genannten Prinzip besitzen alle Stationen einen permanent laufenden Empfänger und nehmen jedes Paket entgegen, das sie empfangen können. Die empfangenen Pakete enthalten alle Informationen, damit die Stationen entscheiden können, ob und wohin Pakete weitergeleitet werden müssen.

Jeder Amateurfunker hat eine eindeutige Kennung, die ihm vom Amateurfunkverband zugewiesen wird. Das Adressfeld eines AX.25 Paketes enthält eine geordnete Liste der Kennungen aller Stationen, die dieses Paket weiterleiten sollen. Falls eine Station ein Paket weiterleitet, markiert sie ihre Rufkennung in diesem Paket und macht damit deutlich, dass dieses Paket bereits bearbeitet ist. Damit wird verhindert, dass die Station das Paket nochmals aussendet, wenn sie die Wiederholung der nachfolgenden Station empfängt. Diese Vorgehensweise ist der Urahne der Routingverfahren in drahtlosen Netzwerken.

2.2.3 Ad-hoc Netzwerke

Ad-hoc Netzwerke sind eine besondere Art von Paketfunknetzwerken. Sie unterscheiden sich von den anderen Paketfunksystemen durch den vollständigen Verzicht auf eine Infrastruktur und ihre Fähigkeit sich jederzeit selbständig neu organisieren zu können. Theoretisch lassen sich Ad-hoc Netzwerke auch mit anderen Datenübertragungsverfahren wie beispielsweise Infrarotmodulen aufbauen. Dabei geht aber durch eingeschränkte Reichweiten oder einen umständlichen Aufbau viel der Flexibilität verloren. Daher werden in der Praxis nur Ad-hoc Netzwerke auf der Basis von Paketfunksystemen genutzt.

Die drahtlosen Paketnetze entwickelten sich schnell aus dem Forschungssektor hinaus. Besonders das Militär war sehr an Kommunikationsnetzen für mobile Stationen interessiert, die sich durch eigenständige Organisation den jeweiligen Gegebenheiten anpassten. Eine militärisch sehr vorteilhafte Eigenschaft ist auch die daraus resultierende Toleranz gegenüber dem Ausfall einzelner Stationen.

Bereits 1972 startete das amerikanische Verteidigungsministerium ein Projekt zur Erforschung solcher Netzwerke. Das Ziel des ersten Programms war die Erforschung der Anwendbarkeit von Packet Radio für die militärische Nutzung. Dabei wurden Grundlagen der Funktechnik und einfache Routingalgorithmen für kleinere Funknetzwerke getestet.

Um 1983 wurde dann das Nachfolgeprojekt SURAN gestartet, das sich auf die Entwicklung von verbesserten Netzwerksystemen konzentriert. Es wurden Netzwerke mit einer größeren Anzahl von Stationen und für schnell bewegte Stationen entwickelt. Im Rahmen dieses Programms wurden einige Experimentalnetze aufgebaut, wobei unter anderem auch der Landmark Routingalgorithmus entstand, der noch ausführlich im Abschnitt 3.3.10 beschrieben wird.

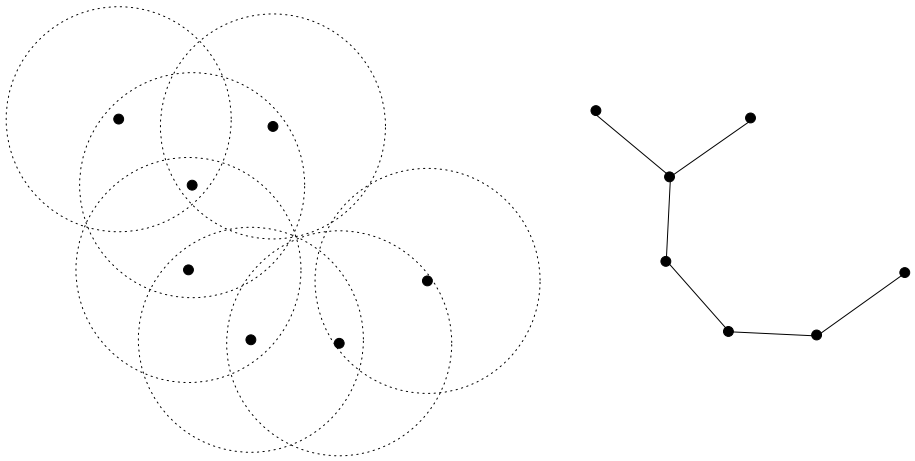


Abbildung 2.2: Grundstruktur eines Relay-Paketfunknetzwerks

Im Gegensatz zum Zellularfunknetzwerk, das in Abbildung 2.1 dargestellt ist, sind alle Stationen in Ad-hoc Netzwerken gleichzeitig Endgeräte und Vermittler (Router) der Kommunikationsdaten. Dadurch entfällt die Unterscheidung zwischen der Netzwerkinfrastruktur und den

Nutzern, die nun in einer Station vereinigt sind. Abbildung 2.2 zeigt eine idealisierte Darstellung eines Ad-hoc Funknetzwerks. Im linken Teil der Abbildung sind die Stationen als Punkte dargestellt und die Reichweiten der Funkgeräte als Kreise eingezeichnet. Im rechten Teil ist die daraus entstandene Netzwerktopologie dargestellt. Die Kanten zwischen den Stationen stellen die möglichen Wege dar, die ein Paket nehmen kann.

Da sich die Positionen der Stationen jederzeit verändern können, ist eine Ad-hoc Netzwerktopologie ein dynamischer Graph, der durch die Position der Stationen, der Reichweite der Sender und den Umgebungsparametern, die auf die Reichweite Einfluss nehmen, bestimmt ist. Im Kapitel 4.4.1 wird ein Simulationssystem für Ad-hoc Netze vorgestellt. Dabei werden auch die Umgebungsparameter genauer untersucht, die auf die Topologie und damit auf das Routing Einfluss nehmen. Das Routing wird allerdings nicht nur von der Topologie bestimmt, auch die Kommunikationsverbindungen zwischen den Stationen besitzen begrenzende Eigenschaften wie beispielsweise Fehlerraten und Verzögerungszeiten, die berücksichtigt werden müssen. Bevor jedoch näher auf das Routing eingegangen wird, muss der generelle Aufbau eines Rechnernetzwerks erläutert werden. Der nächste Teil dieses Kapitels widmet sich deswegen dem Aufbau und zeigt, welche Algorithmen und Protokolle zum Betrieb eines Paketfunknetzwerks notwendig sind.

2.3 Drahtlose Paketfunknetze

In Kommunikationsnetzwerken wird prinzipiell wird zwischen verbindungsorientierten und paketorientierten Vermittlungstechniken unterschieden. Die verbindungsorientierten Netze wurden ursprünglich für den Fernsprechverkehr entwickelt, während die paketorientierten Netze sich erst wesentlich später entwickelten und vorwiegend für den Datenverkehr eingesetzt wurden.

In verbindungsorientierten Netzwerken werden Ressourcen wie z.B. verfügbare Leitungen oder Vermittlungseinheiten reserviert, sie bleiben für die Dauer einer Verbindung bestehen. Dies sorgt dafür, dass für eine Verbindung keine großen Verzögerungen in der Übertragung auftreten können, allerdings entstehen bei der Reservierung von Kapazitäten immer Verluste, wenn sich der Bedarf nicht genau festlegen lässt. Wird die Kapazität für den Fall der maximalen Auslastung reserviert, dann stehen die nicht belegten Ressourcen auch niemandem sonst zur Verfügung.

Eine gegensätzliche Strategie verwenden die paketorientierten Netzwerke, sie sind hauptsächlich für den Datenverkehr ausgelegt und tolerieren daher Verzögerungen bei der Übertragung wesentlich leichter. Durch die Zerlegung der Daten in einzelne Datenpakete oder Datagramme, die getrennt voneinander übertragen werden, entsteht eine Reihe von Vorteilen. Die Ressourcen werden bei der Paketvermittlung jeweils nur für die Dauer der Übertragung eines Paketes belegt. Das ermöglicht eine zeitversetzte mehrfache Belegung einer Ressource. Mehrere Nutzer können quasi-parallel über eine Ressource verfügen oder ein Nutzer kann parallel mit mehreren Zielen kommunizieren. Da sich die Kapazität der Ressourcen durch die Mehrfachnutzung nicht erhöht,

entstehen vereinzelt Wartezeiten, bis die benötigten Ressourcen frei sind.

Paketorientierte Übertragungen erlauben eine Veränderung der Routen während der Übertragung. Es ist daher möglich, dass zwei Pakete unterschiedliche Wege zum Ziel nehmen. Dadurch wird die Flexibilität und die Fehlersicherheit eines Netzwerkes stark verbessert. Besonders die Fähigkeit der dynamischen Anpassung von Routen machen die paketorientierte Übertagung zu einem geeigneten Verfahren für die Mobilkommunikation.

Ein weiterer Vorteil von Paketnetzwerken ist die Möglichkeit, einen Mischbetrieb zwischen den oben erwähnten reservierenden und nichtreservierenden Systemen zu nutzen. Dies geschieht einfach durch eine Garantie, dass eine Ressource einen gewissen Anteil der Zeit nur exklusiv für Pakete zwischen bestimmten Stationen genutzt wird. Damit entsteht eine Verbindung zwischen den Stationen, die ähnliche Eigenschaften wie in verbindungsorientierten Netzwerken besitzt und deshalb auch virtuelle Verbindung genannt wird.

Da die Paketnetzwerke mehr Flexibilität bieten, ohne die Möglichkeiten verbindungsorientierter Kommunikation aufzugeben, haben sich auch die Ad-hoc Netzwerke hauptsächlich auf der Basis von Paketnetzwerken entwickelt. Es wurden auch bereits die Möglichkeiten von verbindungsorientierten Ad-hoc Netzwerken näher untersucht, die Ergebnisse dieser Arbeit über Digital Relay Inter-Communication (DIRC) sind in einer weiteren Dissertation [Bor02] zusammengefasst und werden daher hier nicht näher behandelt.

In einem Funknetzwerk ist die entscheidende Ressource die Funkbandbreite, deren Kapazität möglichst effektiv genutzt werden muss. Dazu existieren Systeme, die mit Reservierungen oder paketbasiert arbeiten. Es existieren praktisch alle Varianten, beginnend mit sehr statischen Systemen, die mit einem genau vorgegebenen Plan ihre Frequenzen und Routen nutzen und endet bei hochflexiblem Systemen, die ihre Möglichkeiten selbständig erkunden. Die Ad-hoc Netzwerke sind Prototypen für die selbständigen Systeme.

Da Datenfunk eine Vielzahl von Technologien umfasst, ist hier nur eine Beschreibung der wichtigsten Techniken der Datenfunkübertragung möglich, für eine tiefergehende Beschreibung dieses großen Themengebietes und deren Standards sei hier das Buch „Data over Wireless Networks“ [Hel01] empfohlen.

Da sich diese Arbeit auf das Routing konzentriert, beschränken sich die im folgenden vorgestellten Netzwerktechniken auf Verfahren für die Implementierung von drahtlosen, paketvermittelnden Netzwerken mit mobilen Stationen und niedrigen Übertragungsraten, die auf gemeinsam genutzten Kanälen zur Verfügung stehen. Dazu wird im ersten Schritt die Standardarchitektur von Netzwerken eingeführt, wie sie beispielsweise durch das OSI-Modell dargestellt wird. Aus dem Modell sind direkt die Netzwerkkomponenten ersichtlich, die mit dem Routing zusammenarbeiten. Diese Komponenten werden anschließend einzeln vorgestellt.

2.3.1 OSI-Modell

Die Datenübertragung durch ein Paketfunksystem ist ein komplexer Vorgang, der durch mehrere aufeinander aufsetzende Komponenten ausgeführt wird. Routing ist dabei nur eine Komponente eines Paketfunksystems. Für ein funktionierendes System sind weitere Komponenten wie z.B. ein Transportprotokoll oder ein Paketübertragungsprotokoll notwendig. Eine Beschreibung aller notwendigen Komponenten und deren Untergliederung findet sich im OSI-Modell [Tan96] oder im daraus abgeleiteten TCP-IP Referenzmodell [Tan96, PD00], das speziell die Verhältnisse im Internet wiederspiegelt.

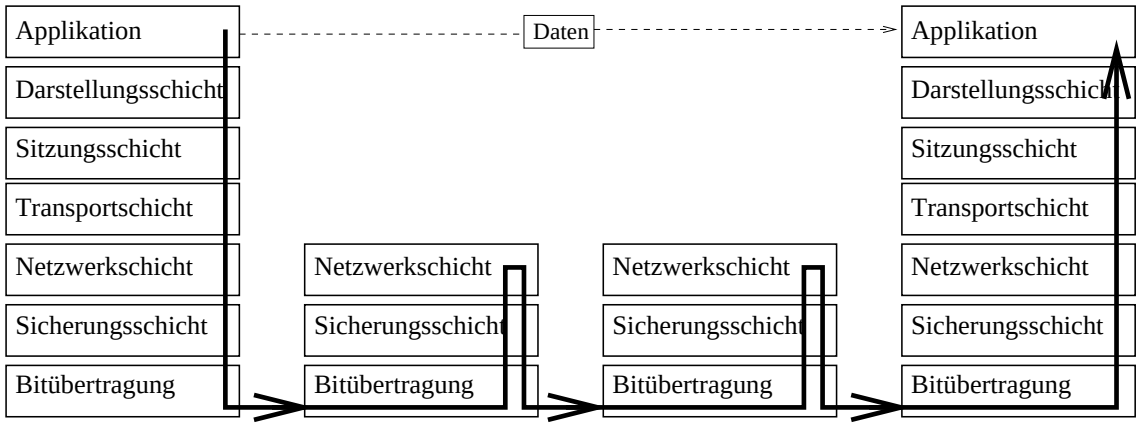


Abbildung 2.3: Das OSI-Schichtenmodell

Die Schichten des OSI-Modells sind in Bild 2.3 dargestellt. Der dick eingetragene Pfeil zeigt den Datenfluss durch die einzelnen Schichten, wobei die Applikation auf der linken Seite Daten über zwei Vermittler an die rechte Applikation versendet. Im OSI-Modell ist Routing die Aufgabe des Netzwerkprotokolls und findet in der dritten Schichtebene statt. Die oberste Schicht ist die Applikation, die einen Kommunikationsdienst benötigt. Die Applikation sendet nun Daten durch die Darstellungsschicht und die Sitzungsschicht, die z.B. eventuell notwendige Datenkonvertierungen übernehmen, an die Transportschicht. Die Transportschicht hat mehrere Aufgaben, sie zerlegt Datenströme in versendbare Pakete, bestimmt die Sendegeschwindigkeit der Pakete und korrigiert Fehler. Die dort erzeugten Pakete werden dann an die dritte Schicht zur Auslieferung übergeben. Die Aufgabe der Netzwerkschicht ist der Weitertransport, eventuell über mehrere Vermittler, bis an die Zielstation. Dazu muss die Netzwerkschicht das Paket an die zweite Schicht, die Sicherungsschicht, übergeben. Die Sicherungsschicht ist nur für den Weitertransport bis zum nächsten Nachbarn verantwortlich. Muss ein Paket über mehrere Vermittler weitergeleitet werden, dann wird das Paket in den Vermittlern jeweils über die Sicherungsschicht empfangen, an die dritte Schicht weitergereicht, und dort wird dann der nächste Vermittler bestimmt und das Paket wieder durch die Sicherungsschicht weitergeleitet, bis das Ziel erreicht ist. Dort angekommen, wird das Paket dann an die Transportschicht zurückgegeben. Die Transportschicht am Ziel setzt die empfangenen Pakete zusammen und übergibt die Daten dann durch die Sitzungsschicht und die Darstellungsschicht zurück an die Applikation.

Die Netzwerkschicht und der darin enthaltene Routingalgorithmus erhält folglich die Aufträge

(Datenpakete) von der übergeordneten Schicht und muss daraus einen oder mehrere Aufträge an die untergeordnete Schicht weiterleiten. Da die Art der Daten und ihre Darstellungsform für das Routing kaum von Bedeutung sind, werden in den folgenden Abschnitten zuerst die Transportschicht und anschließend die Sicherungs- und Bitübertragung erklärt.

2.3.2 Die Transportschicht

In paket-vermittelnden Systemen haben sich zwei Protokolle in der Transportschicht etabliert, die als „Transport-Control-Protocol“ (TCP) und „User-Datagram-Protocol“ (UDP) bezeichnet werden.

Das UDP-Protokoll bietet praktisch keine Dienstleistung, es versendet nur Daten, die in ein Paket passen, und kontrolliert auch nicht die korrekte Auslieferung der Daten, deswegen kann es auch keine Fehler korrigieren.

Dagegen bietet das TCP-Protokoll mehr Dienstleistungen, TCP nimmt einen beliebig langen Datenstrom von der oberen Schicht entgegen, zerlegt diesen in passende Pakete und rekonstruiert auf der Empfängerseite dann wieder den kompletten Datenstrom. Dazu gehört auch eine aufwändige Fehlerkorrektur, wenn Pakete fehlen oder in der falschen Reihenfolge angeliefert werden. Neben der Erkennung und Beseitigung von Fehlern muss dieses Protokoll auch eine angemessene Sendegeschwindigkeit für die einzelnen Pakete ermitteln und nach Bedarf auch korrigieren.

Die Konstruktion eines Transportprotokolls in der Art von TCP ist eine langwierige Aufgabe, da durch die ineinandergreifenden Regelmechanismen zur Fehlerkorrektur und Geschwindigkeitssteuerung (siehe auch [PD00]) ein langer Optimierungsprozess notwendig ist. Das dabei entstehende Protokoll ist dann auf eine bestimmte Netzwerktechnologie hin optimiert. Daraus ergibt sich auch die Hauptschwierigkeit für den Einsatz von TCP in drahtlosen Netzwerken, denn TCP ist auf die im Internet gängigen leitungsgebundenen Systeme angepasst. So ist für TCP beispielsweise ein Paketverlust immer ein Zeichen von Überlastung und führt damit automatisch zur sofortigen Geschwindigkeitsreduktion beim Senden. In drahtgebundenen Systemen geht ein Paket äußerst selten durch eine Signalstörung verloren, diese Tatsache gilt allerdings nicht für Funknetzwerke. Es wurden bereits Arbeiten zu speziellen TCP-Konfigurationen für Funknetzwerke veröffentlicht [BB95, ABSK95, BB97], die einen höheren Durchsatz versprechen, allerdings muss die Kompatibilität zwischen den im Internet eingesetzten TCP-Systemen und möglichen Neuentwicklungen gewahrt bleiben. Solange drahtlose Netzwerke nur ein kleines Marktsegment belegen, wird sich hier kein eigener Standard etablieren, vielmehr werden die unterliegenden Schichten so erweitert, dass sie das gleiche Verhalten wie die gängigen Internetsysteme zeigen. Das ist besonders gut am weiter unten beschriebenen IEEE 802.11 Standard zu erkennen, dessen aufwendige Korrekturmaßnahmen das Verhalten einer Funknetzwerkverbindung möglichst wie ein drahtgebundenes Verfahren erscheinen lässt.

2.3.3 Drahtlose Datenübertragung

Die meisten Rechnernetze arbeiten leitungsgebunden, sie verwenden Kupferdrähte oder Glasfasern zum Informationstransport. Eine Übertragung über Laser, Infrarotsender, Mikrowellen oder Radiosender benötigt keine spezifische feste Infrastruktur. Solche drahtlosen Übertragungssysteme schicken ihre Daten durch ein überall frei verfügbares Medium [1]. Die drahtlosen Übertragungssysteme unterscheiden sich in allen wichtigen Kenndaten wie z.B. der nutzbaren Bandbreite oder der Störungswahrscheinlichkeit von den leitungsgebundenen Systemen.

Es haben sich mittlerweile viele speziell auf diese Eigenschaften angepasste Multiplexverfahren, Zugriffssteuerungen, Fehlerschutz- und Fehlerkorrekturprotokolle entwickelt. Alle Möglichkeiten zur drahtlosen Datenübertragung aufzuzählen würde den Rahmen dieser Arbeit sprengen, eine gute Einführung in digitale drahtlose Datenübertragung bietet das Buch „Wireless LAN Systems“ [SLH94].

Unabhängig davon, welches drahtlose Verfahren eingesetzt wird, zeigen alle gegenüber den leitungsgebundenen Systemen eine deutlich höhere Fehlerwahrscheinlichkeit in der Datenübertragung. Diese Eigenschaft ist durch das genutzte Medium verursacht. In leitungsgebundenen Netzen wird viel Mühe darauf verwendet, äußere Umwelteinflüsse abzuschirmen. In drahtlosen Systemen ist dies nicht möglich, hier wird dann versucht, die unvermeidbaren Fehler z.B. durch eine Fehlerschutzcodierung zu bekämpfen.

Trotz aller Bemühungen ist das Medium immer die begrenzende Ressource in drahtlosen Netzwerken. So müssen sich auch alle Stationen innerhalb ihrer Reichweite die im Medium verfügbare Bandbreite teilen. Versuchen zwei Stationen gleichzeitig ein Medium zu nutzen, überlagern sich die Signale und werden damit unbrauchbar [2].

Funkmedien haben immer diese Broadcasteigenschaft; alle Nachbarn innerhalb der Sendereichweite empfangen die Aussendungen einer Station. Das ist von Nachteil, wenn das exakte Verhalten eines leitungsgebundenen Systems mit einem Funksystem nachgebildet werden muss. Es kann aber auch Vorteile bringen, wenn schon bei der Entwicklung eines Netzwerks die Broadcasteigenschaft berücksichtigt wird. So sendet beispielsweise der in 3.2.1 vorgestellte verteilte Distanz-Vektor Algorithmus seine Botschaften immer an alle Nachbarn. Da ein Funknetzwerk diese Operation effizient erledigen kann, hat der DV-Algorithmus dadurch Vorteile gegenüber den anderen Algorithmen.

Auch das „passive Acknowledge“ nutzt die Möglichkeit aus, die Sendungen der Nachbarn mitzuhören. Wird ein Paket an einen Nachbarn zur Weitervermittlung gesendet, dann braucht der Nachbar den Empfang des Paketes nicht zu bestätigen, wenn er es unmittelbar weitersenden kann. Die Weitersendung wird auch vom Vorgänger empfangen und dient damit als indirekte Bestätigung.

[1] Teilweise erheben staatliche Organisationen Ansprüche auf ein Medium, eine Funkfrequenz muss immer lizenziert werden

[2] Das CDMA Verfahren arbeitet mit gezielten Überlagerungen, ist allerdings in Ad-hoc Netzwerken wegen der komplizierten Sendeleistungskontrolle nicht verwendbar

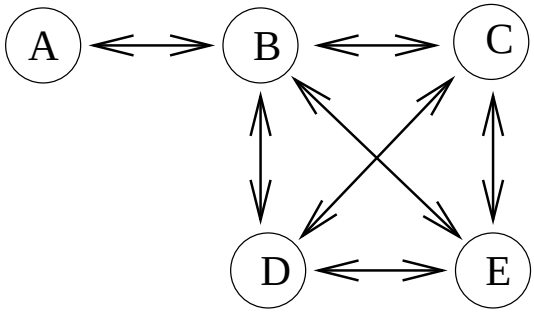


Abbildung 2.4: Netzwerk mit einem „Hidden-Terminal“

Aber die Broadcasteigenschaft des Mediums führt auch zu Problemen, die besonders bei ungünstiger Positionierung der Stationen auftreten. Das Bild 2.4 zeigt ein kleines Netzwerk mit den Stationen *A* bis *E*, bei dem die Station *A* nur über *B* erreichbar ist. Die Station *A* wird als „hidden Terminal“ bezeichnet, denn sie kann weder erkennen, ob eine andere Station zu *B* sendet, noch können die anderen Stationen erkennen, ob *A* sendet. So treten Kollisionen gehäuft bei der Station *B* auf. Das ist nachteilig, weil die Station *B* die meisten Nachbarn besitzt und deswegen besonders häufig als Vermittler arbeiten muss.

2.3.4 Medien-Zugriffsverfahren

Netzwerke verwenden entweder eine Punkt-zu-Punkt Verbindung oder sie benutzen ein Mehrfachzugriffsverfahren. Bei einer Punkt-zu-Punkt Verbindung gibt es nur einen Sender und einen Empfänger, die Nutzung des Mediums ist klar geregelt. Schwierigkeiten entstehen jedoch schon bei der Planung, z.B. bei der Kanalvergabe an die einzelnen Sender. Bei einer festen Zuteilung sind viele Kanäle nicht genutzt, während andere überlastet werden. Bei einem Mehrfachzugriffsverfahren steht die gesamte Kapazität allen Stationen zur Verfügung. Allerdings muss die Verteilung der Kapazität auf die Stationen exakt gesteuert sein, ansonsten entstehen Kollisionen und wertvolle Übertragungskapazität geht verloren.

In Funksystemen ist eine Kollision besonders unangenehm, da die beteiligten Stationen die Kollision nicht erkennen können. Dies wird durch den sogenannten „Capture Effect“ verursacht, durch den ein Empfänger sich immer auf den stärksten Sender einstellt. Daher empfängt eine Station, die ihren Sender aktiviert, nur noch ihre eigenen Signale und kann die Signale anderer Sender nicht mehr wahrnehmen. Aus diesem Grund zerstört eine Kollision nicht nur die Übertragung der gesendeten Daten, denn das Medium wird auch blockiert, bis beide Datensätze vollständig gesendet sind.

Die Sender müssen sich beim Mehrfachzugriffsverfahren ein Übertragungsmedium teilen, das nur exklusiv benutzt werden kann. Die Zugriffskontrolle hat deshalb einen entscheidenden Einfluss auf die erzielbare Datenübertragungsrate in einem Medium. Aus diesem Grund werden nun zwei gebräuchliche Zugriffskontrollverfahren vorgestellt.

Carrier Sense Multiple Access

Kollisionen durch gleichzeitiges Senden sind einfach zu vermeiden, wenn vor dem Senden das Medium auf einen bereits vorhandenen Sender geprüft wird. Genau diese Strategie nutzt das „Carrier Sense Multiple Access“ (CSMA) Verfahren.

Die Probleme ergeben sich dann nur noch durch gleichzeitiges Starten von mehreren Sendern. Dieser Fall tritt häufig auf, denn solange das Medium belegt ist, gehen alle Sendewilligen in einen Wartezustand und beginnen nach Freiwerden des Mediums mit dem Senden. Um dieses Problem zu reduzieren, werden variable Wartezeiten nach dem Freiwerden des Mediums eingefügt. Dazu wählt jede Station zufällig einen Startzeitpunkt aus einem vorgegebenen Wartezeitraum aus. Die Wartezeit hat einen erheblichen Einfluss auf die Kollisionswahrscheinlichkeit: Mit einer kleinen Wartezeit gibt es bei vielen Sendewilligen oftmals zwei Stationen, die den selben Startzeitpunkt wählen und dadurch eine Kollision verursachen. Wird der Wartezeitraum zu groß gewählt, entstehen zwischen den Übertragungen unnötig große Pausen.

In [Tan92] und in den Simulationsergebnissen von Kapitel 4.5 finden sich genaue Studien zum Durchsatz von CSMA. Dabei zeigt sich, dass bei einer wachsenden Anzahl von sendewilligen Stationen der anfängliche Durchsatz von 60 Prozent der Maximalkapazität des Mediums immer weiter absinkt, bis praktisch keine Kommunikation mehr möglich ist. Aus diesem Grund war eine Weiterentwicklung dringend erforderlich, die schließlich zur Standardisierung eines Verfahrens unter der Bezeichnung IEEE 802.11 [LKBP96] führte.

Zugriffskontrolle unter IEEE 802.11

Da CSMA erhebliche Durchsatzprobleme bei großen Senderzahlen hat und auch sehr Anfällig für das Hidden-Terminal Problem war, wurde ein verbessertes Verfahren entwickelt, das erstmals unter dem Kürzel MACA [Kar90] veröffentlicht wurde und nach weiteren Verbesserungen unter dem Namen IEEE 802.11 [Dep97] der Netzwerkstandard für drahtlose Paketfunknetze wurde.

Das besondere an diesem Verfahren ist das Protokoll, mit dem die beteiligten Stationen aushandeln, wer als nächstes senden darf. Dafür gibt es zwei Botschaften: mit der Sendeanfrage (Request to Send - RTS) zeigt eine Station ihre Absicht an, mit der Antwort (Clear to Send - CTS) gibt der Empfänger seine Zustimmung zum Senderstart. Die RTS- und CTS-Botschaften werden mit dem CSMA Verfahren gesendet und sind dadurch wieder einer Kollisionsgefahr ausgesetzt, aber durch die kleinen Botschaften und eine intelligente Regelung der Wartezeiten ist der Verlust relativ gering. Das Hidden-Terminal Problem ist durch das Aushandeln ebenfalls gelöst, da nun alle Stationen in Sendereichweite des Empfängers das CTS empfangen und daher wissen, dass sie nun nicht mehr senden dürfen. Das Verfahren funktioniert jedoch nur bei der Kommunikation zwischen *zwei* Stationen. Im Falle eines Broadcasts, bei dem die Anzahl der Ziele nicht bekannt ist, macht ein RTS-CTS Handshake keinen Sinn, daher wird für Broadcasts wieder auf CSMA zurückgegriffen. Aus diesem Grund ist die Verteilung von Informationen per Broadcast nicht sonderlich zuverlässig.

Der Durchsatz von IEEE 802.11 wurde in [WWW96, WSFW97] und mit dem selbst entwickelten Simulator untersucht (siehe Kapitel 4.5). Durch das Aushandeln entsteht ein Overhead vor jedem Senden. Dieser fällt aber nur bei sehr vielen kleinen Paketen ins Gewicht, bei großen Paketen erreicht das Verfahren einen Durchsatz bis zu 90 Prozent der möglichen Datenrate.

2.4 Begriffsdefinitionen zur Graphentheorie

Für die Beschreibung der Routingalgorithmen und deren Optimierungsmöglichkeiten werden in den nachfolgenden Kapiteln oft Beispiele verwendet. Die dort verwendeten Begriffe sind leider nicht immer eindeutig. Aus diesem Grund sind hier die wichtigsten Begriffe kurz erläutert. An die Begriffserklärung schließt sich ein Abschnitt an, der sich den graphentheoretischen Begriffen widmet, dazu werden dann jeweils die notwendigen mathematischen Definitionen angegeben.

Begriffe

- *Station*: Eine Station ist ein technisches Gerät, welches mit anderen Stationen kommuniziert aber nicht zwangsläufig als Vermittler tätig ist.
- *Leitung*: Eine Kommunikationsverbindung zwischen zwei Stationen. In einem Funknetz gibt es die herkömmliche Leitung in der Form eines Drahtes nicht mehr. Im Funknetz wird daher jede direkte Kommunikationsmöglichkeit durch eine virtuelle Leitung modelliert.
- *Leitungskosten*: Die Leitungskosten werden zur Modellierung der Kommunikationskosten benötigt. Im einfachsten Fall wird jeder Leitung ein Kostenwert von 1 zugewiesen, der üblicherweise auch „ein Hop“ genannt wird. Es lassen sich auch beliebige andere Metriken wie z.B. Länge oder Übertragungszeiten einsetzen.
- *Netzwerk*: Ein Netzwerk besteht aus einer Menge von Stationen und einer Menge von Leitungen, die Stationen miteinander verbinden.
- *Partition*: Eine oder mehrere Stationen, die keine Verbindung mehr zum restlichen Netzwerk besitzen.
- *Router*: Eine Station, die auch Vermittlungsdienste übernimmt.
- *Route/Pfad*: Ein Weg durch ein Netzwerk, der aus einer oder einer Aneinanderreihung mehrerer Leitungen besteht.
- *Nachbar*: Eine Station, die über eine Leitung direkt erreichbar ist.
- *Schleife*: Teilstück eines Weges, der zwei mal über die gleiche Station führt.

Graphendefinitionen

Die in dieser Arbeit angegebenen Algorithmenbeschreibungen richten sich nach den Vorgaben aus dem Algorithmenbuch „Data Networks“ [Ber92]. Die meisten Standardalgorithmen wurden auch aus dieser Quelle übernommen.

Ungerichtete Graphen

Ein ungerichteter **Graph** $G = (N, A)$ ist definiert als eine nichtleere Menge N von Knoten und einer Menge A von ungeordneten Knotenpaaren. Die Knotenpaare enthalten jeweils zwei unterschiedliche Knoten aus N und werden als **Kante** bezeichnet. Es existieren Graphendefinitionen, die Kanten mit beiden Enden am gleichen Knoten zulassen, diese Kanten werden hier explizit ausgeschlossen. Ebenso sind keine mehrfachen Kanten zwischen einem Knotenpaar erlaubt. Sind zwei Knoten (n_1, n_2) durch eine Kante verbunden, dann werden die Knoten als **Nachbarn** bezeichnet. Ein solcher Graph beschreibt exakt die Topologie eines Netzwerks, allerdings lassen sich aus einem Graph nur die Beziehungen der Knoten zueinander ablesen. Es sei erwähnt, dass jeder Graph beliebig viele Darstellungsmöglichkeiten hat, die Positionen der Knoten lassen sich aus dem Graph nicht bestimmen.

Ein **Weg** in einem Graph G ist eine Folge von Knoten $(n_1, n_2, ..., n_l)$ von denen jedes Paar (n_1, n_2),(n_2, n_3),....,(n_{l-1}, n_l) eine Kante von G ist. Ein Weg, in dem sich kein Knoten wiederholt, wird **Pfad** genannt, diese Eigenschaft wird auch als **schleifenfrei** bezeichnet. Ein Weg $(n_1, n_2, ..., n_l)$ mit $n_1 = n_l, l \geq 3$ und der von n_2 bis n_l schleifenfrei ist, wird als **Kreis oder Zyklus** bezeichnet.

Ein Graph ist **zusammenhängend**, wenn für jeden Knoten i ein Pfad
$(i = n_1, n_2, ..., n_l = j)$ zu jedem anderen Knoten j des Graphen existiert.

$G' = (N', A')$ ist dann ein **Teilgraph** von $G = (N, A)$, wenn G' die Graphendefinition erfüllt und $N' \subset N, A' \subset A$ gilt.

Wenn ein Graph G nicht zusammenhängend ist, dann existieren **Partitionen**
$(P_1, P_2, ..., P_k),\ k > 1$, die alle Teilgraphen von G sind. Jede Partition muss zusammenhängend sein und die Partitionen sind echte Teilmengen von G.

Ein **Baum** ist ein zusammenhängender Graph, der keine Zyklen enthält. Ein **Spannbaum** eines Graphen G ist ein Baum, der alle Knoten von G enthält. Der Spannbaum existiert nur, wenn der Graph G zusammenhängend ist.

Gewichtete Graphen

Jede Kante eines Graphen $G = (N, A)$ ist durch ein Knotenpaar $(i, j),\ i, j \in N$ bestimmt. Wird jeder Kante (i, j) zusätzlich ein Gewicht $W_{i,j}$ zugewiesen, dann spricht man von einem **ge-**

wichteten Graphen. Das Gewicht einer Kante repräsentiert die Kommunikationskosten für die Daten, die über eine Kante übertragen werden. Die vorgestellten Algorithmen erlauben in der Regel nur positive Fließkommazahlen zur Angabe der Kantengewichte.

Die **Distanz** d_{n_1,n_l} zwischen zwei Knoten ist das Gewicht eines Pfades $(n_1, n_2, ..., n_l)$. Sie wird aus der Summe der Kantengewichte (n_1, n_2),(n_2, n_3),....,(n_{l-1}, n_l) errechnet:

$$d_{n_1,n_l} = \sum_{i=1}^{l-1} W_{n_i,n_{i+1}}$$

Ebenso errechnet sich das Gewicht eines Graphen d_A aus der Summe der Kantengewichte aller in ihm vorhandenen Kanten:

$$d_A = \sum_{i,j \in A} W_{i,j}$$

Der **kürzeste Pfad** zwischen zwei Knoten i und j , $i, j \in N$ ist der Pfad mit dem minimalen Gewicht aus allen möglichen Pfaden von i nach j.

Ein **minimal spannender Baum** bezüglich eines Graphen G ist der Spannbaum mit dem minimalen Gewicht aus allen möglichen Spannbäumen, die aus dem Graphen G erzeugt werden können.

Gerichtete Graphen

In der Arbeit werden gerichtete Graphen nur für die Beschreibung einiger Algorithmen eingesetzt, die Manipulationen an Bäumen beschreiben. Die meisten Graphenalgorithmen lassen sich auch auf ungerichtete Graphen anwenden. Aus diesem Grund beziehen sich die im weiteren vorgestellten Algorithmen auf ungerichtete Graphen, falls die Beschreibung nicht ausdrücklich auf den Einsatz gerichteter Graphen hinweist.

Ein **gerichteter Graph** $G = (N, A)$ besteht aus einer nichtleeren Menge von Knoten und einer Menge *geordneter* Knotenpaare. Die geordneten Knotenpaare enthalten jeweils zwei unterschiedliche Knoten aus N und werden als **gerichtete Kante** bezeichnet. In einer Abbildung wird gerichteten Kanten eine Pfeilspitze hinzugefügt, die vom ersten Knoten zum zweiten Knoten des Knotenpaares zeigt. Mehrfache Kanten zwischen zwei Knoten $(i, j),\ i, j \in N,\ i \neq j$ sind hier nicht zulässig. Es ist jedoch im Graph eine Kante (i, j) und eine Kante (j, i) erlaubt, da sich die beiden Kanten durch ihre Ordnung unterscheiden.

Zu jedem gerichteten Graphen $G = (N, A)$ gibt es einen beigeordneten ungerichteten Graphen $G' = (N', A')$ in dem $N' = N$ und $(i, j) \in A'$, wenn entweder $(i, j) \in A$ oder $(j, i) \in A$ oder beides zutrifft. In einem gerichteten Graphen existiert ein Weg, ein Pfad oder ein Zyklus nur dann, wenn die Voraussetzungen für einen Weg, Pfad oder Zyklus im beigeordneten ungerichteten Graphen erfüllt sind. Ein **gerichteter Weg** $(n_1, n_2, ..., n_l)$ setzt sich nur aus gerichteten Kanten $(n_i, n_i + 1)$ des Graphen G zusammen wobei $1 < i < l - 1$. Ein **gerichteter Pfad** ist ein gerichteter Weg, in dem sich kein Knoten wiederholt, und ein gerichteter Zyklus ist ein gerichteter Weg $(n_1, .., n_l)$ mit $l > 2,\ n_1 = n_l$, in dem sich kein Knoten wiederholt.

Die **Gewichte** in einem gerichteten Graphen G sind *richtungsabhängig*, damit ist auch die Distanz $D_{i,j},\ (i,j) \in N$ richtungsabhängig. Die Berechnung erfolgt analog wie im ungerichteten Graphen.

2.5 Zusammenfassung

In diesem Kapitel wurde die historische Entwicklung der Funktechnik bis hin zur aktuellen drahtlosen Funknetzwerktechnik beschrieben. Ausgehend von den ersten Versuchen in der Funktechnik wurden die einzelnen Entwicklungsschritte der drahtlosen Kommunikation vorgestellt. Dabei zeigte sich, dass Bandbreite eine nicht vermehrbare Ressource ist, deren sparsame Nutzung z.B. durch Wiederverwendung in Funkzellen schon zu Beginn ein wichtiges Entwicklungsziel war. Auch der anschließende Übergang zur Digitaltechnik brachte neben anderen Vorteilen eine bessere Ausnutzung der Bandbreiten. Mit der Digitaltechnik begann das Zeitalter der Funknetzwerke, die bald eigenständige Wege nahmen. Beginnend mit den Paketfunksystemen der Amateurfunker entwickelten sich Netzwerke, die automatisch ihre Umgebung erfassen und sich dann selbständig jeder Situation anpassen müssen.

Zur Beschreibung der Funktionsweise moderner Netzwerke wurde anschließend das OSI-Modell erklärt und die speziellen Ausprägungen der OSI-Schichten in Funknetzwerken erläutert. Dabei wurde allerdings auf die Netzwerkschicht nicht näher eingegangen, da sich das nächste Kapitel diesem Thema ausführlich annimmt.

Anschließend wurden die in den Beispielen genutzten Begriffe und die wichtigsten Definitionen der Graphentheorie erklärt, die eine wichtige Voraussetzung für die Algorithmen und Netzwerkanalysen der folgenden Kapitel sind.

Kapitel 3

Routing in drahtlosen Netzwerken

3.1 Einleitung

Im vorigen Kapitel wurde erläutert, welche Möglichkeiten zur drahtlosen Übertragung digitaler Daten zur Verfügung stehen. Auf diesen Möglichkeiten baut die Netzwerkschicht auf, die durch Zusammenschalten der einzelnen Übertragungskanäle weiträumige Verbindungen ermöglicht.

Die Aufgabe des Routing besteht in der Bestimmung von Pfaden durch ein Kommunikationsnetzwerk von einem Sender (Quelle) zu einem Empfänger (Ziel) entsprechend den Anforderungen der Nutzer und den Dienstmöglichkeiten des Netzwerkes. Das Ziel eines guten Routing ist eine möglichst effektive Nutzung der verfügbaren Netzwerkressourcen. Durch die unterschiedlichen Anforderungen und möglichen Dienste existieren verschiedenste Metriken zur Bewertung und Optimierung einer Route. Als Metrik für die Optimierung von Routen können beispielsweise die Länge der Pfade, die Bandbreite, eine durchschnittliche Auslastung, Übertragungskosten, die Länge von Warteschlangen oder gemessene Verzögerungszeiten verwendet werden.

Die Anforderungen an das Routing sind aber noch weitergehend. Es zählt nicht nur das Berechnungsergebnis, sondern auch die Geschwindigkeit, mit der ein Ergebnis erzielt wird und insbesondere die Anzahl der Botschaften, die gebraucht werden. Ein Routing braucht für die Berechnungen Informationen, die es nur durch einen Austausch von Botschaften erhält. Damit verbraucht Routing einen Teil der Kapazitäten, die es verplanen soll, für sich selbst. Ein effektiver Routingalgorithmus muss daher auch bezüglich seiner eigenen Ressourcennutzung sparsam sein. Die unterschiedlichen Anforderungen, die sich nicht alle gleichermaßen erfüllen lassen, führen zur Entwicklung von spezialisierten Algorithmen. Die gleichen Ursachen, die zur Entwicklung dieser Algorithmen führen, erklären auch, warum nicht einfach ein Routingalgorithmus, der sich z.B. im Internet bereits bewährt hat, für Ad-hoc Netzwerke übernommen werden kann.

Die Abbildung 3.1 zeigt eine Beschreibung der Umgebungsparameter und der daraus resultie-

renden Abhängigkeiten, wie sie auch in Steenstrups Buch [Ste95] zu finden sind. Im oberen Teil sind die besonderen Voraussetzungen wie dynamisches Netzwerk, geringe Bandbreite und die mehrfach genutzten Kanäle von Ad-hoc Netzwerken aufgelistet. Anschließend werden die daraus resultierenden Konsequenzen dargestellt.

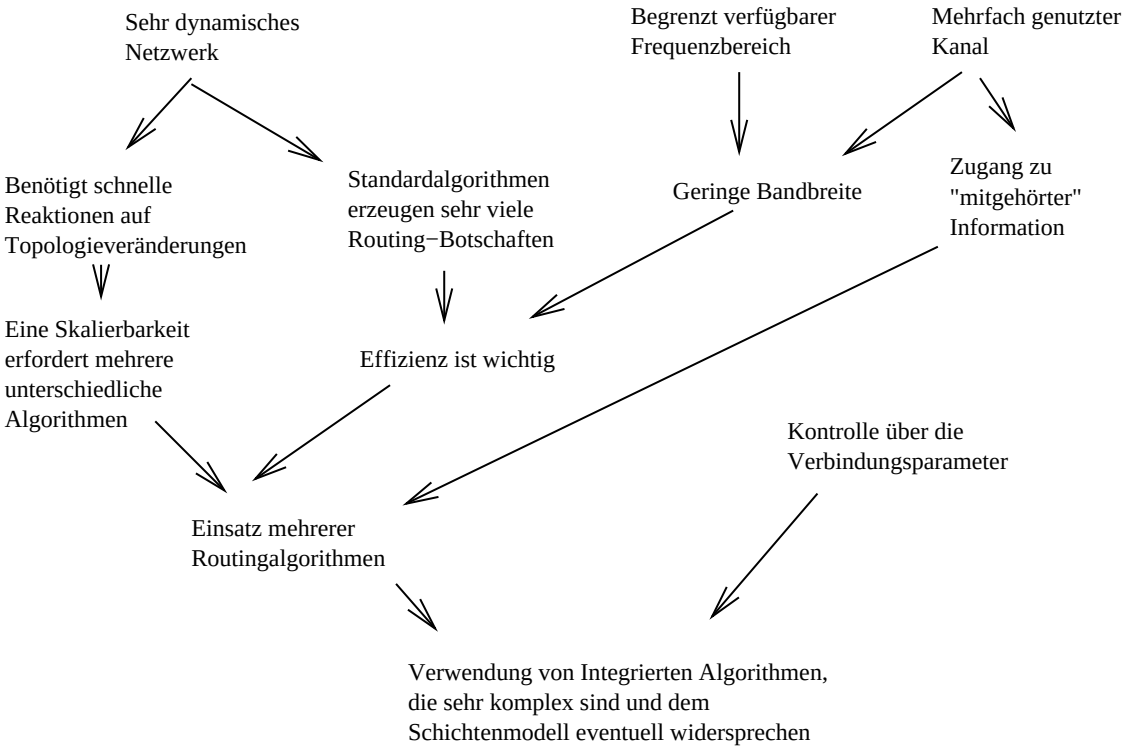


Abbildung 3.1: Warum ist Routing in Paketfunknetzen so komplex? [Ste95]

Normale Routingalgorithmen sind für ihren Eigenbedarf nicht sonderlich Ressourcen schonend und auch die Geschwindigkeit, mit der sie sich an neue Topologien anpassen, ist für dynamische Netzwerke unzureichend.

Die mehrfach genutzten Kanäle sind ein besonderes Beispiel dafür, welche Konsequenzen eine effektive Nutzung der Ressourcen fordert. In der Abbildung wird als Folge der mehrfach genutzten Kanäle ein Zugang zu mitgehörter Information angegeben. In diesem Punkt unterscheiden sich Standardalgorithmen und die Ad-hoc Algorithmen grundsätzlich voneinander. In einem Funksystem ist jede Sendung ein Broadcast, und viele Stationen empfangen Informationen, die eigentlich nicht für sie bestimmt waren. Ein Standardalgorithmus ist nicht dafür konstruiert, solche Informationen zu verarbeiten. Ein guter Ad-hoc Routingalgorithmus gestaltet seine Botschaften so, dass ein Mithörer auch daraus einen Nutzen ziehen kann.

Wird ein besonders effizienter Ad-hoc Routingalgorithmus gefordert, dann ist schon die Festlegung auf das im Abschnitt 2.3.1 beschriebene OSI Modell für Ad-hoc Netzwerke zu begrenzt. Eine Aufteilung der einzelnen Funktionen, wie sie das OSI Modell vorsieht, widerspricht oftmals einer effizienten Implementierung. Im vorliegenden Beispiel ist das Mithören von Paketen anderer Stationen ein unerwünschter Effekt, den die unteren OSI-Schichten durch ein Ausfil-

tern von Paketen zu verhindern haben. Das Kapitel zur Optimierung von Routingalgorithmen führt später noch weitere Beispiele an, wie durch schichtenübergreifende Funktionen die Leistungsfähigkeit des Routing zu steigern ist.

Die Beispiele zeigen, dass ein Standardalgorithmus alleine nicht die gewünschten Eigenschaften für Ad-hoc Netzwerke besitzt. Ein Ad-hoc Algorithmus ist schon sehr spezialisiert, aber er ist weiterhin auf die bekannten Algorithmen zur Berechnung und Optimierung der Routen angewiesen. Aus diesem Grund widmet sich der Rest des Kapitels zuerst den Standardalgorithmen und beschreibt anschließend die bekannten Ad-hoc Routingalgorithmen.

3.2 Basisalgorithmen

3.2.1 Distanz-Vektor

Der Distanz-Vektor (DV) Algorithmus wird auch als Bellman-Ford oder Ford-Fulkerson Algorithmus bezeichnet. Er wurde bereits im Vorläufer des Internet, im ARPANET, unter dem Namen Routing Information Protocol (RIP) verwendet und ist immer noch bei vielen bekannten Netzwerktypen z.B. DECnet, AppleTalk oder Novell IPX im Einsatz.

Der Algorithmus findet Routen, die bezüglich einer Metrik optimal sind. Dabei wird unter dem Optimum die minimale Summe der Leitungskosten verstanden. Leitungskosten können beliebige Kosten sein, die sich addieren lassen, beispielsweise Entfernungen oder Verzögerungen. Nach Anpassung des Algorithmus können auch Metriken verwendet werden, die multiplikativ wachsen, wie dies z.B. bei der Betrachtung von Fehlerfortpflanzung nötig ist. Es ist ebenso möglich, Routen mit maximaler Kapazität zu bestimmen, die jeweils aus dem Minimum der Leitungskapazitäten errechnet wird.

Der Algorithmus

Der iterative DV Algorithmus berechnet bei einem Durchlauf die Routen von jeder Station im Netzwerk zu einem bestimmten Zielknoten. Werden alle Routen benötigt, dann muss die Berechnung für jeden weiteren Zielknoten erneut durchgeführt werden.

Wird ein Netzwerk auf einen Graphen abgebildet, dann werden die Stationen als Knoten und die Leitungen als Kanten dargestellt. Alle Knoten werden durchnummeriert, der Zielknoten erhält die Nummer eins. D_i bezeichnet die Kosten für die günstigste Route vom Knoten eins zum Knoten j. Wenn i und j Nachbarn sind, dann verbindet sie eine Kante mit den Kosten d_{ij}. Für Knoten, die keine Nachbarn sind, ist $d_{ij} = \infty$.

Der DV Algorithmus iteriert in Schritten, die mit der Variablen h bezeichnet werden. Beim Start hat der Zielknoten die Distanz null. Dieser Wert wird auch später nicht mehr geändert. Alle anderen Distanzen werden vor dem Start auf unendlich gesetzt.

$$D_1^h = 0$$

$$D_i^0 = \infty,\ \forall i \neq 1$$

Das Ergebnis D_i^h wird durch folgende Iteration errechnet:

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h],\ \forall i \neq 1$$

Die Iteration ist abgeschlossen, wenn sich nichts mehr ändert.

$$D_i^h = D_i^{h-1}, \forall i$$

Abbildung 3.2 zeigt einen Routingvorgang für ein Netzwerk mit sechs Knoten und Kantengewichten von eins bis vier. Zur Lösung des Problems muss der Algorithmus vier Iterationen durchlaufen. Die vierte Iteration korrigiert jedoch nur noch den Distanzwert des sechsten Knotens und wurde deshalb nicht nochmals dargestellt.

DV ist ein iterativer Algorithmus und benötigt für die Lösung eines Routingproblems für einen Graphen mit N Knoten maximal h Schritte wobei $h \leq N-1$. Die Anzahl der Iterationen h ist proportional zum Diameter des Netzwerkes (gemessen in Hops). Dies lässt sich durch einen Induktionsbeweis zeigen, da mit der ersten Iteration Pfade mit einem Hop Länge gefunden werden und bei jeder weiteren Iteration die mögliche Pfadlänge um einen Hop wächst [Ber92]. Eine genauere Analyse des Algorithmus zeigt, dass der Berechnungsaufwand auch proportional mit der Anzahl der Kanten wächst. Hat ein Graph L Kanten und einen Diameter von m Hops, so ergibt sich schließlich der Aufwand für die Berechnung aller kürzesten Pfade zu einem Ziel als $O(mL)$. Da in einem Netzwerk mit N Knoten maximal N^2 Kanten existieren, ist der maximale Berechnungsaufwand nach oben durch $O(N^3)$ begrenzt. In der Praxis gilt jedoch $L \ll N^2$ und $m \ll N$ wodurch der Aufwand erheblich reduziert wird.

Der herausragende Vorteil des DV Algorithmus ist seine einfache Verteilbarkeit auf die Knoten des Netzwerkes. Dabei teilt sich sowohl der Rechenaufwand als auch der Speicheraufwand unter den Knoten auf. Die Ausführung des verteilten DV kann synchron oder asynchron durchgeführt werden, für beide Fälle ist eine Konvergenz in endlicher Zeit gewährleistet.

Im verteilten DV Algorithmus hat jeder Router eine Tabelle (den Vektor), welche die Wegkosten (Distanz) für jedes Ziel angibt und auch den Nachbarn enthält, über den das Ziel am günstigsten zu erreichen ist. Um die Funktionsweise an einem Beispiel zu erläutern, werden in einem Netzwerk (das in Abbildung 3.3 dargestellt ist) schrittweise Routingtabellen aufgebaut. Der gesamte Vorgang ist in Tabelle 3.1 dargestellt. Das Beispiel zeigt den Ablauf eines synchronen verteilten Routingvorganges. Hierbei werden die Routen zu allen Zielen berechnet. Jeder Knoten hat seine eigene Tabelle, die für jedes Ziel drei Angaben speichert: Die Kennung des Ziels, die Kennung des Nachbarn über den das Ziel am besten erreicht werden kann und die Distanz bis zum Ziel. Beim Start des Algorithmus enthält die Tabelle nur einen Eintrag, der die eigene Kennung und eine Distanz von null enthält. Bei jeder Iteration tauschen Nachbarn ihre Tabellen aus und übernehmen neue oder bessere Angaben in ihre Tabellen. Dadurch ermitteln die Knoten im ersten Schritt alle Nachbarn, im zweiten deren Nachbarn u.s.w..

Der DV Algorithmus konvergiert sehr schnell, wenn neue oder kürzere Pfade im Netzwerk verfügbar werden, er hat jedoch Konvergenzprobleme beim Wegfall oder Gewichtserhöhungen

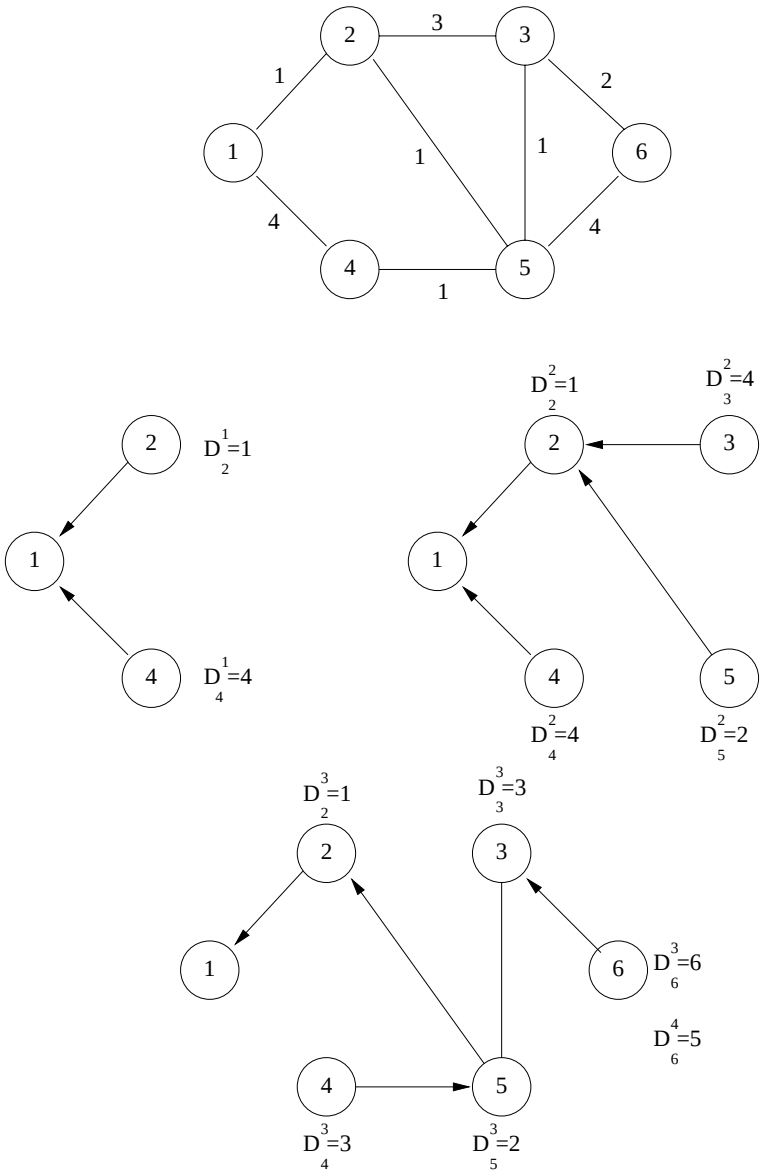


Abbildung 3.2: Routing mit dem Distanz-Vektor Algorithmus

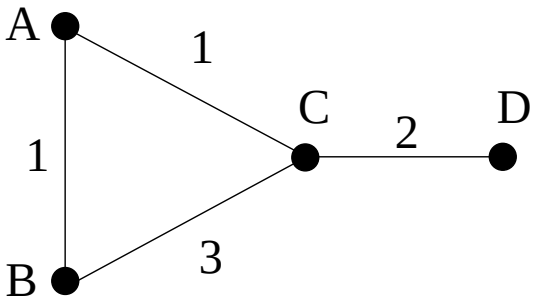


Abbildung 3.3: Graph des Routingbeispiels

Routingtabellen der Knoten vor dem Start

Knoten A			Knoten B			Knoten C			Knoten D		
Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über
A	0	-	B	0	-	C	0	-	D	0	-

Routingtabellen der Knoten nach dem ersten Schritt

Knoten A			Knoten B			Knoten C			Knoten D		
Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über
A	0	-	A	1	A	A	1	A	C	2	C
B	1	B	B	0	-	B	3	B	D	0	-
C	1	C	C	3	C	C	0	-			
						D	2	D			

Routingtabellen der Knoten nach dem zweiten Schritt

Knoten A			Knoten B			Knoten C			Knoten D		
Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über
A	0	-	A	1	A	A	1	A	A	3	C
B	1	B	B	0	-	B	2	A	B	5	C
C	1	C	C	2	A	C	0	-	C	2	C
D	3	C	D	5	C	D	2	D	D	0	-

Routingtabellen der Knoten nach dem dritten Schritt

Knoten A			Knoten B			Knoten C			Knoten D		
Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über	Ziel	Dist.	über
A	0	-	A	1	A	A	1	A	A	3	C
B	1	B	B	0	-	B	2	A	B	4	C
C	1	C	C	2	A	C	0	-	C	2	C
D	3	C	D	4	A	D	2	D	D	0	-

Tabelle 3.1: Routing mit dem Distanz-Vektor Algorithmus

von Kanten. Besonders der Ausfall von Leitungen kann sehr lange Konvergenzzeiten verursachen (count to infinity). Der Unterschied zwischen einem Routenaufbau und dem entsprechenden Abbau für ein lineares Netzwerk aus fünf Knoten (Abbildung 3.4) ist in den Tabellen 3.2 und 3.3 dargestellt.

Die Tabellen zeigen die Distanzen zum Knoten A, welche in den anderen Knoten ermittelt wird. Tabelle 3.2 beschreibt den Routenaufbau, wenn Knoten A im Netzwerk erstmals aktiv wird. Zu Beginn haben alle Knoten noch keine Entfernung zu A gespeichert, deshalb nehmen sie eine unendliche Entfernung von A an. Nach vier Iterationen ist der Endzustand erreicht, und alle Knoten kennen die korrekte Entfernung von A. Beim Routenabbau, der in Tabelle 3.3 dargestellt ist, wird der Endzustand nicht so schnell erreicht. Die Tabelle zeigt die Vorgänge im Netzwerk, nachdem die Verbindung zwischen Knoten A und B unterbrochen wurde. Knoten B geht in diesem Fall fälschlicherweise davon aus, eine Reserveroute über C benutzen zu können. Knoten C versucht dasselbe über D u.s.w.. In der Tabelle ist zu erkennen, wie sich die Distanzen langsam erhöhen, wobei sich andauernd Schleifen bilden. Die Distanzen erreichen letztlich einen Wert der von den Knoten als unendlich angesehen wird. Dies kann - je nach Größe des Netzwerkes und dem verwendeten Grenzwert für unendlich - sehr lange dauern.

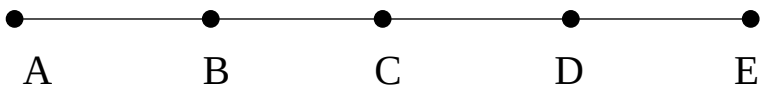


Abbildung 3.4: Graph des Routingbeispiels

A	B	C	D	E	
0	∞	∞	∞	∞	Startwert
0	1	∞	∞	∞	Nach 1. Iteration
0	1	2	∞	∞	Nach 2. Iteration
0	1	2	3	∞	Nach 3. Iteration
0	1	2	3	4	Nach 4. Iteration

Tabelle 3.2: Routenaufbau beim Distanz-Vektor Algorithmus

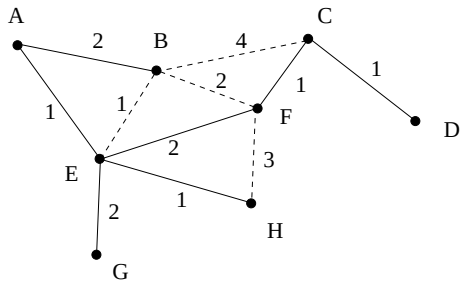
A	B	C	D	E	
0	1	2	3	4	Startwert
	3	2	3	4	Nach einer Iteration
	3	4	3	4	Nach zwei Iterationen
	5	4	5	4	Nach drei Iterationen
	5	6	5	6	Nach vier Iterationen
	7	6	7	6	Nach fünf Iterationen
	7	8	7	8	Nach sechs Iterationen
	⋮	⋮	⋮	⋮	⋮
	∞	∞	∞	∞	Endwert

Tabelle 3.3: Routenabbau beim Distanz-Vektor Algorithmus

Es existieren verschiedene Vorschläge zur Lösung dieses Problems. Das Problem und seine Konsequenzen werden in [Ste95] detailliert beschrieben, dort werden auch diverse Lösungen angeboten. Ein sehr effektiver Vorschlag, der nach vollständigen Pfaden sucht und deswegen Path Finding Algorithm (PFA) genannt wird, wird im Folgenden vorgestellt.

3.2.2 Pfadsuche

Bei der Pfadsuche wird die Routingtabelle der Knoten so erweitert, dass die benutzten Pfade zu jedem Ziel vollständig aus der Tabelle ablesbar sind. Damit ist die Schleifengefahr behoben, denn jede Route kann zurückverfolgt werden und Routen, die eine Schleife enthalten, werden nicht zugelassen.



Knoten A

Ziel	Dist.	über	Vorg.
A	0	-	-
B	2	A	A
C	4	E	F
D	5	E	C
E	1	A	A
F	3	E	E
G	3	E	E
H	2	E	E

Abbildung 3.5: Aufbau einer Routingtabelle mit Vorgängerinformation

Die Abbildung 3.5 zeigt ein Beispiel für eine solche Routingtabelle. Im vorliegenden Netzwerk nutzt der Knoten A nur die durchgehend eingezeichneten Kanten zur Weiterleitung seiner Pakete. Die Tabelle zeigt den Inhalt der Routingtabelle von Knoten A, die der Routingalgorithmus für das dargestellte Netzwerk errechnet hat. Neben den vom DV Algorithmus bekannten Einträgen enthält die Tabelle für jedes Ziel die Kennung des vorletzten Knotens im Pfad. Diese Knoten werden in der Tabelle als Vorgänger bezeichnet. Mit den Kennungen ist der minimal spannende Baum mit Wurzel A eindeutig beschrieben. Aus der Tabelle kann jeder Pfad mit der Vorgänger-Information aus der letzten Spalte rekonstruiert werden. So ist z.B. der Pfad von Knoten C zu Knoten A über die Knoten F und E realisiert.

Der Routingalgorithmus arbeitet nach dem gleichen Prinzip wie der verteilte DV Algorithmus. Die Routingtabellen werden unter den Nachbarn ausgetauscht und neue Ziele oder veränderte Distanzen werden in die eigene Routingtabelle übernommen. Allerdings muss vor jeder Änderung geprüft werden, ob dadurch eine Schleife entsteht. Solche Änderungen dürfen nicht in die Routingtabelle übernommen werden.

Die Erweiterung der Tabelle bringt auch Nachteile mit sich. Da nun mehr Informationen verfügbar sind, können übermittelte Tabellen widersprüchlich sein. Durch die teilweise Redundanz zwi-

schen Distanz- und Vorgängerinformation dürfen beispielsweise Knoten nie die gleiche Distanz wie ihr Vorgänger aufweisen. Da alle im Rahmen der Arbeit entwickelten Algorithmen die Pfadsuche nutzen, erfolgt eine ausführliche Diskussion der möglichen Probleme und entsprechende Lösungsmöglichkeiten in Kapitel 5 und 7.4.

3.2.3 Dijkstra-Algorithmus

Beim Algorithmus von Dijkstra werden, wie schon beim Distanz-Vektor Algorithmus, alle kürzesten Pfade von den Knoten des Netzes zu einem Zielknoten bestimmt. Der Dijkstra-Algorithmus hat einen deutlich verringerten Berechnungsaufwand, allerdings ist er nicht so flexibel. Beispielsweise ist er auf positive Metriken beschränkt.

Die Geschwindigkeit des Dijkstra-Algorithmus beruht auf der Verwendung einer sortierten Liste, aus der immer die kürzesten Pfade herausgesucht werden.

Der Algorithmus basiert auf folgender Überlegung: Die Kürzeste aller Routen zum Ziel muss in einem Hop zu einem Nachbarn des Ziels führen, denn kein Pfad kann kürzer sein, da er den ersten Hop enthalten muss und negative Metriken nicht zulässig sind. Die nächst kürzere Route muss entweder eine Route mit einem Hop zu einem anderen Nachbarn des Ziels sein oder die kürzeste Route mit zwei Hops über den zuerst gefundenen Nachbarn. Die Routensuche wird so lange fortgesetzt, bis Routen zu allen Nachbarn gefunden sind.

Algorithmus

Eine formale Darstellung des Algorithmus geht von einem Graphen aus, in dem die Knoten durchnummeriert sind. Der Zielknoten erhält dabei die Nummer 1. D_j bezeichnet die Kosten für die günstigste Route vom Knoten 1 zum Knoten j. Wenn i und j Nachbarn sind, dann verbindet sie eine Kante mit den Leitungskosten d_{ij}. Für Knoten, die keine Nachbarn sind, ist $d_{ij} = \infty$. Anfangs ist D_j nur eine Abschätzung, die sich ändern kann. Wenn der Wert jedoch feststeht, wird D_j als permanent bezeichnet. Alle Knoten mit permanenten Werten werden in einer Liste P erfasst. In jedem Schritt wird ein Knoten dieser Liste hinzugefügt. Zu Anfang ist $P = \{1\}$, $D_1 = 0$ und $D_j = d_{j1} \forall j \neq 1$. Danach werden zwei Schritte solange wiederholt, bis alle Knoten erfasst sind, oder die Liste mangels Knoten nicht mehr erweitert werden kann (z.B. in einem partitionierten Netz).

1. Finde den besten Knoten $i \notin P$ mit $D_i = \min_{j \notin P} D_j$ und erweitere $P = P \cup \{i\}$
2. Prüfe die nicht permanenten Werte: $\forall j \notin P$ setze $D_j = \min[D_j, d_{ij} + D_i]$

In der Abbildung 3.6 ist ein Routingvorgang mit dem Dijkstra-Algorithmus dargestellt. Dabei sind die Knoten, deren Distanz noch nicht permanent ist, mit gestrichelten Pfeilen gezeichnet. Aus Platzgründen ist nur die erste, zweite und vierte Iteration dargestellt. Das endgültige Ergebnis wird bei der fünften Iteration erreicht, die sich von der vierten nur durch den permanent gewordenen Knoten Nummer sechs unterscheidet.

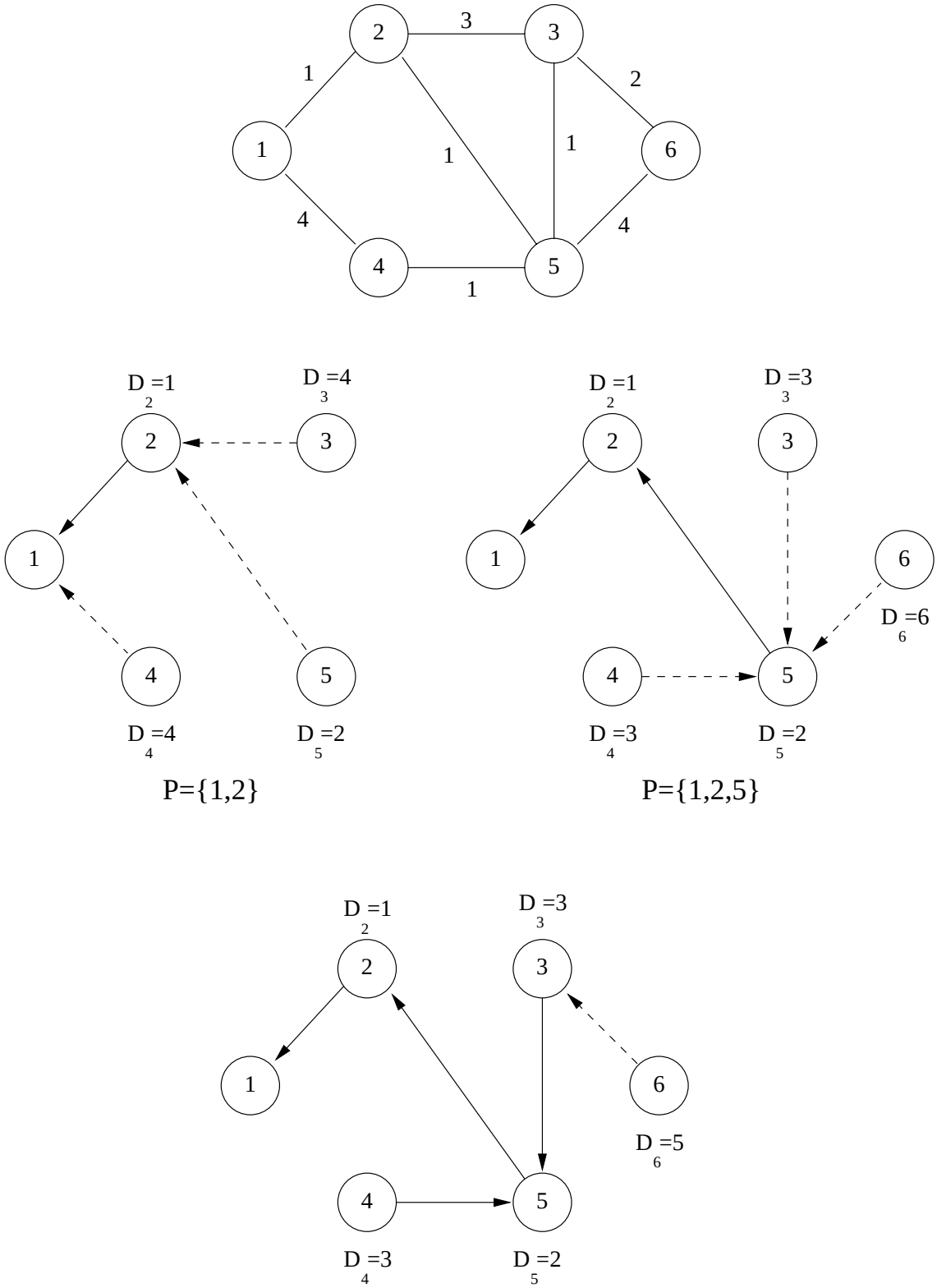


Abbildung 3.6: Routing mit dem Dijkstra-Algorithmus

Zur Abschätzung des Berechnungsaufwands wird die maximale Zahl der Iterationen und die Anzahl der Operationen pro Iteration betrachtet. Die Anzahl der Iterationen ist durch die Anzahl der Netzknoten N begrenzt, da der Zielknoten keinen Iterationsschritt braucht. Es sind demnach maximal $N-1$ Iterationen möglich. Eine Iteration sucht ein Minimum aus einer Liste, was maximal N Rechenschritte erfordert. Dadurch wird der maximale Berechnungsaufwand des Dijkstra-Algorithmus zu $O(N^2)$. Dies stellt gegenüber dem Aufwand $O(N^3)$ des DV Algorithmus eine beachtliche Verbesserung dar.

Es wurden bereits Studien veröffentlicht, die DV und den Dijkstra-Algorithmus miteinander vergleichen und dabei diverse Variationen zur Leistungssteigerung berücksichtigen [SAMDZ92]. Dabei gibt es keinen klaren Sieger; beide Algorithmen sind in der Praxis bei einer nicht verteilten Anwendung etwa gleichwertig. Für den Fall einer verteilten Anwendung ist DV klar überlegen, da die Berechnung des Dijkstra-Algorithmus nicht verteilt durchgeführt werden kann. Trotzdem wird im Internet meist der Dijkstra-Algorithmus eingesetzt, wobei jeder Knoten im Netzwerk eine Berechnung durchführt. Die zur Berechnung notwendigen Informationen müssen durch ein anderes Verfahren, das sogenannte Fluten, an alle Knoten verteilt werden. Dies führt zu einem höheren Kommunikationsaufwand und vervielfacht den Berechnungsaufwand in den einzelnen Knoten. Der Nachteil der höheren Kosten ist aber durch leistungsfähigere Technik auszugleichen und daher leichter zu kompensieren als die Schleifengefahr des DV Algorithmus.

3.2.4 Fluten

Die einfachste Art des Routing ist das Fluten der Daten. Beim Fluten werden die Pakete an alle Stationen im Netz verteilt. Fluten unterscheidet sich von einem ungerichteten Aussenden (Broadcast) durch die Anzahl der Zielstationen. Beim Fluten werden alle Stationen im Netzwerk erreicht, ein Broadcast erreicht nur alle Nachbarn eines Knotens.

Fluten ist eine hochparallele Methode, die sogar ohne Wissen über die Topologie des Netzwerkes angewandt werden kann. Zudem verwendet es immer den kürzesten Pfad, da es jeden Pfad benutzt.

Es ist manchmal unvermeidbar, Daten zu fluten, obwohl damit ein Netz stark belastet wird. Da z.B. der Dijkstra-Algorithmus nicht verteilt ausgeführt werden kann, müssen die notwendigen Daten erst durch Fluten an alle Stationen übermittelt werden, bevor sie verarbeitet werden können. In Situationen, in denen keine Informationen über das Netzwerk verfügbar sind, bietet Fluten oft die einzige Möglichkeit, Daten zwischen den Stationen auszutauschen.

Algorithmus:

In einen fest verdrahteten Netzwerk wird beim Fluten ein eingehendes Paket über alle Leitungen außer der Empfangsleitung weitergegeben. So entstehen eine große Zahl von Duplikaten, die sich schnell im Netzwerk ausbreiten. Wenn ein Netz Zyklen enthält, dann setzt sich ein Fluten ohne Begrenzungsmaßnahmen unendlich lange fort.

Das Fluten in drahtlosen Netzen unterscheidet sich erheblich vom Fluten in fest verdrahteten Netzwerken. Im Paketfunk ist jede Aussendung ein Broadcast, den alle Nachbarn empfangen. Deswegen muss eine Station eine Botschaft nur einmal aussenden, um die Botschaft an alle Nachbarn zu verteilen. (Kollisionen und Verluste durch überlastete Stationen werden hier nicht berücksichtigt) Diese positive Eigenschaft des Funkmediums hat auch einen Nachteil, denn beim drahtlosen Broadcast wird die Botschaft auch an die Nachbarn gesendet, die sie bereits bearbeitet haben. Deshalb ist die Begrenzung von Duplikaten hier absolut notwendig.

Der Mehrfachempfang von Paketen hat noch einen weiteren Nutzeffekt, wenn er als passives Acknowledge verwendet wird. Durch den Empfang der Wiederholungen ihrer Nachbarn kann eine Station in Erfahrung bringen, ob ihre Aussendung erfolgreich war.

Die Berechnungskomplexität des vorgestellten Flutalgorithmus ist konstant für jeden Knoten und deshalb $O(N)$ für das Netzwerk, wenn keine Maßnahmen zur Begrenzung von Duplikaten eingesetzt werden. Allerdings darf dabei der Graph des Netzwerkes keine Zyklen enthalten, denn sonst läuft der Algorithmus unendlich lange. Der entscheidende Berechnungsaufwand entsteht durch die eventuell notwendigen Begrenzungsmaßnahmen.

In [Ber92] werden zwei Maßnahmen zur Verhinderung von Duplikaten vorgestellt. Das erste Verfahren verwendet eine Kombination aus Senderkennung und Sequenznummern um Botschaften eindeutig zu kennzeichnen. Sind die Botschaften unterscheidbar, kann jede Station mit Hilfe einer Liste die bereits bearbeiteten Botschaften von unbearbeiteten unterscheiden und entsprechend reagieren. Dadurch sendet eine Station jedes Paket nur einmal aus.

Das zweite Verfahren verzichtet auf Sequenznummern und reduziert statt dessen das benutzte Netzwerk zu einen zyklenfreien Graph, der dann zum Fluten der Pakete genutzt wird. Leider funktioniert dies nur für Leitungsnetze, da sich Funkverbindungen nicht auf einzelne Stationen beschränken lassen. Die verteilte Berechnung eines zyklenfreien Graphen (Baum) aus einem beliebigen Graphen verursacht jedoch selbst schon einen Berechnungsaufwand, der in seiner Komplexität dem Routing vergleichbar ist.

Durch ein einfaches Fluten ist in der Praxis die Auslieferung von Paketen an alle Stationen nicht immer zu gewährleisten. Das Hauptproblem liegt dabei an den unzuverlässigen Broadcasts, bei denen die Aussendung an eine unbekannte Anzahl von Stationen erfolgt und eine Empfangsbestätigung deswegen auch nicht erwartet wird. Abbildung 3.7 demonstriert an einem Beispiel, wie durch Kollisionen einzelne Stationen vom Empfang einer gefluteten Botschaft ausgeschlossen werden.

Das Bild zeigt mit vier Schritten den Ablauf eines Flutens in einem Netzwerk aus 3×3 Stationen. Der Senderadius aller Stationen sei dabei so begrenzt, dass nur die horizontal und vertikal nebeneinanderliegenden Stationen sich erreichen können.

Im ersten Schritt wird ein Fluten in der linken oberen Ecke initiiert, das sich im zweiten Schritt weiter ausbreitet. Bereits in Schritt 2 kollidieren die Aussendungen zweier Stationen in der mittleren Station, die deswegen keinen korrekten Empfang hat. In der dritten Stufe breitet sich das Fluten weiter aus und im vierten Schritt wird wiederum durch synchrone Aussendung der Empfang der mittleren und der rechten unteren Station gestört.

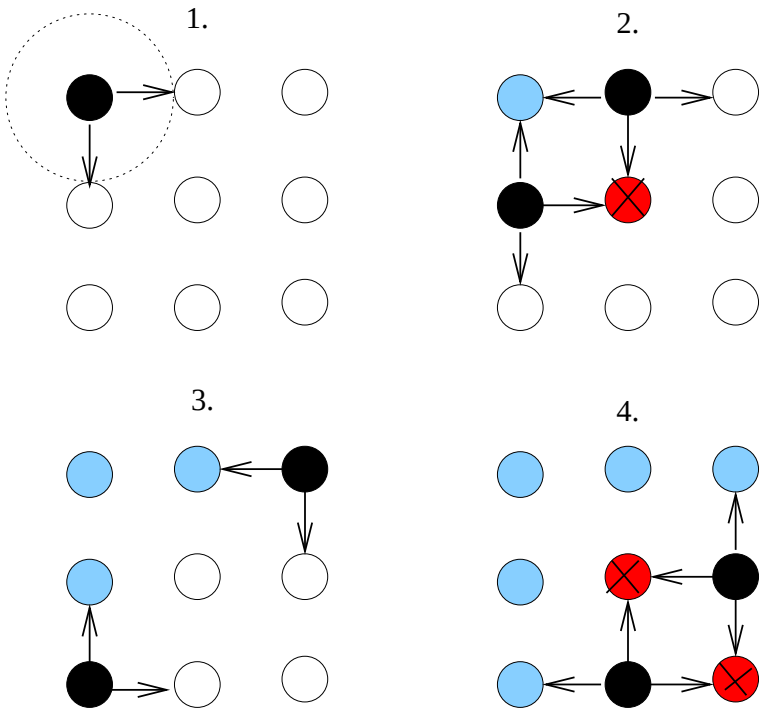


Abbildung 3.7: Kollisionen beim Fluten eines Paketes

Mit Verzögerungszeiten, die zufällig zwischen dem Empfang und der Wiederaussendung von Paketen eingefügt werden, lässt sich das Problem entschärfen [BMJ+98b], aber Fluten ist generell unzuverlässig, solange unzuverlässige Broadcasts als Basisdienst benutzt werden.

3.2.5 Link-State Routing

Das Link-State Routing (LSR) ist eine Kombination aus Fluten und einem lokal arbeitenden Routingalgorithmus. Dazu kann jeder Algorithmus verwendet werden, der kürzeste Pfade berechnen kann.

Algorithmus

Beim Start des Algorithmus ermittelt jeder Knoten seine Nachbarn und die Distanz zu ihnen. Diese Information wird an alle anderen Knoten durch Fluten übermittelt.

Jeder Knoten sammelt die gefluteten Informationen, errechnet daraus seine Sicht des Netzwerks und speichert sie als Graph. Auf diesen Graph wird der lokale Algorithmus, z.B. der Dijkstra-Algorithmus angewandt, der die kürzesten Wege zu allen anderen Knoten des Graphen bestimmt.

Ändern sich bei einem Knoten die Distanzen zu den Nachbarn, dann muss diese Information erneut an alle verteilt werden. Jeder Empfänger dieser Information hat dann seine Sicht des

Netzwerkes auf den neuen Stand zu bringen und die Pfade erneut zu berechnen.

Durch das Fluten wird jeder lokal arbeitende Routingalgorithmus auch in einem verteilten System einsetzbar. Dabei entstehen allerdings Konsistenzprobleme, weil Fluten immer eine Verzögerung der Botschaften verursacht und deshalb die Stationen teilweise unterschiedliche Sichten der Netztopologie haben.

Das Problem wird noch verschärft, wenn das Netzwerk zeitweilig in Teilnetze zerfällt (Partitionierung). Dann werden Informationen über Veränderungen nicht mehr im ganzen Netzwerk verteilt. Verbinden sich die Teilnetze wieder, dann haben die Stationen der Teilnetze dauerhaft eine unterschiedliche Sicht der Netztopologie.

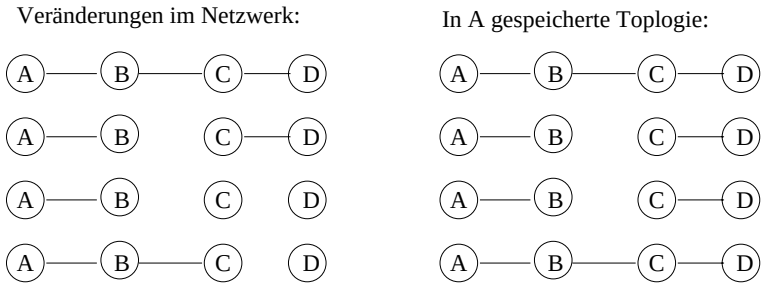


Abbildung 3.8: Netzwerkunterbrechungen beim Fluten

Abbildung 3.8 zeigt ein Beispiel für Informationsverlust durch Partitionierung. Auf der linken Seite ist dabei in vier Schritten die jeweils existierende Netzwerktopologie angegeben. Auf der rechten Seite ist die Topologie angegeben, die Station A zu diesem Zeitpunkt gespeichert hat. Jede Station flutet sofort eine Botschaft, wenn sich eine Leitung der Station verändert. In der Abbildung ist zu erkennen, dass im vierten Schritt das Netzwerk und die in A gespeicherte Topologie für die Leitung C-D nicht identisch ist. A konnte die Botschaft, dass diese Leitung nicht mehr existiert, im dritten Schritt nicht empfangen, da zu diesem Zeitpunkt die Leitung B-C unterbrochen war.

Dieses Problem wird in der Praxis dadurch gelöst, dass jede Station ihre Nachbarschaftsbeziehungen in regelmäßigen Abständen fluten muss. Das bewirkt eine hohe Netzbelastung durch die Botschaften, garantiert aber, dass nach einer gewissen Zeit alle Stationen die gleichen Informationen haben. Die große Anzahl von gefluteten Paketen ist auch der Hauptgrund, weshalb Link-State Protokolle für Ad-hoc Netzwerke nicht geeignet sind.

3.3 Ad-hoc Routingalgorithmen

Im Folgenden werden aus der Literatur bekannte Algorithmen vorgestellt, die speziell für den Einsatz in selbstkonfigurierenden Netzwerken entwickelt wurden. Zur besseren Orientierung werden die Algorithmen nach ihrem Funktionsprinzip in Gruppen unterteilt. Abbildung 3.9 gibt einen Überblick zu dieser Unterteilung.

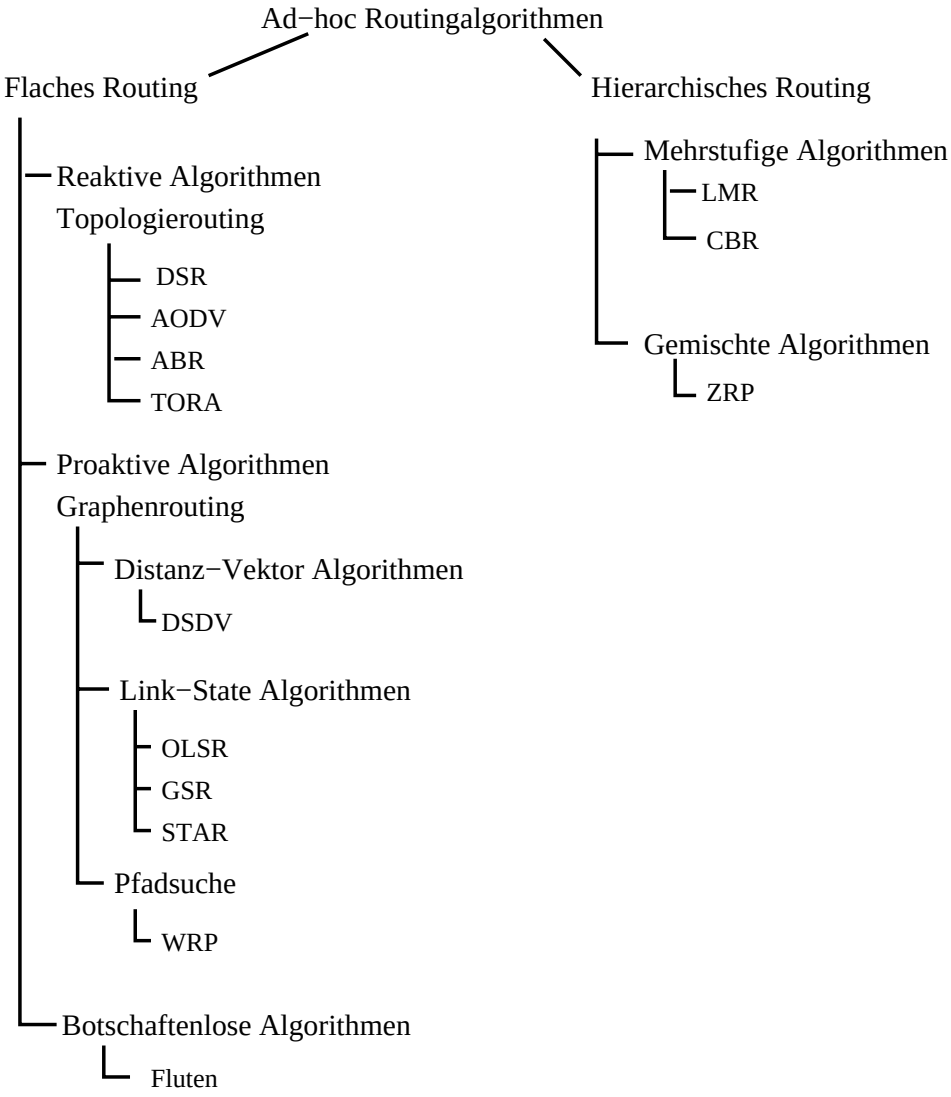


Abbildung 3.9: Einordnung von Ad-hoc Algorithmen

Die wichtigste Unterscheidung wird dabei zwischen Algorithmen für flaches Routing und den hierarchischen Algorithmen getroffen. Das Kriterium für hierarchisches Routing ist die inhomogene Funktionsaufteilung unter den Stationen, dabei übernehmen einzelne Stationen spezialisierte Funktionen innerhalb einer Gruppe. Eine Aufteilung der Funktionen ist besonders vorteilhaft, wenn große Netzwerke verwaltet werden müssen, da der Verwaltungsaufwand in großen Netzwerken überproportional anwächst. Dies wird schon durch die Komplexität der bereits vorgestellten Routingalgorithmen deutlich, bei denen die Rechenkomplexität etwa mit $O(N^2)$ gegenüber der Knotenzahl N anwächst. Hierarchien teilen die Netzwerke in kleinere, überschaubare Einheiten auf und vereinfachen so deren Verwaltung. Hierarchische Routingalgorithmen müssen sich aber um die Verwaltung der Hierarchie kümmern, die in mobilen Systemen keinen dauerhaften Bestand hat. Die Pflege der hierarchischen Beziehungen kann einen hohen Aufwand verursachen, der eventuell die Vorteile der Funktionsaufteilung wieder aufzehrt. Ein interessantes Forschungsgebiet der Zukunft ist daher die Untersuchung der Parameter, bei denen hierarchische Systeme günstiger als flaches Routing sind.

Im flachen Routing sind die Funktionen homogen verteilt, die einzelnen Stationen unterscheiden sich nicht in ihrem Funktionsumfang. Da die Stationen nun beliebig austauschbar sind, wird das Routing dadurch wesentlich vereinfacht. Für flaches Routing existieren bereits unzählige Algorithmen. Aus diesem Grund wird innerhalb der Algorithmenklasse noch einmal zwischen reaktiven und proaktiven Algorithmen unterschieden. Für Sonderfälle wurde noch eine eigene Klasse geschaffen, die botschaftenloses Routing genannt wird. Hier ist das Fluten angesiedelt, und es lassen sich auch verwandte Verfahren wie z.B. geographisches Routing (vgl. DIRC [Sie97, Bor02]) in diese Klasse einordnen.

Die reaktiven Algorithmen suchen eine Route erst dann, wenn ein Bedarf dafür erkannt wird. Diese Vorgehensweise ist günstig, wenn wenig Kommunikation erforderlich ist, da das Netzwerk nur mit wenigen Suchanfragen belastet wird. Die Nachteile von reaktiven Verfahren zeigen sich, wenn die Anzahl der Stationen ansteigt und damit auch der Kommunikationsbedarf wächst [SR96]. Da reaktive Verfahren ihre Routen im gesamten Netz suchen müssen, werden die einzelnen Stationen in größer werdenden Netzwerken immer stärker mit Suchanfragen belastet. Aus diesem Grund können Netzwerke mit reaktiver Routensuche eine bestimmte Größe nicht überschreiten, ansonsten wird die gesamte Netzwerkkapazität für die Routensuche aufgebraucht.

Die proaktiven Routingalgorithmen versuchen kontinuierlich, ihre Routen an das aktuelle Netzwerk anzupassen, dadurch stehen immer aktuelle Routen zur Verfügung. Die Pflege der Routen verursacht permanente laufende Kosten hinsichtlich der Bandbreite und der Batteriekapazität, dafür sind aber die notwendigen Informationen über das verfügbare Netzwerk immer vorhanden. Diese Informationen werden von höheren Diensten benötigt, um situationsabhängige Entscheidungen hinsichtlich der verwendeten Algorithmen, Übertragungsverfahren und mobiler Applikationen zu treffen. Die aus den Basisalgorithmen bekannten Verfahren Distanz-Vektor, Pfadsuche und Link-State werden genutzt, um die Algorithmen im proaktiven Routing weiter einzuordnen.

3.3.1 Dynamic Source Routing

Einer der ersten Routingalgorithmen, der speziell für Ad-hoc Netzwerke entwickelt wurde, ist der Dynamic Source Routing Algorithmus (DSR) [Joh94, JMHJ01]. DSR ist ein klassisches Beispiel für einen reaktiven Algorithmus, da hier gezielt nach benötigten Pfaden gesucht wird.

Das besondere Merkmal von DSR ist sein extrem einfaches Funktionsprinzip. Das DSR Verfahren beruht ausschließlich auf protokolliertem Fluten, bei dem eine Suchbotschaft lawinenartig im Netzwerk versendet wird. Dabei wiederholt jede Station eine Suchbotschaft genau einmal und trägt damit zur vollständigen Verbreitung der Suchbotschaft im Netzwerk bei.

Das Fluten ist eine sehr aufwändige Vorgehensweise und begrenzt auch die Anwendbarkeit dieses Verfahrens auf kleine Netzwerke. Um nicht übermäßig viele Botschaften fluten zu müssen, wird die Kennung jeder Station, die eine Suchbotschaft weitergegeben hat, in der jeweiligen Suchbotschaft vermerkt. Damit entsteht eine Liste aller an der Kommunikation beteiligten Stationen. Wenn die Suchbotschaft das gewünschte Ziel erreicht, dann enthält diese Liste alle Vermittler entlang des Pfades vom Sender zum Ziel. Dieser Pfad kann nun rückwärts genutzt werden, um eine Bestätigung zurückzusenden, damit auch der Sender eine Kopie des Protokolls und damit der gewünschten Pfadinformation bekommt.

Ist der Pfad ermittelt, kann mit der Versendung der Nutzdaten begonnen werden. Jeder Sender hinterlegt in den Kopf der ausgesendeten Nutzdaten die vollständige Liste aller Stationen, die die Daten weiterreichen sollen. Wenn alle genannten Stationen sich an diese Verhaltensanweisung halten, gelangt das Paket zum richtigen Empfänger. Tritt während der Übertragung ein Fehler auf, so wird dies mittels einer Botschaft dem Sender mitgeteilt. Fällt nun eine Verbindung während einer Übertragung aus oder tritt ein Übertragungsfehler auf, muss der Sender erneut versuchen, eine Verbindung über eine neu geflutete Suchbotschaft zu ermitteln.

Es existieren mehrere Optimierungsvorschläge für den DSR Algorithmus. Das einfachste Verfahren setzt auf den massiven Einsatz von sogenannten Route-Caches; hierbei analysieren alle Stationen, die in den vorbeilaufenden Paketen gespeicherten Pfade und bauen diese in die eigenen Tabellen ein. Damit lässt sich ein erheblicher Teil der Suchbotschaften einsparen, da im Route-Cache oft ein passender Pfad zu einem Ziel bereits vorhanden ist. Weitere Verbesserungen versuchen z.B. durch Befragen der Nachbarn die fehlende Pfadinformation zu beschaffen. Alle Verbesserungsvorschläge kämpfen aber mit dem Problem der Routenalterung, die in mobilen Netzwerken häufig auftritt. Da DSR eine gespeicherte Route nie nachbessern kann, veralten die Routen schnell und führen dann zu Problemen durch ungültige Einträge im Routen-Cache.

Der DSR Algorithmus ist durch das Fluten in seiner Skalierbarkeit deutlich eingeschränkt. Er ist aber in kleinen Netzwerken trotz der hier aufgezeigten Nachteile eine sehr effiziente Lösung, insbesondere wenn lange Phasen ohne Kommunikation im Netzwerk auftreten [BMJ+98a]. Dann wird die fehlende Pflege der Routen zu einem Vorteil, denn es werden keine Aussendungen mehr durchgeführt, die Stationen können in einen Schlafmodus gehen und so Batteriekapazität und Prozessorleistung einsparen.

3.3.2 Ad-hoc On-Demand Distance Vector

Ein Verwandter von DSR ist der Ad-hoc On-Demand Distance-Vector Algorithmus (AODV), der 1997 erstmals von Perkins vorgestellt wurde [Per97, Per02]. Beim AODV werden die Algorithmen von DSR und der in Abschnitt 3.3.5 beschriebene proaktive Destination Sequence Distance Vector (DSDV) Algorithmus miteinander kombiniert.

Wird eine Route benötigt, dann muss bei AODV ein Request durch das Netzwerk geflutet werden. Im Unterschied zu DSR werden die Kennungen der Knoten, die den Request weitervermittelt haben, nicht im Request selbst gespeichert, sondern jeder Knoten speichert seinen Vorgänger für eine gewisse Zeit in seiner eigenen Routingtabelle. Das ermöglicht ein Zurückreichen der Antwort wie beim DSR. Allerdings muss nun die Zeit für die Speicherung der Route klar definiert werden. So hat jede Route eine begrenzte Lebensdauer, die bei Bedarf durch eine Botschaft verlängert wird.

Erreicht der Request die Zielstation, dann antwortet diese mit einem Reply. Der Reply läuft entlang des gespeicherten Rückweges und fixiert so den gefundenen Weg. Dazu ist eine Verlängerung der Lebensdauer der Route nötig und jeder Vermittlungsknoten speichert in seiner Routingtabelle zusätzlich noch die Distanz zum Ziel und die bei DSDV beschriebene Sequenznummer.

3.3.3 Associativity Based Routing

Beim Associativity Based Routing (ABR) [Toh97a, Toh97b] wird versucht, die Routen über die stabilsten, d.h. die am wenigsten bewegten Knoten zu legen. Dazu messen die Knoten die Lebensdauer der Verbindungen zueinander und speichern so zu jedem Nachbarn den Stabilitätswert. Bei diesem Routingverfahren wird implizit davon ausgegangen, dass alte Verbindungen länger stabil bleiben und somit zuverlässiger sind.

In dem Protokoll werden die Routen, wie bei DSR, durch eine geflutete Suchanfrage ermittelt. Dabei erweitern die beteiligten Knoten das Suchpaket aber nicht nur um ihre Kennung, sondern fügen zusätzlich noch den Stabilitätswert und die gemessene Belastung der Verbindung (mittlere Verzögerung) an. Die zusätzlichen Werte sind nicht für jeden Hop gespeichert, sie werden kumuliert. Der gesuchte Knoten hat beim Empfang der Suchanfrage dann einen Stabilitätswert, einen Belastungswert und die Routenlänge zur Verfügung. Gehen beim Zielknoten mehrere Suchanfragen ein, dann kann der Zielknoten nach den Kriterien seiner Wahl die günstigste Route heraussuchen. Bei den gefluteten Suchanfragen ist aber nicht feststellbar, ob mehrere Suchanfragen eingehen werden und wie lange darauf gewartet werden muss.

Da sich ABR auch wie DSR völlig auf das Fluten von Suchanfragen verlässt, ist seine Skalierbarkeit mit der von DSR vergleichbar. In ABR wird aber, im Gegensatz zu DSR, die Route nicht in jedem Paket abgelegt. Nur die Suchanfrage und die dazugehörige Antwort enthalten die volle Route. Alle beteiligten Knoten speichern beim Zurückübermitteln der Antwort den für sie relevanten Teil in ihrer Routingtabelle. Dies hat den Vorteil, dass ein Routendefekt lokal

repariert werden kann. Bricht eine Route, so wird ein lokal begrenztes Fluten verwendet, um den Anschluss wieder herzustellen. Gelingt dies nicht, dann werden die Teilrouten gelöscht, und eine neue, netzweite Suchanfrage wird gestartet.

Obwohl ein lokaler Reparaturmechanismus sehr erstrebenswert ist, bringt er in der Implementierung von ABR einen deutlichen Nachteil mit sich. Eine aktive Route wird andauernd gewartet, deshalb muss sie nach Abschluss der Datenübertragung auch gelöscht werden, um unnötige Reparaturmaßnahmen zu verhindern. Dies blockiert aber eine kurzfristige Wiederverwendung der Route, denn für einen neuen Routenaufbau muss immer eine Suchanfrage gestartet werden, die durch ihr Fluten eine hohe Netzbelastung und Wartezeiten verursacht. Das ist notwendig, da die Routenplanung Metriken berücksichtigt, die sich mit der Zeit verändern und deswegen jedes Mal neu bestimmt werden müssen.

Der ABR Algorithmus ist durch ein US-Patent geschützt; dies verhindert den Einsatz in allen offenen Netzwerksystemen. Seine Vorteile, die Nutzung besonders stabiler Pfade, wird mit einem erheblichen Nachteil erkauft: um die Stabilität eines Pfades zu ermitteln, sind andauernde Kontrollen der Verbindungen notwendig. Damit ist der Hauptvorteil eines reaktiven Algorithmus, die Energieersparnis, in ABR nicht vorhanden.

3.3.4 Time Ordered Routing Algorithm

Der Time Ordered Routing Algorithmus (TORA) [PC97, PC01] verwendet, wie das Dynamic Source Routing, eine Suche per Broadcast, um den Zielknoten zu finden. Im Gegensatz zum DSR werden die Routen nicht als Liste in den Paketen gespeichert, sondern jeder Knoten speichert den nächsten Schritt in seiner Routingtabelle. Für die Suche und eventuell notwendige Korrekturen stehen drei Botschaftentypen zur Verfügung, die den Botschaften von DSR sehr ähnlich sind.

TORA ist für Netzwerke mit hoher Mobilität entwickelt worden und kann deswegen auch schnell auf Veränderungen reagieren. Falls Korrekturen der Routen notwendig sind, werden diese, falls möglich, lokal vorgenommen. Die Routen werden erst bei Bedarf gesucht, und es stehen sogar mehrere Routen gleichzeitig zur Verfügung. Die Länge der Routen ist dabei allerdings zweitrangig, denn es werden Umwege in Kauf genommen, wenn dadurch keine neuen Routen gesucht werden müssen.

TORA benutzt eine aufwändige Metrik für die Steuerung des Paketflusses. Der Vorgang lässt sich mit einem System vergleichen, in dem Wasser von einem Hügel hinab durch ein Netzwerk von Rohren zu einer Zielstation fließt. In diesem Modell entsprechen die Verzweigungen der Rohre den Routern und das Wasser steht für den Paketfluss. Jeder Router liegt auf einer bestimmten Höhe, die je nach Entfernung zum Ziel immer weiter ansteigt. Wenn eine Leitung von einer Station A zu einer tiefer gelegenen Station B blockiert ist, so dass kein Wasser mehr hindurchfließen kann, dann vergrößert A seine Höhe so lange, bis A höher als alle seine Nachbarn ist und folglich das Wasser nun wieder abfließt. Dadurch fließt das Wasser eventuell zu den Stationen zurück, von denen es kommt. Diese müssen dann ebenfalls ihre Höhe korrigieren. So

entsteht eine Kettenreaktion, die entweder wieder zu einer funktionsfähigen Höhenanordnung der Stationen führt oder im Fehlerfall eine Auflösung des Systems bewirkt.

In jeder Station wird eine eigene Routingtabelle für jedes Ziel angelegt; es ist nicht möglich, die Informationen aus anderen Routingvorgängen zu verwenden. Wenn ein Knoten eine Route benötigt, dann broadcastet er eine Query, die die Kennung der Zielstation enthält. Die Query wird durch das Netzwerk geflutet, dabei speichert jeder Router seinen Vorgänger in seiner Routingtabelle, damit er eine Antwort zurückreichen kann. Erreicht die Query das Ziel, so antwortet diese mit einem Update. Das Ziel legt seine eigene Höhe fest und fügt die Information in die Update-Botschaft ein. Alle Knoten, die den Update an den Ursprung zurückleiten, setzen sich selbst höher als die Vorgänger, von denen sie den Update erhalten haben. So entsteht ein gerichteter Graph mit einer Kette von Verbindungen, die vom Sender der Query zum Ziel führen.

Entdeckt ein Knoten, dass er keine Verbindung mehr zum Ziel hat, dann vergrößert er seine Höhe, so dass diese nun grösser ist als die seiner Nachbarn und sendet ein neues Update Paket. Wenn der Knoten keine Nachbarn mehr hat, über die er sich stellen kann, dann versucht er eine neue Route durch eine erneute Query zu finden.

TORA erzeugt durch die für jedes Ziel voneinander unabhängigen Vorgänge eine große Anzahl von Botschaften und hat dadurch Probleme in größeren Netzwerken. So kann der Bruch einer Verbindung mehrere voneinander unabhängige Kettenreaktionen zur Höhenkorrektur auslösen. Damit TORA korrekt arbeitet, müssen die Botschaften in der richtigen Reihenfolge übertragen werden. Dies erfordert ein eigenes Sub-Protokoll, welches wiederum Botschaften erzeugt. Dieses Subprotokoll kann immerhin Botschaften ansammeln und in einem gemeinsamen Paket verschicken und damit das Netzwerk etwas entlasten. Solange das Problem der großen Botschaftenanzahl und der gelegentlich auftretenden Kettenreaktionen nicht gelöst ist, wird TORA allerdings kaum sinnvolle Einsatzfelder finden. Der Vorteil von mehreren parallel nutzbaren Pfaden, und der damit verbundenen Lastverteilung, wird derzeit durch die anderen Probleme nicht kompensiert.

3.3.5 Destination Sequence Distance Vector

Das Destination Sequence Distance Vector (DSDV) Verfahren wurde erstmals 1994 von Perkins und Bhagwat vorgeschlagen [PB94]. Der Distanz Vektor (DV) Algorithmus, der schon bei den Basisalgorithmen in 3.2.1 vorgestellt wurde, bildet den Kern des Routingverfahrens. Da der DV Algorithmus die bereits beschriebenen Schwachpunkte hat, erweitert DSDV die Routingtabelle in den Knoten um Einträge für Sequenznummern. Diese bieten eine Lösung für die Probleme bei ansteigenden Kantengewichten und den dadurch entstehenden Schleifen.

In DSDV hat jeder Router eine Routingtabelle mit Einträgen für die Kennung des Zielknotens, die Distanz und den Nachbarn, über den das Ziel am besten erreicht wird. Die vorgeschlagene Verbesserung erweitert die Tabelle um einen Eintrag für eine Sequenznummer. Diese Sequenznummer gibt die „Neuheit“ der Route an. Eine neuere Route ist einer kürzeren Route immer

vorzuziehen. Nur wenn beide Routen gleich alt sind, entscheidet die Distanz. Die Nummer für den jeweiligen Eintrag wird vom betroffenen Zielknoten festgelegt. Er wählt dazu einen geraden Wert, und erhöht ihn immer nur in Zweierschritten. Sollte irgendein Knoten des Netzwerkes feststellen, dass die Verbindung zum Zielknoten abgebrochen ist, dann setzt er die Distanz auf unendlich und erhöht die Sequenznummer um eins. Da außer dem Zielknoten alle anderen Knoten nun die neuere Route akzeptieren müssen, verbreitet sich die unendliche Distanz ohne Verzögerung im Netz. Die Verbreitung stoppt, wenn der Zielknoten das nächste Mal die Sequenznummer um zwei erhöht und den neuen Eintrag verschickt. Dadurch startet der Routingvorgang erneut, und es bauen sich aktuelle Routen zum Zielknoten auf.

Der DSDV Algorithmus ist eine deutliche Verbesserung gegenüber dem reinen DV Algorithmus, da er Schleifenfreiheit garantiert. Die Kosten für die Schleifenfreiheit sind allerdings erheblich. Ein Verbindungsbruch kann nicht mehr lokal bearbeitet werden. Statt dessen werden alle Routen zu einem Ziel im gesamten Netz gelöscht und anschließend neu erzeugt. Die Garantie für Schleifenfreiheit gilt nur, wenn die Leitungskosten in Hops gemessen werden, denn dann sind die Kosten für eine Leitung entweder unendlich oder eins. Bei einer Metrik mit mehr Zuständen tritt eine Schleifenbildung immer noch auf, wenn Distanzen erhöht werden.

DSDV ist ein wichtiger Meilenstein in der Entwicklung von Routingalgorithmen für drahtlose Netzwerke, denn er ist der erste Vertreter einer vollständig verteilten Routenberechnung. Allerdings wird er nicht mehr eingesetzt, da seine Nachfolger effizienter sind.

3.3.6 Optimized Link-State Routing

Das Optimized Link-State Routing (OLSR) [CQJM01] verwendet ein auf Ad-hoc Netzwerke angepasstes Link-State Protokoll. Das Hauptproblem von LSR ist die große Zahl gefluteter Botschaften. In OLSR wird deshalb das Fluten optimiert, dazu werden bestimmte Knoten, ähnlich wie bei einem Backbone, als Verteiler genutzt um die Botschaften möglichst schnell im Netz versenden zu können und unnötige Wiederholungen zu vermeiden. Da OLSR am eigentlichen Problem aber nichts ändern kann, ist das Verfahren, besonders bei Netzwerken mit vielen Topologieveränderungen, wenig effizient.

3.3.7 Global-State Routing

Das Global-State Routing (GSR) [CG98] basiert auf dem Link-State Verfahren, aber es verzichtet auf Fluten zur Verteilung der Topologieinformation. Statt dessen werden Vektoren mit den Link-States in regelmäßigen Intervallen zwischen benachbarten Stationen ausgetauscht. Der Verteilungsmechanismus führt zu einer gleichmäßigen Anzahl von Botschaften, und ist unabhängig von den Veränderungen im Netzwerk. Die Wahl der Intervallzeit ist aber für den Algorithmus von entscheidender Bedeutung. Ein kleines Intervall verursacht viele Botschaften, ein zu groß gewähltes Intervall lässt viele Updates zu spät ankommen, so dass der Routingalgorithmus die meisten Routen nicht rechtzeitig neu berechnen kann. Nach Spezifikation erfolgt

zwar die Routenberechnung durch den Algorithmus von Dijkstra, es ist aber jeder Algorithmus zur Berechnung kürzester Pfade einsetzbar, da die gesamte Topologie in jeder Station gespeichert ist.

Durch seine speziellen Eigenschaften ist der Algorithmus nur für Netzwerke geeignet, die eine konstante Datenrate für das Routing benötigen. Die feste Datenrate legt aber gleichzeitig fest, wie schnell der Algorithmus auf Topologieveränderungen reagieren kann.

3.3.8 Source Tree Adaptive Routing

Ein mit GSR verwandter Algorithmus ist das Source Tree Adaptive Routing (STAR) [GLAS01] in dem die Link-State Daten auch nur zwischen Nachbarn ausgetauscht werden, aber im Gegensatz zu GSR werden die Updates durch Topologieänderungen ausgelöst. Da Link-State Updates an alle Stationen im Netzwerk gesendet werden müssen, ist eigentlich wieder ein Fluten nötig, um die Information zu verteilen. STAR begrenzt die Weitergabe aber auf Updates, die zur Funktionsfähigkeit des Netzwerkes unbedingt erforderlich sind. Daher werden letztlich nur Informationen über Links weitergeleitet, die zu den aktuellen kürzesten Pfaden gehören oder diese erweitern oder kürzen. Die Spezifikation von STAR geht sogar noch einen Schritt weiter und erlaubt es, Updates zu verwerfen, die nur eine kleine Verbesserung der aktuellen Pfade bewirken.

Der STAR-Algorithmus benutzt für die Weitergabe von Updates eine relativ aufwändige Berechnung, da er jedes Mal überprüfen muss, ob eine Botschaft wichtige Links betrifft. Da beim Link-State Verfahren jede Station eine eigene Routenberechnung durchführt, müssen auch Botschaften weitergeleitet werden, die verhindern, dass die Routenberechnung eines Nachbarn Schleifen erzeugen.

Die Reduktion der weiterzugebenden Botschaften bewirkt eine hohe Effizienz des STAR Algorithmus. Er nutzt die Funkbandbreite wesentlich effektiver aus als alle vergleichbaren Link-State Algorithmen. Die aufwändigeren Berechnungen sind dennoch vergleichsweise einfach durchzuführen, da auch Kleinststationen mittlerweile sehr leistungsfähige Prozessoren besitzen und die Rechenleistung oder der Speicher einer Station viel einfacher zu erweitern ist als die Funkbandbreite.

3.3.9 Wireless Routing Protocol

Eine interessante Alternative zu DSDV ist das Wireless Routing Protocol (WRP) [MGLA96]. Auch dieser proaktive Routingalgorithmus basiert auf dem DV Algorithmus. Er benutzt jedoch die Pfadsuche, um das Schleifenproblem zu beheben. Dazu wird die Routingtabelle der Knoten so erweitert, dass die benutzten Routen, also ein minimal spannender Baum, vollständig aus der Tabelle abgelesen werden können. Damit ist die Schleifengefahr behoben, denn jede Route kann zurückverfolgt werden. Routen, die eine Schleife enthalten, werden nicht zugelassen.

WRP arbeitet nach dem gleichen Prinzip wie der verteilte DV Algorithmus. Die Routingtabellen werden unter den Nachbarn ausgetauscht, und neue Ziele oder veränderte Distanzen werden in die eigene Routingtabelle übernommen. Allerdings muss vor jeder Änderung geprüft werden, ob dadurch eine Schleife entsteht. Solche Änderungen dürfen nicht in die Routingtabelle übernommen werden.

Dieser Algorithmus hat mehrere Vorteile: er berechnet die Routen über das Netzwerk verteilt, und in den Routingtabellen sind alle Informationen vorhanden, um Schleifenfreiheit zu garantieren. Da der Algorithmus alle Berechnungen lokal ausführt, ist er fähig zu entscheiden, ob eine Topologieveränderung im ganzen Netzwerk weitergemeldet werden muss oder einfach ignoriert werden kann. Damit ist die Skalierbarkeit von WRP allen anderen Algorithmen überlegen, die ihre Suchbotschaften oder Topologieinformationen immer im ganzen Netzwerk verbreiten müssen. Aus diesem Grund verwenden auch die im Rahmen dieser Arbeit entwickelten Algorithmen eine Baumstruktur in ihren Tabellen. Während dieser Weiterentwicklungen [Küh00, Tür00, JF01] hat sich auch gezeigt, dass mit der Baumstruktur weitere Optimierungen wie beispielsweise eine Lastverteilung möglich ist.

3.3.10 Landmark Routing

Das Landmark Routing [Tsu88] ist einer der interessantesten Ansätze für das Routing in großen Netzwerken. Landmark Routing baut selbstständig eine Hierarchie unter den Knoten auf. Die Organisation der Hierarchie richtet sich nach den Positionen der Knoten, dabei werden nahe zusammenliegende Konten zu einer Einheit zusammengefasst. Dieses Vorgehen vereinfacht das Routing in großen Netzen erheblich, denn durch das Zusammenfassen von Knoten zu größeren Einheiten müssen viel weniger Routen gepflegt werden.

Der Begriff „Landmark“ veranschaulicht den Routingvorgang. Durch die Gruppenbildung wird ein Pfad nicht mehr in Richtung eines speziellen Knotens, sondern zuerst in die Richtung einer Gruppe berechnet. Die Gruppe wird durch einen zentralen Knoten, dem Landmark, repräsentiert. Der erste Knoten, um den sich eine Gruppe bildet, ist das Landmark. Das Routing erfolgt durch die Hierarchien von Landmark zu Landmark. Dadurch ist das Routing sehr intuitiv, denn auch Menschen planen Wege von einer größeren Stadt zur nächsten und arbeiten sich so in die richtige Richtung voran.

Die Routen werden letztlich mit einem Distanz-Vektor Algorithmus ausgerechnet. Ein besonderer Vorteil liegt in den durch die Hierarchien wesentlich keineren Tabellen. Beim Landmarkrouting ist die Anzahl der Hierarchiestufen nicht begrenzt. Der Algorithmus kann also sogar neue Hierarchiestufen hinzufügen, wenn die Anzahl der Knoten steigt.

Durch die Bildung von Gruppen entstehen aber auch Probleme, denn bevor ein Pfad zu einem Knoten bestimmt werden kann, muss dessen Gruppenzugehörigkeit bekannt sein. Knoten können eine Gruppenzugehörigkeit nicht beliebig lange aufrechterhalten, denn durch die Mobilität der Knoten lösen sich die Nachbarschaftsbeziehungen und damit auch die Einheiten immer wieder auf. Aus diesem Grund ist eine verteilte Verwaltung notwendig, bei der die Gruppen-

zugehörigkeit aller Knoten im Netzwerk erfasst wird. Vor dem Routen muss ein Sender also die Gruppenzugehörigkeit des Ziels erfragen. Aus diesem Grund ist Landmark-Routing relativ kompliziert zu implementieren: es wird ein Algorithmus benötigt, der fortwährend die lokalen Gruppen pflegt, außerdem eine verteilte Datenbank, die für alle Knoten die Gruppenzugehörigkeit speichert, und der Routingalgorithmus selbst, der letztlich die Routen errechnet.

Die komplizierte Funktionsweise hat lange Zeit verhindert, dass eine frei nutzbare Implementierung zur Verfügung steht. Inzwischen existiert eine Spezifikation [GHM01], die allerdings eine manuell festgelegte Gruppenzugehörigkeit benutzt. Das US-Verteidigungsministerium hat einige Forschungsaufträge zur Analyse von Landmark und vergleichbaren Algorithmen für große Netzwerke in Auftrag gegeben. Bisher sind jedoch nur Simulationsergebnisse für die eingeschränkte Spezifikation erhältlich [GHP00].

3.3.11 Cluster Based Routing Protocol

Das Cluster-Based Routing-Protocol (CBRP) [JLT98] teilt die Stationen eines Netzwerks in sich teilweise überlappende Gruppen auf, die als Cluster bezeichnet werden. Dadurch dürfen Stationen im Randbereich eines Clusters auch mehreren Gruppen angehören. Die Gruppen bilden sich aus einer Ansammlung von Stationen, die sich zuerst einen Clusterhead wählen, wobei die Station mit der kleinsten Seriennummer gewinnt. Alle direkten Nachbarn der Station sind dann diesem Cluster zugeordnet, dessen Nummer einfach durch die Seriennummer des Clusterhead bestimmt wird. Die Beziehungen der Stationen untereinander müssen aber durch regelmäßig ausgesendete Kennungen geprüft werden, um die Cluster eventuell neu zu organisieren, dadurch werden die Stationen fortwährend belastet. Eine Route wird, wie bei DSR, durch Fluten gesucht. Die dazu notwendigen Botschaften werden aber nur über die Clusterheads geschickt. Da in dichten Netzwerken jeder Cluster eine große Zahl an beigeordneten Stationen hat, denen das Fluten erspart bleibt, reduziert dieses Vorgehen insbesondere in dichten Netzwerken die Botschaftenanzahl für das Fluten.

CBRP ist innerhalb des Clusters ein proaktives, außerhalb ein reaktives Protokoll. Es erkauft sich das schnellere und wesentlich effizientere Fluten durch die Gruppenbildung. Der Verwaltungsaufwand für die Bildung und Pflege von Gruppen ist der Hauptnachteil von CBRP.

3.3.12 Zonerouting

Im Zone Routing Protocol (ZRP) [HP98] werden zwei unterschiedliche Routingprotokolle eingesetzt und Zonen festgelegt, die eine bestimmte Menge von Stationen als Nachbarschaft definieren. Das erste Protokoll ist für die Weiterleitung der Pakete innerhalb der eigenen Zone verantwortlich, das zweite ist für das Inter-Zonen-Routing zuständig. In der Spezifikation der beiden Routingalgorithmen [HPS01a, HPS01b] ist das innerhalb der Zone verwendete Routing proaktiv, dagegen ist das Inter-Zonen-Routing ein reaktiver Algorithmus.

Mit dieser Aufteilung kann in der lokalen Umgebung immer auf gut gepflegte Routen zurückge-

griffen werden. Für ein Routing über die Zone hinaus wird dann eine besondere Art des Flutens für die Suchbotschaften eingesetzt. Das Fluten ist im ZRP durch die Kenntnis der Topologie lokaler Zonen wesentlich effektiver als normales Fluten. Es ist nicht mehr notwendig, alle Stationen eine Botschaft wiederholen zu lassen, sondern es reicht aus, zu prüfen, ob entweder eine Suchbotschaft eine Station in der eigenen Zone erreichen möchte oder ob die Suchbotschaft an andere Zonen weitergereicht werden muss. Im ersten Fall wird die Suchbotschaft direkt über den bekannten Pfad an die richtige Station weitergeleitet, im zweiten Fall sind die Pfade bis zur Zonengrenze an die benachbarten Zonen bekannt.

ZRP erkauft sich so den Vorteil eines effizienten Flutens durch die dauernde Pflege der Routen in der nahen Umgebung. Damit sind Ruhezeiten ohne Netzaktivitäten, wie sie DSR anbietet, nicht möglich. Die Effizienz von ZRP hängt nun stark davon ab, wie groß der Anteil der Inter-Zonen-Routen ist, da sie die teuren Suchanfragen auslösen.

3.4 Klassifikation der Algorithmen

Für die folgende Klassifizierung ist die von den Algorithmen gespeicherte Information über die Netztopologie von zentraler Bedeutung. Diese Information entscheidet darüber, wie intelligent die Algorithmen reagieren können, gleichzeitig bestimmt sie aber auch den Kommunikationsbedarf eines Algorithmus. Die im Rahmen dieser Arbeit entwickelten Algorithmen operieren auf Graphen, da nur diese Verfahren entsprechend erweiterbar sind. Die verwendete Einteilung spezialisiert sich daher auf die Algorithmen, die für die weitere Arbeit von besonderer Bedeutung sind. Diese Algorithmen erfassen kontinuierlich das gesamte Netzwerk und errechnen andauernd die aktuellen Routen. Algorithmen, die Routen nur bei Bedarf ermitteln, werden von den hier vorgestellten Klassifizierungsmerkmalen nicht erfasst, da die in Ihnen gespeicherte Information sich mit dem Bedarf verändert. Klassifikationen für diese Art von Algorithmen finden sich in [RT99].

Ein Algorithmus, der ständig die gesamte Netztopologie kennt, kann jederzeit den kürzesten Pfad und sogar Alternativpfade errechnen. Um das Wissen über die Netztopologie auf dem aktuellen Stand zu halten, sind jedoch andauernde Updates nötig, die entsprechend hohe Kommunikationskosten verursachen. Die Alternative sind Algorithmen, die ausschließlich mit lokalen Informationen arbeiten und deswegen nur Teile der Netztopologie speichern müssen. Durch die beschränkten Informationen sind dann auch die Möglichkeiten dieser Algorithmen, z.B. Alternativrouten vorzuhalten, deutlich eingeschränkt oder nicht vorhanden.

Aufteilung nach Informationsgehalt

Die hier vorgestellte Klassifizierung teilt Routingalgorithmen in Klassen ein, wobei in der Klasse mit der größten Informationsmenge alles über die Netzwerktopologie gespeichert wird und in der Klasse mit der geringsten Informationsmenge keine Topologie mehr gespeichert wird.

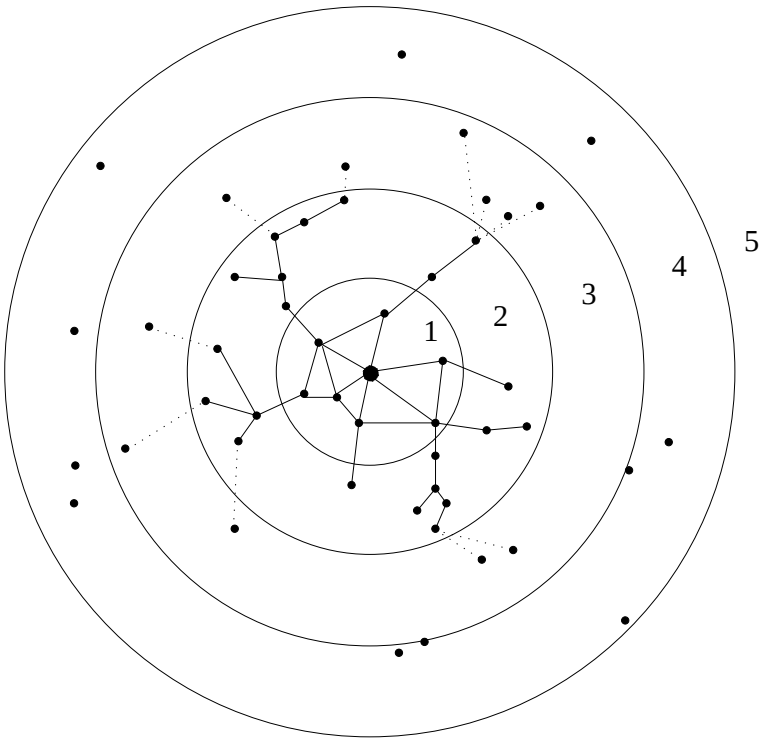


Abbildung 3.10: Die verschiedenen Klassen der Routinginformation

Abbildung 3.10 veranschaulicht eine Möglichkeit, die gespeicherte Informationsmenge in Klassen einzuteilen. In der hier vorgestellten Einteilung werden fünf Klassen verwendet. Die erste Klasse steht für die maximale Informationsmenge und die weiteren Klassen enthalten immer weniger Informationen. Die dargestellten Verbindungen repräsentieren die Informationen, die der Knoten im Zentrum der Abbildung in den jeweiligen Klassen zur Verfügung hat.

- Klasse 1

  In dieser Klasse kennt der Knoten alle anderen Knoten und deren Verbindungen. Er kennt damit den vollständigen Graphen, der dieses Gebiet beschreibt. Mit diesem Wissen lassen sich alle möglichen Routen zu anderen Knoten berechnen.

- Klasse 2

  Hier sind alle Knoten und die Verbindungen für **einen** Weg dorthin bekannt. Da alle Querverbindungen fehlen, reduziert sich die Information über den Graphen dieser Zone zu einem Baum.

- Klasse 3

  Hier ist nur noch die Existenz und eine Distanzangabe zu den Knoten bekannt. Eventuell sind weitere lokale Informationen z.B. über Nachbarn verfügbar. Mit diesen Informationen lässt sich der Graph nicht rekonstruieren. Deshalb können bei der Routenberechnung leicht Schleifen entstehen.

– Klasse 4

  Von den Knoten ist nur noch die Existenz bekannt. Es muss eine eindeutige Kennung des Knotens verfügbar sein. Damit kann der gewünschte Knoten gesucht werden kann.

– Klasse 5

  Die Knoten sind vollständig unbekannt. Solche Knoten lassen sich nur durch Fluten erreichen. Soll ein Knoten gesucht werden, der eine bestimmte Eigenschaft oder einen bestimmten Namen hat (nicht zu verwechseln mit der eindeutigen Kennung), so muss eine Suchanfrage geflutet werden. Dabei ist nicht vorhersehbar, ob die Suche nach einem Knoten erfolgreich sein wird.

Die in Abschnitt 3.2 vorgestellten Basisalgorithmen lassen sich eindeutig den angegebenen Klassen zuordnen. Der Link-State Algorithmus sammelt Informationen entsprechend der Klasse 1, der Distanz-Vektor Algorithmus verwendet Informationen aus Klasse 3, und ein Fluten benötigt keine Informationen und ist deshalb der Klasse 5 zuzuordnen.

Die verwendete Metrik für die Kantengewichte wird bei dieser Klasseneinteilung nicht berücksichtigt. Für die Einteilung ist nur die Information über die Existenz von Kanten maßgeblich. Da Routingalgorithmen teilweise sehr aufwändige Metriken verwenden, z.B. die Auslastung einer Kante, können dadurch innerhalb einer Klasse erhebliche Unterschiede in der Informationsmenge und beim Kommunikationsaufwand entstehen.

Die Tabelle 3.4 ordnet den Routingalgorithmen die nun eingeführten Klassen zu.

	DSDV	OLSR	GSR	STAR	WRP	Fluten
Informationsklasse	3	1	1	2	2	5
Kosten einer Topologieänderung:						
Rechenkompl.	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N)$	0
Botschaftenkompl.	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$	0
Kommunikationskosten:						
Optimierungziel	kürzeste Pfade			nutzbare	kürzeste	-
Pfadlänge.	$O(d)$	$O(d)$	$O(d)$	$O(d)$	$O(d)$	$O(N)$
Sonstige Eigenschaften:						
Sendet Kennungen	J	J	N	J	J	N
Flutet Nachrichten	N	J	N	N	N	J
Benutzte Tabellen	2	4	3	6	4	1
Lokale Reparaturen	J	N	N	J	J	-

Tabelle 3.4: Eigenschaften der Routingalgorithmen

Tabelle 3.4 gibt für die Algorithmen Destination Sequenced Distance Vector (DSDV), Optimized Link State Routing (OLSR), Source Tree Adaptive Routing (STAR), Global State Routing (GSR), Wireless Routing Protocol (WRP) und das Fluten (mit Zyklensperre) eine Klassifizierung und eine kurze Übersicht zu weiteren Eigenschaften an. Die Angaben zur Aufwandsanalyse wurden [RT99, GP00] und [CG98] entnommen.

Bei der Aufwandsanalyse sind Werte für die Pflege der Routen dem späteren Aufwand bei der Nutzung der Routen gegenübergestellt. Die Kosten für die Pflege von Routen werden durch den Berechnungsaufwand und die Anzahl benötigter Botschaften bei einer Topologieänderung angegeben. Die Topologieänderung entsteht dabei entweder durch Wegfall oder das Hinzufügen einer Leitung im Netzwerk. Die jeweils angegebene Rechenkomplexität ist bezüglich der Gesamtkontenzahl N angegeben und reicht von 0 beim berechnungslosen Fluten[1] bis zu $O(N^2)$ bei OLSR, GSR und STAR. Der Aufwand entsteht dabei in jeder Station. Um den Gesamtaufwand im Netzwerk zu erhalten, müssen die angegebenen Werte nochmals mit N multipliziert werden. Die Botschaftenkomplexität bezeichnet die Anzahl der im gesamten Netzwerk versendeten Botschaften.

Die Routingberechnungen optimieren die Kommunikationskosten, daher ist in der Tabelle angegeben, welches Optimierungsziel von den Algorithmen angestrebt wird. Die meisten Algorithmen berechnen möglichst kurze Pfade zwischen den Stationen, nur Fluten optimiert überhaupt nicht und STAR kann sich statt mit kürzesten Pfaden auch mit funktionierenden Pfaden begnügen. Leider ist der Vorteil, der dadurch bei der Routenberechnung entsteht, nicht so groß, dass sich eine erkennbare Veränderung (um eine Größenordnung) bei den Kosten einer Topologieänderung ergibt. Die folgende Zeile gibt die maximale Länge eines Pfades an. Die Anzahl ist für die meisten Stationen $O(d)$, wobei d den Diameter des Netzwerks bezeichnet. Fluten bezahlt den Vorteil beim Berechnungsaufwand hier mit Kosten von $O(N)$.

Als weitere Eigenschaften der Algorithmen sind noch der Bedarf an regelmäßig gesendeten Kennungen (engl. beaconing) und die Nutzung gefluteter Botschaften angegeben. Nur GSR und Fluten verzichten darauf, jedoch ersetzt GSR die Kennungen durch seine ohnehin regelmäßig ausgesendeten Tabellen, die diese Funktion übernehmen.

Die letzten beiden Zeilen beschreiben die Reparaturfähigkeiten der Algorithmen und die Anzahl der zur Routenverwaltung benötigten Tabellen. Die Bezeichnung „lokale Reparaturen“ beschreibt dabei die Möglichkeit, eine defekte Route durch eine auf wenige Stationen begrenzte Operation zu korrigieren. Diese Funktion ist in Ad-hoc Netzwerken häufig nötig und spart viel Aufwand gegenüber dem sonst üblichen kompletten Löschen und Neuberechnen einer Route.

Schlussfolgerungen

Die in der Tabelle 3.4 angegebenen Komplexitäten stellen die typischen Eckwerte für Routing- und Kommunikationskosten in Ad-hoc Netzwerken dar. Dabei ist Fluten das eine Extrem mit den niedrigsten Berechnungskosten, aber dem höchsten Aufwand bei der Kommunikation. Das andere Extrem ist z.B. WRP, dort fallen regelmäßige Kosten für die Aussendung der Kennungen und die Routenberechnung an. Dafür sind aber die Kommunikationskosten mit $O(d)$ wesentlich geringer, da in der Regel $d \ll N$ gilt.

Die hier nicht betrachteten reaktiven Algorithmen liegen mit ihren Kennwerten ebenfalls zwi-

[1] Der Aufwand für die Zyklensperre ist mehr den Kommunikationskosten zuzurechnen und wird hier vernachlässigt.

schen Fluten und WRP. Die Kommunikationskosten unterscheiden sich bei beiden Algorithmenklassen kaum [RT99]. Aber das Auffinden einer Route benötigt bei reaktiven Algorithmen jeweils einen Suchvorgang mit der Botschaftenkomplexität $O(N)$. Allerdings entsteht dieser Aufwand pro benötigter Route, und nicht pro Topologieänderung. Die beiden Algorithmenklassen lassen sich daher nicht direkt miteinander vergleichen.

Anhand der vorgestellten Ergebnisse lässt sich bereits erkennen, dass die Auswahl eines Routingalgorithmus aus der Abwägung der Berechnungskosten gegenüber den Kommunikationskosten erfolgen muss. Für ein Netzwerk mit hoher Mobilität und sehr kleinem Kommunikationsbedarf ist das Fluten oder das damit verwandte DSR immer vorzuziehen. Für Netzwerke mit hoher Kommunikationslast sind Algorithmen wie WRP die richtige Wahl.

3.5 Zusammenfassung

In diesem Kapitel wurde das Routing in drahtlosen Netzwerken ausführlich besprochen. Zu Beginn des Kapitels wurde der Begriff durch seine Aufgabenstellung definiert. Dabei wurden die wichtigsten Anforderungen an das Routing bereits vorgestellt, wobei die besonderen Anforderungen für Ad-hoc Netzwerke im Vordergrund standen.

Danach wurden die bekannten Standardalgorithmen wie Distanz-Vektor und Link-State erläutert. Dabei wurden jedes Mal auch die Schwachpunkte der Standardalgorithmen dargestellt, die beim Einsatz in Ad-hoc Netzwerken zu Problemen führen. Es wurde dargelegt, dass – wegen der Schwachpunkte – spezialisierte Ad-hoc Algorithmen notwendig sind. Diese wurden anschließend vorgestellt und eingeordnet. Die Algorithmen wurden, zur besseren Übersicht, nach ihrem Funktionsprinzip in mehrere Algorithmengruppen unterteilt.

Das Kapitel endete mit einem Vergleich von graphenbasierten Routingalgorithmen. Im Vergleich wurde zuerst eine Klassifizierung nach der Informationsmenge vorgestellt, die in den Stationen gespeichert ist. Danach wurden die Eigenschaften der Algorithmen anhand einer Tabelle gegenübergestellt. Eine Auswertung fasste die in der Tabelle angegebenen Eigenschaften dann nochmals zusammen.

Kapitel 4

Analyse von Routingalgorithmen

4.1 Einleitung

In diesem Kapitel werden Untersuchungsmethoden für die Bewertung von Routingalgorithmen vorgestellt. In drahtlosen Netzwerken mit mobilen Knoten ist ein Vergleich von Routingalgorithmen schwierig, da viele Algorithmen sehr spezielle Eigenschaften besitzen, die sich nicht direkt miteinander vergleichen lassen. Es werden daher verschiedene Analyseverfahren vorgeschlagen und deren Möglichkeiten und Grenzen aufgezeigt.

Dabei wird besonders auf Simulationstechniken eingegangen, da diese Untersuchungsmethode am häufigsten in der Arbeit eingesetzt wird. Das dazu notwendigen Simulationsmodell wird ausführlich diskutiert. Eine Simulatorarchitektur für das Modell wird vorgestellt und die Implementierung beschrieben. Der Simulator wird anschließend genutzt, um den Paketdurchsatz eines drahtlosen Netzwerks bei unterschiedlichen Kollisionsvermeidungsstrategien zu untersuchen.

Die in diesem Kapitel vorgestellte Simulatorarchitektur wurde bereits in [FJ97] veröffentlicht. Die Ergebnisse der Simulationen zum Paketdurchsatz wurden in später in [Jan98] publiziert.

4.2 Analyseverfahren

Zur Analyse von Routingalgorithmen stehen eine ganze Reihe von Methoden zur Verfügung, die sich grob in drei Gruppen zusammenfassen lassen. Die erste Gruppe analysiert einen Algorithmus durch mathematische Methoden. Die Komplexitätsanalyse, wie sie schon in Tabelle 3.4 angewendet wurde, ist ein Beispiel für diese Vorgehensweise. Die zweite Methode ist ein praktischer Ansatz. Dabei werden Algorithmen implementiert und anschließend durch Messungen bei Testläufen geprüft. Die letzte Gruppe versucht mittels Simulationen die Eigenschaften eines Algorithmus zu erfassen. Jede Vorgehensweise hat spezifische Vor- und Nachteile, die in den

nächsten Abschnitten vorgestellt werden. Für eine unterhaltsame, gleichzeitig aber auch sehr ausführliche Beschreibung von Analysemethoden für Computersysteme sei das Buch [Jai91] empfohlen.

4.2.1 Analytische Modelle

Algorithmen lassen sich am schnellsten durch mathematische Methoden bewerten. Eine sehr bekannte Methode ist die Komplexitätsanalyse von Algorithmen. Dabei wird der Ressourcenbedarf eines Algorithmus abgeschätzt. Zur vollständigen Erfassung der Eigenschaften müssen oft mehrere Ressourcen in die Betrachtung mit einbezogen werden. Für die hier untersuchten Routingalgorithmen sind die Rechenkapazität, der Speicherverbrauch und der Verbrauch an Bandbreite wichtige Kriterien.

Die Abschätzung erfolgt dabei in Klassen, die sich jeweils um eine Größenordnung unterscheiden. Diese Unterteilung ist relativ grob, und ermittelt deswegen bei einigen Algorithmen keine unterscheidbaren Ergebnisse. Die Tabelle 3.4 ist ein gutes Beispiel für dieses Problem. Algorithmen, die der gleichen Kategorie angehören, haben in dieser Analyse identische Komplexitäten. Die Komplexitäten der Link-State Algorithmen OLSR und GSR sind ebenso identisch wie die DV Algorithmen DSDV und WRP. Die Komplexitätsanalyse eignet sich daher auch besser, um Algorithmenklassen voneinander abzugrenzen, als einzelne Algorithmen zu bewerten.

Die Komplexitäten lassen sich für den ungünstigsten Fall (engl. worst case analysis) oder für den Durchschnittsfall (engl. average case analysis) durchführen. Dabei treten erhebliche Unterschiede auf. Der in Abschnitt 3.2.1 beschriebene DV Algorithmus hat im ungünstigsten Fall eine Botschaftenkomplexität von $O(N^3)$, die in einem normalen Netzwerk unakzeptabel ist. Die durchschnittliche Botschaftenkomplexität liegt jedoch unterhalb von $O(N^2)$, weshalb der Algorithmus im Internet noch häufig benutzt wird. Die gravierenden Unterschiede zwischen dem ungünstigsten und dem durchschnittlichen Fall verzerren oft das Analyseergebnis. Besonders in großen Systemen, mit vielen Eingangsparametern, hat der ungünstigste Fall kaum eine Aussagekraft über die Funktionseigenschaften eines Algorithmus. Der ungünstige Fall ist eindeutig und daher leichter zu ermitteln als der Durchschnittsfall. Für diesen besteht das Problem in der Festlegung der Eingangsparameter, die ein Standardsystem beschreiben müssen. Hier entstehen die selben Probleme, die auch im Abschnitt über Simulationen noch ausführlich angesprochen werden. Die dort ausgeführte Modellbildung für ein durchschnittliches und faires System nimmt den größten Teil der Beschreibung ein.

4.2.2 Implementierung und Messung

Ein völlig anderer Ansatz ist das Messen und Austesten eines bereits implementierten Algorithmus. Dabei ergeben sich allerdings eine ganze Reihe von Schwierigkeiten, die oft einen fairen Vergleich von Algorithmen verhindern.

Die Vorteile der Methode ergeben sich aus dem direkten Praxisbezug. Wenn die Implementie-

rung vorhanden ist, dann kann ein Algorithmus direkt unter Einbeziehung aller relevanten Nebenbedingungen getestet werden. Die dabei erzielten Ergebnisse entsprechen – korrekte Messungen vorausgesetzt – den beim späteren Einsatz auftretenden Eigenschaften. Bei der Implementierung werden viele Probleme bereits erkannt und gelöst. Daher stellt sich bei dieser Methode auch nicht mehr die Frage nach der Umsetzbarkeit oder der Dauer bis zur Fertigstellung eines nutzbaren Systems.

Die erste Schwierigkeit dieses Vorgehens ergibt sich schon bei der Umsetzung des Algorithmus in das Testsystem. Dabei hängt es oft von Implementierungsdetails ab, wie schnell der Algorithmus später seine Aufgabe erfüllt. Das Testsystem selbst hat auch einen großen Einfluss auf das spätere Ergebnis. Wenn die Implementierung des Algorithmus später in exakt diesem System eingesetzt wird, dann sind die Messergebnisse sehr aussagekräftig. Anhand der Messungen kann aber kaum abgeschätzt werden, welches Verhalten der Algorithmus in ähnlichen Systemen zeigt. So ist beispielsweise in Ad-hoc Netzwerken die bei der Messung verwendete Nutzlast von entscheidender Bedeutung. Werden dabei nur wenige Kommunikationsverbindungen geprüft, dann erhalten reaktive Algorithmen eine bessere Bewertung als proaktive Algorithmen. In Systemen mit vielen Kommunikationsbeziehungen kehrt sich das Bild entsprechend um.

Für den Test von Ad-hoc Routingalgorithmen ergeben sich noch weitere Probleme. Ein Versuchssystem benötigt ein ganzes Netzwerk von Funkstationen, die bewegt werden müssen. Die Kosten für dieses System sind durch die große Anzahl von Stationen relativ hoch. Durch die notwendigen Bewegungen und die in Funksystemen zufällig auftretenden Störungen lassen sich Versuche nicht exakt wiederholen. Die so entstehenden Varianzen in den Ergebnissen erschweren zusätzlich die Auswertung von Algorithmenvergleichen.

4.2.3 Simulationen

Simulationen ermöglichen den Test eines Algorithmus in einer exakt definierten Umgebung. Die erzielten Ergebnisse sind dabei nie so realitätsnah wie in der oben beschriebenen Methode. Aber Simulationen bieten eine Reihe von anderen Vorteilen.

Durch die exakt definierte Umgebung ist eine Simulation beliebig wiederholbar, wodurch Algorithmenvergleiche unter identischen Bedingungen möglich sind. Im Falle eines Fehlers ist eine Simulation auch Schritt für Schritt nachvollziehbar. Dies erleichtert die Fehleranalyse erheblich.

Die Simulation eines großen Netzwerks ist - entsprechende Rechenleistung und Speicherkapazität vorausgesetzt - ohne Probleme möglich. Damit erlaubt diese Methode auch die Untersuchung von Netzwerken, die durch die anderen Methoden entweder nicht mehr überschaubar oder einfach nicht realisierbar sind.

Der größte Nachteil der Simulationen liegt in der notwendigen Einschränkung des betrachteten Systems. Die Vorgänge in einem Funknetzwerk sind zu komplex, um sie realitätsgetreu im Simulator nachzustellen. Das Simulationsmodell ist deshalb eine stark vereinfachte Abbildung der Vorgänge. Dies ist notwendig, um die Rechenzeit für die Simulationen auf einem erträglichen Niveau zu halten. Ein einfaches Modell hat zusätzlich den Vorteil, wenige Eingangsparameter

zu besitzen.

Für jede Simulation müssen die Eingangsparameter festgelegt werden, dazu gehören unter anderem die Anzahl der Stationen, der verwendete Graph und die Mobilität der Stationen. Je größer die Zahl dieser Parameter wird, um so schwieriger ist die Festlegung einzelner Werte, so dass letztlich ein realitätsnahes und aussagekräftiges Modell ensteht. Die folgenden Abschnitte analysieren z. B. die Auswirkungen der Funkreichweite auf das Routing. Wenn die Reichweite zu groß gewählt wird, dann können sich alle Stationen direkt erreichen; ein Routing ist überflüssig. Ist die Reichweite dagegen zu gering, kommt keine Kommunikation mehr zustande und jedes Routing-Verfahren ist dann chancenlos. Der nächste Abschnitt beschreibt das in der Simulation verwendete Netzmodell und die dort eingesetzten Parameter.

4.3 Definition eines Paketfunknetzwerkmodells

4.3.1 Abschätzung der Erreichbarkeit

In den Modellen für drahtlose Netzwerke wird immer von einer Dynamik in den Netzwerken ausgegangen. Durch die Bewegungen der Benutzer oder Veränderungen im Übertragungsmedium gehen Verbindungen verloren, während andere neu entstehen. Diese Beschreibung behandelt die Auswirkungen von Knotenbewegungen auf die Verbindungen des Netzes. Die Untersuchung ist Voraussetzung für eine genaue Kenntnis der Zusammenhänge von Bewegungsmustern und Netzwerkänderungen, die für eine analytische Betrachtung von Algorithmen erforderlich ist.

Definition der Netzwerkparameter

Ein Netzwerk kann durch eine Adjazenzmatrix beschrieben werden. Eine Adjazenzmatrix zeigt im einfachsten Fall alle Verbindungen zwischen den existierenden Knoten an. In komplizierteren Fällen können die Verbindungen auch beispielsweise qualitativ bewertet werden. In den Zeilen der Matrix sind alle möglichen Sender aufgelistet, in den Spalten die Empfänger. Bei den hier untersuchten Netzen ist jeder Sender gleichzeitig auch ein Empfänger, deswegen ist die Anzahl der Sender gleich der Anzahl der Empänger und die Matrix ist daher quadratisch. Eine Verbindung zwischen zwei Knoten wird durch eine 1 in der entsprechenden Zeile und Spalte gekennzeichnet. Alle anderen Matrixelemente werden zu 0 gesetzt. Die Diagonale der Matrix enthält nur Einträge mit dem Wert 1, da jeder Sender sich selbst erreichen kann.

Alle derzeit verwendeten Funksysteme bestätigen ihre Botschaften. Deshalb ist für eine erfolgreiche Kommunikation immer eine Verbindung notwendig, die in beiden Richtungen arbeitet. In der weiteren Untersuchung werden ausschließlich ungerichtete Verbindungen betrachtet. Damit ist die Matrix immer symmetrisch, und die Untersuchung kann auf eine Hälfte der Matrix beschränkt werden.

Die Adjazenzmatrix ist eine einfache Beschreibungsform, in ihr sind die Zustände jeder möglichen Verbindung eines Netzwerkes beschrieben. Ein beliebiges Netzwerk kann durch eine Adjazenzmatix dargestellt werden. In der Matrix werden nur die Nachbarschaftsbeziehungen aber keine Positionen der Knoten gespeichert. Somit lassen sich aus einer Matrix mehrere Darstellungen eines Netzwerkes erzeugen.

Die Simulation erfasst die Bewegungen von Stationen auf einem rechteckigen Feld. Das Gesamtfeld hat die Fläche $A = X \cdot Y$, auf dem die Knoten beliebig positioniert werden können. Die Position eines Knotens wird durch seine Koordinaten (x, y) angegeben, mehrere Knoten können gleichzeitig einen Punkt belegen.

Die Knoten bauen ihre Verbindungen über Funk auf. Das Modell sieht eine einheitliche Reichweite R für alle Sender vor. Ein Knoten hat eine Verbindung zu einem Nachbarknoten, wenn der Abstand zwischen den beiden Knoten kleiner R ist. Durch diese Definition sind die Verbindungen immer in beide Richtungen nutzbar.

Die Bewegungssteuerung wählt für jeden Knoten zuerst einen Startpunkt und einen Zielpunkt sowie eine Bewegungsgeschwindigkeit zufällig aus. Die Koordinaten werden zufällig mit einer Gleichverteilung aus dem Wertebereich $(0, 0)$ bis (X, Y) ermittelt, so dass die Punkte gleichmässig auf dem Feld verteilt sind. Die einzelnen Stationen i bewegen sich mit festgelegten Geschwindigkeiten G_i über das Feld. Die Geschwindigkeit für jeden Knoten wird ebenfalls mit einer Gleichverteilung ermittelt, es wird aber nur eine Schwankung von maximal 50 Prozent um die gewünschte Durchschnittsgeschwindigkeit G zugelassen.

Während der Simulation wird in jeder Zeiteinheit die Position der Knoten neu berechnet und eine neue Adjazenzmatrix erstellt. Diese Matrix wird mit der Matrix des vorherigen Schrittes verglichen, und jede Änderung wird den betroffenen Stationen mitgeteilt. Die Simulation hat eine vorgegebene Laufzeit L, die interne Simulationsuhr startet bei 0 und die Simulation wird beendet, wenn die interne Uhr L erreicht hat.

In der folgenden Liste sind alle Parameter des Netzmodells nochmals zusammengefasst:

- N Anzahl der Knoten im Netzwerk.
- R Senderadius der Knoten in Metern.
- X Horizontale Feldausdehnung in Metern.
- Y Vertikale Feldausdehnung in Metern.
- G Durchschnittliche Geschwindigkeit der Knoten in Metern pro Sekunde.
- L Laufzeit der Simulation in Sekunden.

Theoretische Betrachtung des Modells

Die oben beschriebenen Parameter ermöglichen Simulationen zur Leistungsbewertung von Routingalgorithmen. Werden die Parameter allerdings ungünstig gewählt, dann entstehen Situationen, die für eine faire Bewertung ungeeignet sind. Wird beispielsweise die Reichweite zu klein gewählt, dann bildet sich kein verbundenes Netzwerk aus, sondern nur vereinzelte Gruppen, die untereinander nicht kommunizieren können. Der Erfolg einer Übertragung ist dann mehr oder weniger zufällig und die Simulationsergebnisse zeigen große Varianzen.

Eine theoretische Betrachtung kann die Simulation nicht ersetzen, es ist aber möglich, aus den Parametern Kenndaten zu bestimmen, die Obergrenzen definieren. Diese Obergrenzen sind eine Abschätzung der maximalen Leistung, die ein optimaler Routingalgorithmus unter den gegebenen Parametern erreichen kann.

Die Vermaschung

Für alle folgenden Berechnungen ist die Vermaschung V entscheidend. Die Vermaschung wird durch die Anzahl der Kanten eines Netzwerks bestimmt.

Für die Berechnung der Vermaschung werden folgende Parameter benötigt:

- Die Anzahl der Knoten N.
- Die Reichweite der Knoten R. (Dem Modell wird eine einheitliche Reichweite zugrundegelegt.)
- Die Größe des Simulationsfeldes $A = X \cdot Y$.

Die N Stationen befinden sich an einer beliebigen Position auf einem Feld fester Größe A und haben eine feste Sendereichweite R. Damit wird eine Wahrscheinlichkeit V bestimmt, mit der zwei Stationen direkt miteinander kommunizieren können. Diese Wahrscheinlichkeit wirkt auch auf den Erwartungswert für die durchschnittliche Anzahl der Verbindungen einer Station: Eine Station kann maximal $N-1$ Verbindungen besitzen. Da jede mögliche Verbindung mit der Wahrscheinlichkeit V existiert, ist der Erwartungswert der Verbindungen $(N-1) \cdot V$. Dies kann weiterhin auf alle Knoten des Netzes angewendet werden. Die Wahrscheinlichkeit V bestimmt dann die Gesamtzahl der Kanten im Netzwerk. Das Verhältnis der Gesamtzahl der existierenden Kanten zu der Anzahl der theoretisch möglichen Kanten wird als Vermaschung bezeichnet.

Ein vollständig vermaschtes Netz hat bei N Knoten und $(N-1) \cdot N/2$ Kanten eine Vermaschungsrate V von 100 Prozent. Die Vermaschung ist der entscheidende Parameter für alle weiteren Rechnungen und Simulationen, denn sie entscheidet hauptsächlich über den Erfolg von Verbindungen über mehrere Stationen.

Besitzt ein Netzwerk N Knoten und L Kanten, dann ist

$$V = \frac{L \cdot 2}{(N-1) \cdot N} \tag{4.1}$$

Der Senderadius

Für die praktische Anwendung ist der Zusammenhang zwischen Sendereichweite der Knoten und der Vermaschung von Bedeutung: Sind mehrere Stationen auf einem Gebiet der Fläche A gleichmäßig verteilt, dann gibt der Parameter V die Wahrscheinlichkeit an, mit der sich andere Stationen innerhalb der Sendereichweite R einer Station befinden. Die Reichweite R wird als Radius eines Kreises um die Station betrachtet. Der Parameter V beschreibt hierbei die Vermaschung des Knotens in Abhängigkeit von R.

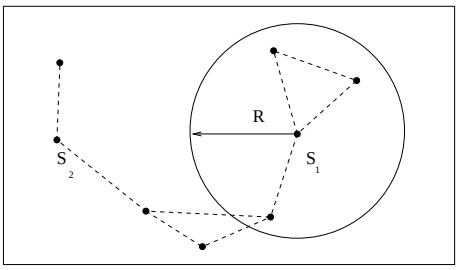


Abbildung 4.1: Darstellung eines Simulationsszenarios

Das Bild 4.1 veranschaulicht diesen Zusammenhang. Im Bild sind Stationen auf einem rechteckigen Feld dargestellt. Für die Station $S1$ ist der Senderadius und die damit abgedeckte Fläche eingezeichnet. Durch die Sendereichweite R wird ein Gebiet $(\pi \cdot R^2)$ festgelegt, das erreicht werden kann. Die Station $S1$ kann andere Stationen erreichen, die sich innerhalb der Sendereichweite R befinden.

Geht man von einer zufälligen Verteilung der Stationen auf dem Feld aus, dann entspricht V dem Verhältnis von $(\pi \cdot R^2)$ zu A. Die Abschätzung ist ungenau, da Punkte des Sendegebietes außerhalb des Spielfelds liegen können. Sie ist aber für die Berechnung verwendbar wenn $(\pi \cdot R^2) \ll A$ gilt und die Fläche in etwa die Form eines Quadrates besitzt. Dann kann V mit folgender Formel abgeschätzt werden:

$$V \approx \frac{\pi \cdot R^2}{A} \tag{4.2}$$

Die Erreichbarkeit

Über ein Netzwerk, dessen Topologie sich andauernd verändert, können nur wenige Aussagen gemacht werden. Die wenigen konkreten Aussagen, die möglich sind, wurden bereits mit den

Simulationsparametern aufgezählt. Weitergehende Aussagen betreffen nur noch Wahrscheinlichkeiten, mit denen bestimmte Ereignisse auftreten oder möglich sind. Für das Routing ist die Erreichbarkeit die wichtigste Eigenschaft dieser Art. Mit ihr wird die Wahrscheinlichkeit angegeben, mit der eine andere Station über einen oder mehrere Router (kurz: Hops) erreichbar sind. Die Erreichbarkeit ist ein Durchschnittswert für jede mögliche Verbindung zwischen zwei Knoten. Dieser Wert ist für einen Routingalgorithmus sehr wichtig, da er die Wahrscheinlichkeit festlegt, ob eine Route existiert, die der Algorithmus finden kann.

Für eine Berechnung der Erreichbarkeit muss ein vollständig verbundenes Netzwerk betrachtet werden. Es wird vorausgesetzt, dass alle Kanten in diesem Netzwerk mit der gleichen Wahrscheinlichkeit existieren, ansonsten ist die Berechnung der Erreichbarkeit zwischen zwei bestimmten Knoten ein NP-vollständiges Problem [GJ79].

Analysen zur Erreichbarkeit von Knoten wurden bereits in Arbeiten über die Zuverlässigkeit von Netzwerken vorgestellt [Col87]. Die Berechnung für eine Wahrscheinlichkeit, dass alle Knoten miteinander kommunizieren können, wird dort All-Terminal Problem genannt. Aus dem All-Terminal Problem läßt sich eine Lösung für das Two-Terminal Problem herleiten, die der Erreichbarkeit entspricht.

Im Folgenden wird nun für ein vollständig verbundenes Netzwerk, bei dem jede Kante die gleiche Existenzwahrscheinlichkeit hat, die Erreichbarkeit berechnet. Die Existenzwahrscheinlichkeit wird dabei durch den Vermaschungsgrad V bestimmt. So erschließt sich der Zusammenhang zwischen Vermaschung und Erreichbarkeit und – durch die oben erläuterte Abschätzung – auch der Zusammenhang zwischen der Reichweite R, der Feldgröße A und der Erreichbarkeit T_n.

Gegeben sei ein vollständig vermaschtes Netzwerk K_n mit n Knoten. In der Berechnung wird die Wahrscheinlichkeit für das Fehlen einer Kante mit q angegeben. Für eine Anwendung auf Ad-hoc Netzwerke kann diese Fehlerwahrscheinlichkeit durch die Vermaschung V bestimmt werden. Damit ist dann $q = 1 - V$. Die Wahrscheinlichkeit, dass alle n Knoten eines Netzwerkes miteinander kommunizieren können, sei A_n.

Es wird ein beliebiger Knoten s aus K_n ausgewählt. Dann wird das Netz in Komponenten zerlegt. Die verschiedenen möglichen Zerlegungen werden danach unterschieden, wie viele Knoten die Komponente enthält, die s einschließt. Für eine bestimmte Komponentengröße j existieren $\binom{n-1}{j-1}$ Möglichkeiten. In jeder dieser Möglichkeiten muß die Komponente intern verbunden sein und alle Kanten zu einem Zielpunkt müssen versagen. Daraus ergibt sich folgende Gleichung:

$$1 = \sum_{j=1}^{n} \binom{n-1}{j-1} A_j q^{j(n-j)} \tag{4.3}$$

Dabei ist A_n die Wahrscheinlichkeit, dass ein Teilnetz existiert in dem alle n Knoten in der selben Komponente sind wie s. Die Geichung kann zur Berechnung von A_n umgeformt werden:

$$A_n = 1 - \sum_{j=1}^{n-1} \left(\begin{array}{c} n-1 \\ j-1 \end{array} \right) A_j q^{j(n-j)} \tag{4.4}$$

Zur Berechnung der Erreichbarkeit zwischen zwei Knoten werden s und t gewählt. Die Knoten können sich nicht erreichen, wenn s in einer Komponente liegt, die t nicht enthält. Diese Komponente kann maximal $n-1$ Knoten umfassen. Daraus ergibt sich die Gleichung:

$$T_n = 1 - \sum_{j=1}^{n-1} \left(\begin{array}{c} n-2 \\ j-1 \end{array} \right) A_j q^{j(n-j)} \tag{4.5}$$

Mit den Gleichungen 4.4 und 4.5 ist die Erreichbarkeit T_n für ein Netzwerk mit einer rekursiven Prozedur berechenbar.

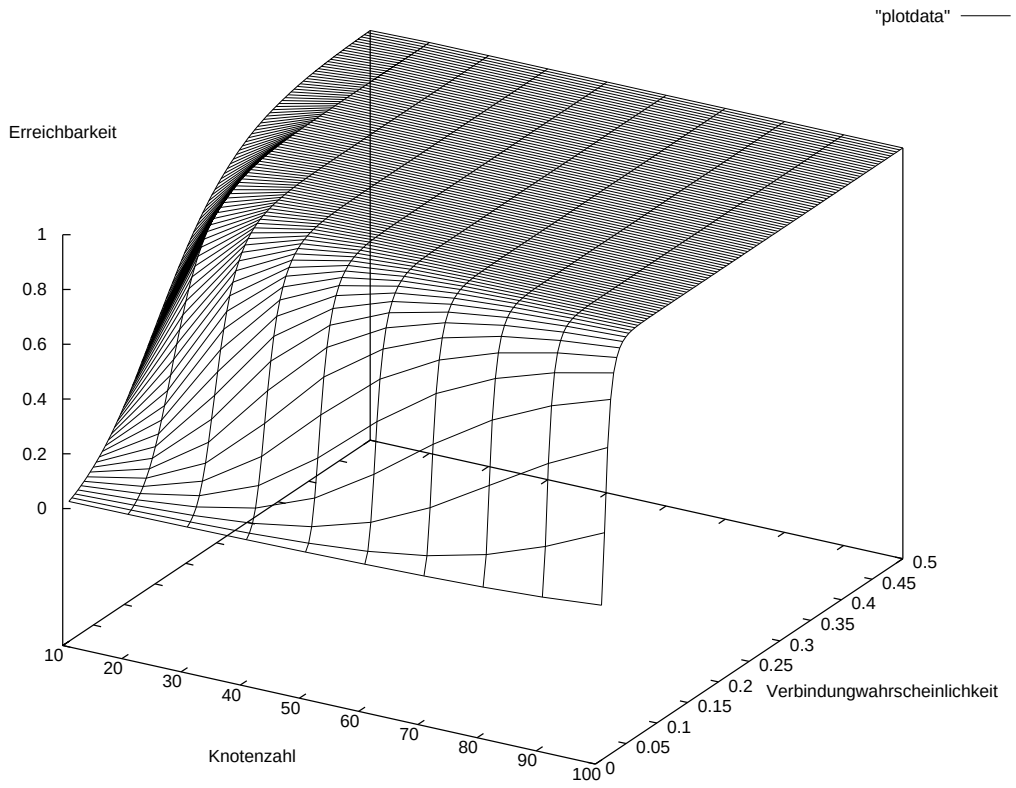


Abbildung 4.2: Darstellung der berechneten Erreichbarkeit

Abbildung 4.2 zeigt einen Funktionsplot der berechneten Wahrscheinlichkeit. Die Erreichbarkeit ist dabei in Abhängigkeit von der Verbindungswahrscheinlichkeit V und der Knotenzahl N dargestellt.

Das Bild enthält die Parameterbereiche, in denen ein Netzwerk so stark partioniert ist, dass ein Routingalgorithmus kaum eine funktionierende Route berechnen kann. Da Routingalgorithmen auch in kritischen Situationen nur wenige Prozent fehlerhafter Routen produzieren, ist es schwierig, statistisch aussagekräftige Vergleiche anzustellen, wenn eine Fehlerquelle existiert, die unabhängig vom Algorithmus wesentlich mehr Fehler verursacht als der Algorithmus selbst.

Für eine faire Simulation ist deswegen eine minimale Erreichbarkeit von mindestens 95 Prozent nötig. Netzwerke mit einer großen Anzahl von Stationen erreichen diesen Wert schon bei einer Vermaschung unterhalb von 10 Prozent. Bei 100 Stationen genügen sogar 3 Prozent, damit braucht im Durchschnitt jede Station 3 funktionierende Leitungen, um ein zusammenhängendes Netz zu bilden. Ein Netzwerk mit 10 Stationen braucht aber für die Erreichbarkeit von 95 Prozent mindestens eine Vermaschung von 50 Prozent. In solch einem Netzwerk kann ein Routingalgorithmus praktisch nicht getestet werden, da die Hälfte aller Ziele schon mit einem Hop erreicht wird. Der Routingalgorithmus muss daher nie komplizierte Routen ermitteln.

4.3.2 Abschätzung der Vermittlungskapazität

Die Grenzen drahtloser Großnetze

Alle Funknetzwerke müssen mit einem beschränkten Funkfrequenzbereich auskommen, den sie von Kontrollbehörden zur Verfügung gestellt bekommen. Funknetze lassen sich danach bewerten, wie gut sie diesen nutzen, das heißt wie viele Benutzer gleichzeitig das Netz in akzeptabler Qualität verwenden können. Eine genaue Untersuchung dieser als spektrale Effizienz bezeichneten Eigenschaft benötigt exakte Definitionen der Netzwerke, des Benutzerverhaltens und eine Bewertungsfunktion für die akzeptable Qualität der Dienstleistungen. Da die Berechnung der spektralen Effizienz sehr aufwändig ist, beschränkt sich die folgende Abschätzung auf den Vergleich der wichtigsten Kenndaten von Zellular- und Ad-hoc Netzwerken. Zellularnetzwerke werden überwiegend für Telefonie eingesetzt. Diese Kommunikationsform garantiert jedem aktiven Teilnehmer eine bestimmte Übertragungskapazität. Aus diesem Grund wird für die folgende Abschätzung auch für die Ad-hoc Netzwerke angenommen, dass eine Reservierung von Kapazitäten für einzelne Verbindungen möglich ist.

Zellularnetzwerke (siehe auch 2.2.1) besitzen eine aufwändige Infrastruktur, die dazu dient, den Funkverkehr auf das letzte unvermeidbare Teilstück des Kommunikationsweges zu beschränken. Es gibt mehrere Gründe, warum der Funkweg möglichst klein gehalten werden soll:

- Durch den frühzeitigen Wechsel ins Festnetz wird die meiste Funkkapazität eingespart. Ein Gespräch belegt dann nur einen Kanal in einer Zelle, wenn es ins Festnetz geht. Geht das Gespräch zu einem anderen Mobilfunkteilnehmer, wird zur Übertragung ein weiterer Kanal in der Zelle des Ziels benötigt.
- Der Funkweg ist wesentlich empfindlicher gegen Störungen als jede Kabelverbindung.

– Die Verzögerungen bei Funkübertragungen sind durch die Fehlerschutzmaßnahmen deutlich größer als in Kabelnetzen.

– Ein kurzer Funkweg erlaubt kleine Zellen und damit eine bessere Wiederverwendung der Frequenzen.

Besonders der letzte Punkt entscheidet über die spektrale Effizienz von Zellularnetzwerken. Die Länge der Funkstrecke bestimmt die benötigte Sendeleistung, und damit gleichzeitig die Intensität der Störungen in benachbarten Zellen. Je kleiner die Störungen sind, desto öfter können die Frequenzen wiederverwendet werden.

Im Gegensatz zu den Zellularnetzwerken wird in einem Ad-hoc Netzwerk die gesamte Kommunikation ausschließlich über Funkstationen geführt. Ein Gespräch wird über eine Kette von Stationen weitergeleitet und belegt in jeder Station einen Teil der dort vorhandenen Funkkapazität. Das bedeutet, ein Gespräch belegt in jeder Station entlang des Kommunikationsweges einen Kanal. Dabei wird die Sendeleistung in jeder Station so geregelt, dass sie ihre Nachbarn gerade noch erreichen kann. Dadurch entsteht praktisch eine Mikrozelle um jede Station. Diese Zellen sind wesentlich kleiner als die Zellen von Zellularnetzwerken. Dafür wird für jedes Gespräch aber eine ganze Kette von Mikrozellen benötigt.

Abschätzung der Vermittlungskapazität eines Netzwerks

Zur Berechnung der nötigen Vermittlungskapazität wird ein Netzwerk mit folgenden Eigenschaften untersucht:

– Das Netz besitzt eine hexagonale Struktur, ähnlich der in Abbildung 2.1

– Das Netz hat N Stationen.

– Die Stationen verteilen sich gleichmäßig über eine Fläche der Größe A.

– Jeder Station führt Gespräche mit einer Häufigkeit von L Prozent der Gesamtzeit zu zufällig gewählten Zielen.

– Jede Station hat ein Kapazitätslimit. Sie kann maximal V Gespräche vermitteln.

– Die Gespräche gehen über die durchschnittliche Distanz D.

– Ein Gespräch benötigt die Vermittlung von durchschnittlich H Stationen.

Durch diese Festlegung wird die Kanalkapazität in einer vereinfachten Form angegeben. In der Telephonie wird dem Kunden üblicherweise eine Verbindung mit einer festen Bandbreite garantiert. Diese Verbindungsart wird in der Rechnung als Gespräch bezeichnet.

Die gesamte im Netzwerk benötigte Vermittlungskapazität K_b ergibt sich dann zu:

$$K_b = N \cdot L \cdot H \tag{4.6}$$

Im Netzwerk steht insgesamt eine Vermittlungskapazität von K_e zur Verfügung:

$$K_e = N \cdot V \tag{4.7}$$

Diese Vermittlungskapazität ist jedoch in jeder Station auf V Gespräche beschränkt.

Für eine beliebige Verteilbarkeit der Kapazität kann dann $K_b = K_e$ gesetzt werden. Daraus folgt:

$$N \cdot L \cdot H = N \cdot V \tag{4.8}$$

oder

$$V = L \cdot H \tag{4.9}$$

Gleichung 4.9 sagt aus, dass die benötigte Vermittlungskapazität nur von der Gesprächshäufigkeit L und der Anzahl der vermittelnden Stationen H abhängig ist. Allerdings steigt H bei wachsendem N ebenfalls an, da in größeren Netzwerken für eine Vermittlung mehr Zwischenstationen gebraucht werden als in kleinen Netzwerken. Der Zusammenhang zwischen N und H kann durch einige zusätzliche Annahmen abgeschätzt werden.

In der Voraussetzung wurde bereits festgelegt, dass die Stationen gleichmäßig als ein hexagonales Netz auf der Fläche A verteilt sind. Werden die Stationen so verteilt, dass sie jeweils in der Mitte eines Hexagons liegen, dann muss jede Station von einer Fläche von $\frac{A}{N}$ umgeben sein.

Die Fläche eines Hexagons berechnet sich zu

$$A_{HEX} = 6 \cdot r^2 \cdot \tan\frac{\pi}{6} = 2 \cdot \sqrt{3} \cdot r^2 \tag{4.10}$$

Damit ist der Abstand D_{ST} zwischen benachbarten Stationen:

$$D_{ST} = 2 \cdot r = 2 \cdot \sqrt{\frac{\frac{A}{N}}{2 \cdot \sqrt{3}}} = \sqrt{\frac{2 \cdot A}{N \cdot \sqrt{3}}} \tag{4.11}$$

Steigt die Anzahl der Stationen im Netzwerk an, dann steigt auch die Dichte der Stationen, wenn die Fläche A sich nicht verändert. Durch die größere Dichte werden für jedes Gespräch mehr Vermittler nötig. Die Steigerung in der Anzahl der Vermittler soll nun abgeschätzt werden.

Durch eine höhere Stationendichte ändert sich am durchschnittlichen Abstand D zwischen den an einem Gespräch beteiligten Stationen nichts. Muss eine Verbindung zwischen zwei Knoten aufgebaut werden, die M Meter voneinander entfernt sind, dann sind dazu mindestens $H(M)$ Vermittler nötig:

$$H(M) = \frac{M}{D_{ST}} = \frac{M}{\sqrt{\frac{2 \cdot A}{N \cdot \sqrt{3}}}} = \sqrt{\frac{M^2 \cdot N \cdot \sqrt{3}}{2 \cdot A}} \tag{4.12}$$

Aus der Formel ist ersichtlich, dass $H(M)$ proportional zu $\sqrt{N}$ ansteigt. Aus der Gleichung $V = L \cdot H$ folgt, dass auch V proportional mit $\sqrt{N}$ ansteigen muss um die Gespräche zu ermöglichen.

In der Voraussetzung war V als die maximale Vermittlungskapazität einer Station definiert. Die Vermittlungskapazität ist durch die physikalisch verfügbare Funkbandbreite begrenzt. Da dieser Wert nicht beliebig vergrößert werden kann, begrenzt er die erlaubte Anzahl der Stationen. Durch diese Berechnung wird demnach die maximale Größe eines Netzwerks bestimmt, die sich bei einer gegebenen Vermittlungskapazität und einer festen Nutzlast der Stationen realisieren lässt.

Schlussfolgerungen

Die oben durchgeführte Rechnung zeigt den engen Zusammenhang von physikalisch verfügbarer Bandbreite und der möglichen Anzahl von Stationen in einem Paketfunknetzwerk. Ein Netzwerk mit beliebig großer Ausdehnung ist unabhängig vom gewählten Routingverfahren schon aus Kapazitätsgründen nicht möglich.

Die Zellularnetzwerke sind ein Beispiel dafür, wie Funknetzwerke mit beliebig großer Ausdehnung konzipiert werden können. Diese Netzwerke kommen aber nie ohne eine feste Infrastruktur aus. Ein Ad-hoc Netzwerksystem, das beliebig erweiterbar sein soll, muss deswegen ab einer bestimmten Größe das Konzept des infrastrukturlosen Betriebes aufgeben.

Für die Entwicklung von Routingalgorithmen ist also weniger die Anzahl der direkt verwaltbaren Stationen relevant für die Skalierbarkeit eines Algorithmus, sondern viel mehr die Fähigkeit zur Zusammenarbeit mit einer Infrastruktur. Aus diesem Grund wird in der vorliegenden Arbeit kein mehrstufiger hierarchischer Algorithmus angestrebt, sondern statt dessen ein Weg aufgezeigt, Ad-hoc Netzwerke mit dem Internet zu verbinden. Ein Zugang zum Internet erlaubt die Kommunikation mit beliebigen Partnern, und über diesen Weg ist auch eine Kommunikation zwischen einzelnen Funknetzwerken möglich.

4.4 Der Simulator

Um verschiedene Charakteristiken der Routingalgorithmen wie Effizienz, Funktionalität, Reaktionszeiten u.ä. leicht testen zu können, wurde im Rahmen dieser Arbeit ein Simulator entwickelt, der Ad-hoc Netzwerke simuliert. In diesem Simulator können Routingverfahren getestet werden, so dass man ohne teure Hardwareimplementierungen, die nur durch komplizierte

Messungen testbar sind, zu aussagekräftigen Resultaten gelangen kann. Dieser Simulator soll im Folgenden genauer erläutert werden.

4.4.1 Architektur

Ein Routingalgorithmus ist nach dem OSI-Modell (siehe Kap. 2.3.1) zwischen der Transportschicht und der Sicherungsschicht angeordnet. Von der Transportschicht werden Pakete an den lokalen Routingalgorithmus übergeben, die über eine oder mehrere Stationen zu übertragen sind und dann am Ziel an die dortige Transportschicht zurückgeben werden. Die unter dem Routingalgorithmus liegende Sicherungsschicht hat die Aufgabe, ein Paket bis zum nächsten Nachbarn oder an eine Gruppe von Nachbarn zu übertragen. Diese Schicht ermöglicht die Kommunikation zwischen den Routingmodulen. Wie in Abbildung 4.3 dargestellt, wird jede simulierte Station durch eine Gruppe, bestehend aus den Modulen Lasterzeugung, Routing und MAC/PHYS repräsentiert. In der Simulation kommuniziert ein Routingalgorithmus so über zwei Schnittstellen mit den beiden angegliederten Schichten.

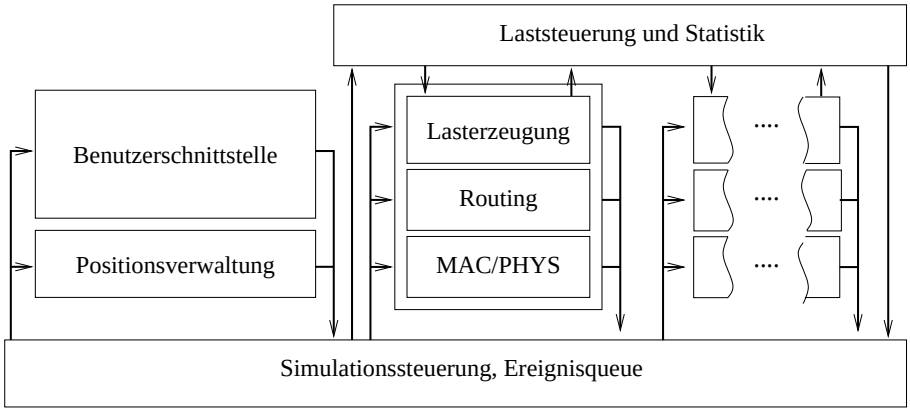


Abbildung 4.3: Architektur des Simulators

Über die Schnittstelle zwischen Lasterzeugung und Routing werden Sendeaufträge von der Transportschicht an die Routingschicht übergeben. Im Simulationsmodell wird das Modul hinter dieser Schnittstelle deswegen als Lastgenerator bezeichnet. Die Aufträge sind Pakete mit Zieladressen. Diese durchlaufen das Routing und die unteren Schichten und werden bei richtiger Verarbeitung dem Lastgenerator zurückgegeben. Der Lastgenerator führt eine Statistik, mit der die erfolgreiche Übertragung von Paketen überprüft wird. Dazu ist ein globales Modul erforderlich, das über alle Aufträge informiert ist und auch deren Abwicklungsdauer erfasst.

Die zweite Schnittstelle entspricht dem Übergang vom Routing zur Sicherungsschicht. An dieser Schnittstelle liefert der Routingalgorithmus Nutz- oder Verwaltungsdaten ab, die an die Nachbarn übertragen werden. In der Simulation verbirgt sich hinter dieser Schnittstelle die gesamte Funktionalität der drahtlosen physikalischen Übertragung.

Die Benutzerschnittstelle zeigt den jeweils aktuellen Zustand der Netzwerksimulation. Mit ihrer

Hilfe lassen sich die Fortschritte der Berechnung verfolgen und dort werden auch Fehler, die im Simulationssystem auftreten, gemeldet.

Die Positionsverwaltung enthält die Adjazenzmatrix, die das Netzwerk beschreibt. Mit diesen Daten werden die Kommunikationsmöglichkeiten zwischen den simulierten Stationen festgelegt.

Die Simulation verwendet eine zeitdiskrete Steuerung. Das zentrale Modul des Simulators ist die Botschaftenverwaltung (auch: Event-Manager), mit dem die Ereignisse verwaltet werden. Die ganze Simulation besteht aus einer Abfolge von Botschaften, die von den einzelnen Modulen erzeugt wird. Wenn ein Modul eine Botschaft vom Event-Manager zugestellt bekommt, dann kann das Modul mit weiteren Botschaften darauf antworten. Jede Botschaft enthält ein Zeitfeld, in dem die gewünschte Auslieferungszeit eingetragen wird. Der Event-Manager sortiert alle anstehenden Botschaften und liefert sie in der richtigen Reihenfolge aus.

Mit dem Simulationssystem wurden sämtliche Vergleiche von Routingalgorithmen erstellt, die in dieser Arbeit vorgestellt werden. Darüber hinaus wurden weitere Untersuchungen, unter anderem auch für die DIRC Netzwerke, mit diesem Simulator durchgeführt [Jan98, FJ97, JF98, JF01, Bor02, Küh00, Tür00].

4.4.2 Netzwerkgenerator

Beim Simulator handelt es sich um ein System zur Untersuchung von Netzwerkprotokollen in mobilen, drahtlosen Rechnernetzen. Das System simuliert dabei eine Anzahl von N Knoten, die sich innerhalb einer rechteckigen Fläche der Größe A beliebig bewegen können. Jeder dieser Knoten hat nur eine begrenzte Sendereichweite r, die bei allen Knoten gleich groß ist. Dies wurde angenommen, damit alle Verbindungen in beide Richtungen funktionieren (Duplex) und keine einseitigen Verbindungen (Simplex) entstehen können. Diese Annahme stellt zwar eine deutliche Vereinfachung der Realität dar, ist aber für ein Netzwerk, in dem alle Knoten mit den gleichen oder ähnlichen Transceivern ausgestattet sind, durchaus realistisch. Außerdem können komplexere Szenarien wie z.B. Rechner in sich bewegenden Autos, Zügen, usw. mittels einiger zusätzlicher Regeln und Parameter leicht auf dieses Szenarium abgebildet werden.

Der Simulator besitzt die Fähigkeit, Übertragungsverluste durch Funk zu erfassen. Eine Untersuchung dieser Verluste ist am Ende dieses Kapitels in der Durchsatzanalyse zu finden. Da manche Algorithmen aber eine fehlerfreie Übertragung einfach voraussetzen, wird in den meisten Simulationen auf das Fehlermodell verzichtet. Die Alternative hierzu ist die zusätzliche Implementierung eines Fehlerschutzprotokolls, die gleichzeitig mit dem Fehlermodell eingesetzt wird, um die geforderte fehlerfreie Übertragung zu erzeugen.

Wenn die Entfernung zwischen zwei Knoten geringer als die Sendereichweite ist, sind beide Knoten in der Lage, voneinander Datenpakete zu empfangen. Allein dadurch, dass sich Knoten im Raum bewegen und sich aus der Reichweite des einen Senders in die Reichweite eines anderen Senders bewegen, gehen also Verbindungen verloren bzw. kommen neue hinzu. Abschattungseffekte, Überlagerungen u.ä. werden nicht berücksichtigt.

Es gibt eine Reihe verschiedener vorstellbarer Szenarien für ein derartiges Netzwerk:

- Alle Knoten bewegen sich mit der gleichen, konstanten Geschwindigkeit zufällig im Raum: In diesem Szenario kann man untersuchen, wie sich das Netzwerkprotokoll in einem Netzwerk mit wenigen, langsamen Veränderungen bei langsamer Fortbewegungsgeschwindigkeit im Gegensatz zu einem Netzwerk verhält, in dem viele, schnelle Veränderungen auftreten und sich die Knoten mit hoher Geschwindigkeit fortbewegen.
- Alle Knoten bewegen sich mit unterschiedlichen, variierenden Geschwindigkeiten zufällig im Raum: Bei dieser Variante kann man untersuchen, wie sich das Netzwerkprotokoll in einer heterogenen Umgebung verhält, in der manchmal viele Veränderungen auftreten und manchmal nur wenige. Man kann also feststellen, wie gut sich ein Algorithmus an sich ändernde Umstände anpassen kann.
- Die Knoten bewegen sich nicht zufällig, sondern mit einem gewissen Ziel im Raum: Realistischer gegenüber den beiden vorangegangenen Szenarien ist die Annahme, dass sich die Knoten nicht wahllos im Raum bewegen, sondern dass sie ein Ziel haben, auf das sie sich zu bewegen, um dort dann für eine gewisse Zeitspanne zu verweilen.

Zwischen diesen Szenarien gibt es natürlich noch unzählige Kombinationsmöglichkeiten. Um eine Vielfalt verschiedener Bewegungsmuster errechnen zu können und dies von der eigentlichen Simulation zu trennen, besteht das Simulationssystem aus zwei Komponenten: Dem Netzdatei-Generator und dem eigentlichen Simulator.

Der Netzdatei-Generator erhält als Eingabe diverse Parameter: Die Anzahl der Knoten, die Sendereichweite sowie die Breite und Länge des Raums, in dem sich die Knoten bewegen. Wählt man für die Netzdatei eine Variante, in der die Knoten sich ein beliebiges Ziel im Raum aussuchen, mit Höchstgeschwindigkeit darauf zulaufen und dort angekommen eine gewisse Zeit verweilen, ehe sie sich ein neues Ziel suchen, so muss zusätzlich noch eine durchschnittliche Pausenzeit angegeben werden, in der die Knoten an ihrem augenblicklichen Ziel verweilen. Eine weitere Möglichkeit ist es, Gruppen unter den Knoten zu bilden, die sich jeweils mit unterschiedlicher Durchschnittsgeschwindigkeiten bewegen. Auf diese Weise erhält man eine Vielzahl von Parametern, die Berechnungen erfordern, die mit der eigentlichen Simulation des Netzprotokolls nichts zu tun haben und deshalb abgetrennt wurden.

Der Netzdatei-Generator erzeugt, wie der Name schon sagt, eine Netzdatei, die dem eigentlichen Simulator als Eingabe dient. In dieser Netzdatei sind, neben der Anzahl der Knoten, alle Verbindungsänderungen des Netzwerks mit dem dazugehörigen diskreten Zeitpunkt abgespeichert, d.h. jeder Zusammenbruch und jedes Neuentstehen einer Verbindung ist hier verzeichnet. Auf diese Weise kann die komplizierte Berechnung der Bewegungen der Knoten und der daraus resultierenden Verbindungen in einem separaten Modul erledigt werden, welches alle Topologieänderungen in chronologischer Reihenfolge an den Simulator weiterliefert.

Ein weiterer Vorteil dieser Methode ist es, dass man unterschiedliche Netzwerkprotokolle bzw. verschiedene Versionen eines Protokolls mit exakt den gleichen Bewegungsmustern testen kann.

Auf diese Weise sind genauere Vergleiche möglich, und außerdem werden redundante Berechnungen vermieden. Die Auswertung des jeweiligen Netzwerkprotokolls zu einer gegebenen Netzdatei liefert der Simulator in einer Ausgabedatei zurück. Zusätzlich zu der Protokollversion und der Netzdatei erhält der Simulator als Eingabe auch noch eine Laufzeitangabe, die der zu simulierenden Zeitspanne entspricht.

Zur besseren Veranschaulichung ist in Abbildung 4.4 das Prinzip des Simulationssystems mit Parametern abgebildet.

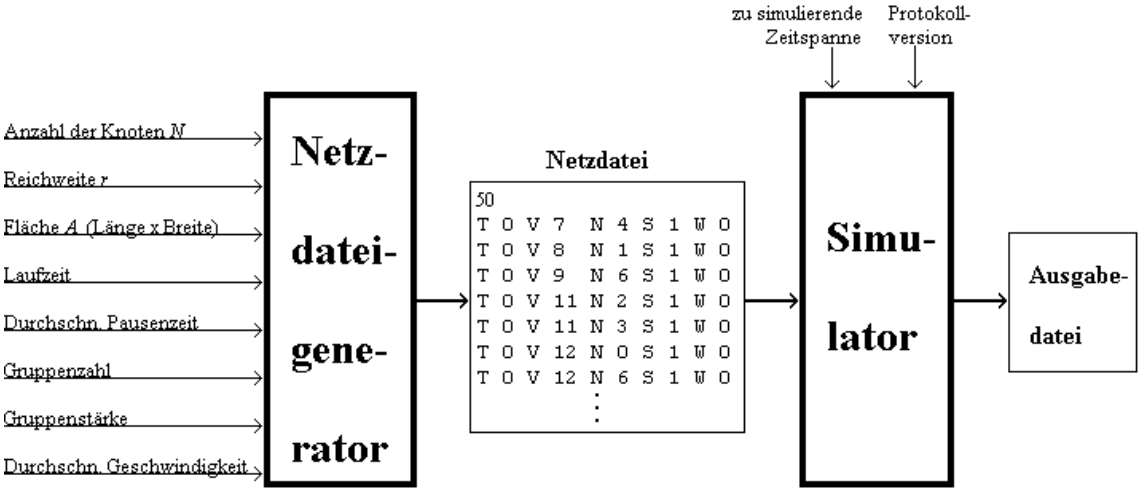


Abbildung 4.4: Funktionsweise des Simulationssystems

4.4.3 Event-Manager

Im Simulator werden die zu simulierenden Ereignisse in ihrer zeitlichen Abfolge durchgerechnet. Ein Ereignis findet dabei zu einem eindeutigen Zeitpunkt statt und kann beliebig viele Folgeereignisse nach sich ziehen, die allerdings erst zu späteren Zeitpunkten stattfinden können. Eine zeitliche Reihenfolge muss also gewährleistet sein. Ein Ereignis kann z.B. das Aussenden eines Erkennungspaketes sein. Folgeereignisse wären dann der Empfang dieses Paketes von verschiedenen Knoten innerhalb der Reichweite des Senders. Da alle Ereignisse zu einem eindeutigen Zeitpunkt stattfinden, spricht man hier auch von zeitdiskreter Simulation.

Innerhalb des Simulators existiert eine eindeutige, globale Uhr, die den Fortlauf der simulierten Zeit misst. Alle Ereignisse werden in einer nach ihrer zeitlichen Reihenfolge sortierten Liste, der Event-Queue, abgespeichert. Der Event-Manager entnimmt nun das erste Ereignis aus der Event-Queue und übergibt es an die zu seiner Verarbeitung zuständigen Programmteile. Diese verändern die Zustandsvariablen und berechnen die entsprechenden Folgeereignisse, die sie wiederum an den Eventmanager zurückgeben, welcher sie korrekt in die Eventqueue einfügt.

Das Ende der Simulation ist erreicht, wenn entweder keine Ereignisse mehr in der Eventqueue vorhanden sind oder wenn eine vorher definierte zu simulierende Laufzeit erreicht oder über-

schritten ist. Dann wird eine Endauswertung aufgerufen, die die Ergebnisse der bereits vorliegenden Zwischenauswertungen zusammenfasst. Da diese Auswertung am Ende einer Simulation stattfindet, kann sie zu diesem Zeitpunkt keine Aussagen mehr über den Verlauf der Simulation machen. Aus diesem Grund ist es sinnvoll, möglichst viele Zwischenabrechnungen durchzuführen, diese Ergebnisse zu speichern, so dass sie in der Endauswertung analysiert werden können. Auf diese Weise kann ein genaues Bild über das Verhalten des Algorithmus über die gesamte Laufzeit erstellt werden.

Das besondere Kennzeichen der zeitdiskreten Simulation ist die von Ereignis zu Ereignis weiterspringende Uhr. Da die Zustandsvariablen nur durch Ereignisse verändert werden können, die zu diskreten Zeitpunkten stattfinden, ist es nicht nötig, die Uhr kontinuierlich weiterzustellen. Sie kann deshalb nach der Abarbeitung eines Ereignisses direkt zum nächsten Ereignis weitergestellt werden. (Dies ist auch der Grund, warum sich die Verwaltung der globalen Uhr im Event-Manager befindet.)

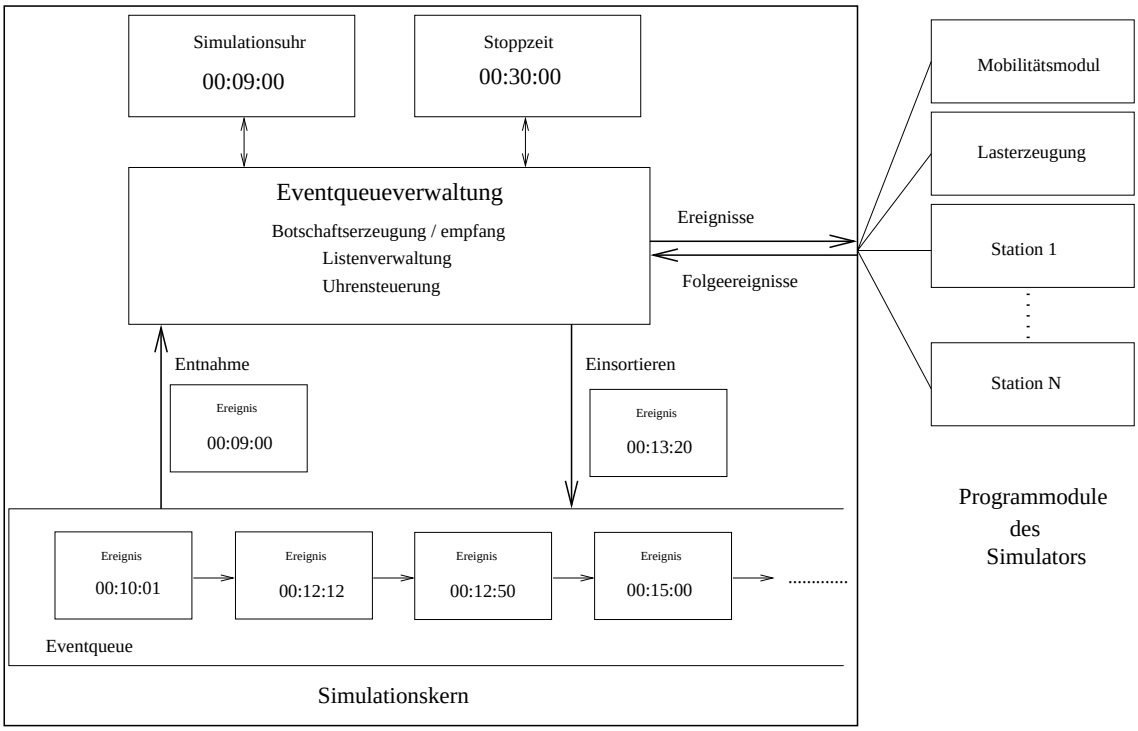


Abbildung 4.5: Schematische Darstellung des Event-Managers

Abbildung 4.5 zeigt eine schematische Darstellung des Eventmanagers. Eine zentrale Komponente ist die Verwaltung der Eventqueue. Aus der Eventqueue werden Ereignisse entnommen und an die entsprechenden Programmmodule weitergegeben. Empfangene Folgeereignisse werden wieder korrekt in die Event-Queue einsortiert. Dann wird die Uhr auf den Zeitpunkt des nächsten Ereignisses vorgestellt, dieses aus der Eventqueue entnommen usw., bis entweder keine Ereignisse mehr vorhanden sind oder die Stoppzeit erreicht wurde.

4.4.4 Statistische Auswertung

Die Auswertung ist ein Modul, das die Funktionsfähigkeit bzw. Korrektheit von Routen kontrolliert. Dabei überprüft sie nicht, ob die in der Routingtabelle eines einzelnen Knotens gespeicherten Informationen konsistent sind, sondern sie versucht, den Weg, den ein abgeschicktes Datenpaket vom Sender bis zum Empfänger durch das Netzwerk nehmen würde, nachzuvollziehen. Die einzelnen Tests werden periodisch mit einer gewissen Frequenz (Default: 100 ms) für eine einstellbare Anzahl von Routen (Default: 100) wiederholt.

Untersucht man ein relativ kleines Netzwerk, d.h. bis zu 100 Knoten, so kann es durchaus Sinn machen, sämtliche Routen des Netzwerks zu überprüfen. Dies erfordert bei N Knoten zwar einen Rechenaufwand von genau $\frac{N(N-1)}{2}$ Routen, aber da einstellbar ist, mit welcher Frequenz die Auswertung ihre Tests durchführt, kann man den Berechnungsaufwand damit auf das gewünschte Maß reduzieren.

Wenn jedoch häufige Tests durchgeführt werden sollen, um das Verhalten des Netzwerks in kurzen Abständen zu untersuchen und damit ein genaueres Bild über die gesamte Laufzeit zu erhalten, so ist das Überprüfen sämtlicher Routen des Netzwerks im Allgemeinen zu rechenintensiv. Um dies zu vermeiden, sollte die Auswertung so eingestellt werden, dass sie nicht mehr sämtliche Routen überprüft, sondern nur eine bestimmte Anzahl von Routen. Die einzelnen Routen werden dann mittels eines einen Zufallszahlengenerators ausgewählt und das Netzwerk somit stichprobenartig untersucht.

Das Überprüfen der einzelnen Routen geht folgendermaßen vonstatten: Um die Route von Knoten A nach Knoten B auf ihre Korrektheit zu testen, versucht die Auswertung mit den in den Routingtabellen der Knoten des Netzwerks verteilt gespeicherten Informationen von Knoten A nach Knoten B zu routen. Dabei beginnt sie mit Knoten A und schaut in dessen Routingtabelle nach, über welchen Knoten C er ein Paket an Knoten B schicken würde. Als nächstes wird überprüft, ob diese Verbindung von A nach C überhaupt existiert, d.h. die Auswertung schaut in der Adjazenzmatrix des Netzwerks nach, ob die Verbindung existiert. In dieser Adjazenzmatrix wird vom Modul „Node“ der momentane Status sämtlicher Verbindungen des Netzwerks abgespeichert. Die Knoten selbst haben jedoch keinen Zugriff auf diese Matrix, die ein Teil des Simulators ist, d.h. sie müssen das Vorhandensein von Verbindungen durch das Aussenden bzw. Empfangen von Erkennungspaketen bemerken. Ist in der Adjazenzmatrix eine funktionierende Verbindung von A nach C eingetragen, so fährt die Auswertung mit Knoten C fort. Sie schaut in der Routingtabelle von C nach, über welchen Knoten er ein Datenpaket für Knoten B weiterschicken würde. Dieser Vorgang wird solange fortgesetzt, bis man am Knoten B angelangt ist, d.h. bis die Route vollständig abgelaufen wurde oder bis ein Fehler aufgetreten ist. Folgende Fälle können dabei auftreten:

- Korrekte Route: Im günstigsten Fall sind in allen Routingtabellen der Knoten auf der Route von A nach B gültige Nachfolger eingetragen, über die man auch tatsächlich zum Knoten B gelangt. In diesem Fall funktioniert die Route und wird als korrekt gewertet.
- Defekte Route: Es kann jedoch auch folgende Situation auftreten: Auf der Route von A

nach B gelangt man zum Knoten X. In der Routingtabelle des Knoten X ist Knoten Y als derjenige Rechner eingetragen, über den ein Datenpaket an B weitergeschickt würde. Die Adjazenzmatrix weist jedoch keine Verbindung zwischen X und Y auf, d.h. X hat den Zusammenbruch der Verbindung zu Knoten Y noch gar nicht realisiert und kann das Datenpaket somit auch nicht weiterschicken. Die Routingtabelle von X enthält also ungültige Informationen, wodurch die Route von A nach B über X nicht zum Ziel führt. Die Route wird als defekt gewertet.

– Unerreichbar: Ein weiterer Fall ist, dass ein Knoten (A oder irgendein anderer Knoten auf der Route) keine Route zum Knoten B ermitteln konnte. In seiner Routingtabelle ist dann eingetragen, dass Knoten B im Moment nicht erreichbar ist, d.h. die Route wird mit „unerreichbar“ bewertet.

– Schleife: In sich dynamisch verändernden Netzwerken kann es vorkommen, dass auf einer Route eine Schleife entsteht. D.h. wenn man die Route eines Datenpaketes von A nach B verfolgt, gelangt man irgendwann ein zweites Mal zu einem bestimmten Knoten der Route. Das Paket wird also im Kreis geschickt. Um dieses Phänomen zu bemerken, zählt die Auswertung, wie viele Knoten beim Überprüfen der Route bereits besucht wurden. Wenn diese Zahl größer als die Anzahl der Knoten im Netzwerk ist, bedeutet dies, dass mindestens ein Knoten schon zweimal besucht wurde. Es liegt also eine Schleife vor.

Die Auswertung überprüft also in periodischen Abständen eine gewisse Anzahl bzw. alle Routen und summiert die Zahlen der korrekten Routen, der defekten Routen, der Schleifen und der Fälle, in denen ein Zielknoten unerreichbar ist. Diese Werte werden am Ende der Simulation in der Ausgabedatei ausgegeben.

4.5 Beispiel einer MAC-Simulation

Am Anfang wurden mit dem Simulator Versuche unternommen, um den Durchsatz von Medium Access Control (MAC) Protokollen zu bestimmen. Für die Simulation wurden folgende Parameter eingesetzt:

– Knotenzahl: 25

– Vermaschung: 100 Prozent.

– Übertragungskapazität: Die physikalische verfügbare Netzkapazität ist 1 Mbit pro Sekunde.

– Störeigenschaften: Alle Pakete einer Kollision gehen verloren.

– Kollisionsvermeidungsstrategie: Es wird MACA [Kar90] oder CSMA eingesetzt [Tan96].

– Warteschlangen: Die Simulation braucht keine Warteschlangen, ein Paket wird immer dann neu bereitgestellt, wenn das letzte Paket übertragen wurde.

– Datenmenge: Die verfügbare Datenmenge ist unendlich groß, das Ergebnis der Simulation ist die Anzahl der tatsächlich übertragenen Pakete.

– Paketgröße: Die Pakete umfassen 256,512,1024 oder 2048 Bytes.

– Datenrate: Die Stationen versuchen Pakete mit einer Rate von 0 bis 300 Prozent der Netzkapazität zu senden.

– Fehlerkorrektur: keine.

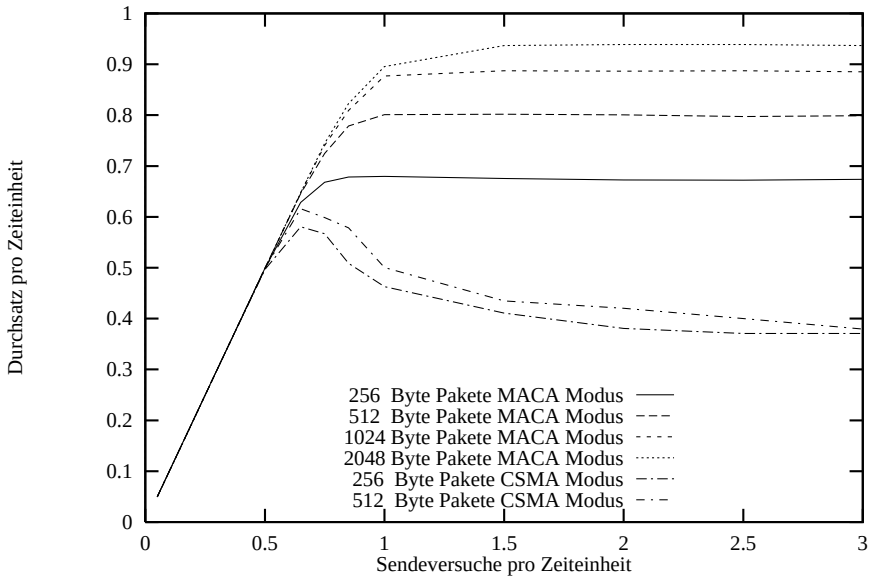


Abbildung 4.6: Durchsatz bei unterschiedlichen Paketgrößen und MAC-Verfahren

Als Ergebnis wird das Verhältnis der erfolgreich übertragenen zu den insgesamt versendeten Pakete gewertet. Die Abbildung 4.6 zeigt das Ergebnis von 6 Simulationsserien. Davon verwenden vier Serien MACA [Kar90] und zwei Serien CSMA [Tan96] als MAC-Verfahren. Als bestes Verfahren erreicht MACA einen Durchsatz von 94 Prozent fast unabhängig von der Netzbelastung. CSMA liegt weit darunter; dies ist dadurch zu erklären, dass MACA immer eine Reihenfolge für die Sendungen unter den Stationen aushandelt, während CSMA Kollisionen nur mit einer bestimmten Wahrscheinlichkeit ausweicht. Deswegen treten in CSMA mit zunehmender Netzlast immer häufiger Kollisionen auf.

4.6 Zusammenfassung

Dieses Kapitel widmet sich den Modellierungs-, Simulation- und Bewertungsmethoden für paketorientierte Routingalgorithmen, die in Ad-hoc Netzwerken eingesetzt werden.

Für diese Netzwerke existiert eine große Zahl von Vorschlägen zur Lösung des Routingproblems, die schon im Kapitel 3.3 vorgestellt wurden. Ebenso unfangreich ist die Palette der Testkriterien, nach denen die Algorithmen auf ihrer Leistungsfähigkeit hin untersucht werden können.

Analytische Methoden können herangezogen werden, um gezielt einzelne Eigenschaften eines Algorithmus zu ermitteln. Eine solche Analyse ist aber entweder oft undurchführbar oder zumindest mathematisch sehr komplex, wenn alle Umgebungsbedingungen von Ad-hoc Netzwerken zu berücksichtigen sind. Für eine möglichst umfassende Bewertung ist daher die Simulation der Algorithmen erforderlich. Eine Simulation verlangt zwar die vollständige Implementierung eines Algorithmus, sie ermöglicht dafür im Anschluss eine detaillierte Untersuchung des Verhaltens, die sich mit keiner anderen Technik erreichen lässt. Selbst ein Test im Feldversuch kann dies nicht bieten, da Messungen im Feld wegen der zufällig auftretenden Übertragungsfehler nicht wiederholbar sind.

Die Simulation von Ad-hoc Netzwerken erfordert einen speziellen Simulator, der für die hier durchgeführten Untersuchungen eigens entwickelt wurde. Der Simulator setzt sich aus einer Reihe von Modulen zusammen, deren Funktionalität in diesem Kapitel vorgestellt wurde. Die Besonderheit des Simulators sind die schnell veränderlichen Netzwerke, die durch eine vorgeschaltete Bewegungssimulation erzeugt werden. Dabei wird im ersten Schritt eine Netzwerkbeschreibung aus den Bewegungen einer Gruppe von Stationen erzeugt. Diese Beschreibung verwendet der nachgeschaltete Simulator des Routingprotokolls um die Leistungen von Routingalgorithmen zu testen. Die dazu eingesetzte Auswertung wurde ebenfalls vorgestellt.

Das Kapitel endet mit der Simulation eines Funksystems, in der der Durchsatz verschiedener MAC-Protokolle ermittelt wird.

Kapitel 5

Routing durch Pfadsuche

5.1 Einleitung

Das folgende Kapitel stellt einen verbesserten Routingalgorithmus vor, der besonders für den Einsatz in Funknetzwerken mit beschränkter Bandbreite und schnellen Topologieveränderungen geeignet ist. Der Algorithmus ist eine Weiterentwicklung der bereits in Kapitel 3 vorgestellten Pfadsuche. Um die Funktionsprinzipien des Algorithmus zu verdeutlichen, beginnt dieses Kapitel mit einer Beschreibung der bisher verwendeten Pfadsuche und zeigt deren Stärken und Schwächen auf.

Anschließend wird der neue Algorithmus vorgestellt, der ausschließlich über den Austausch von Spannbäumen arbeitet und daher „Tree Exchange Routing Algorithm“ (TERA) genannt wird. Das Funktionsprinzip von TERA wird dabei anhand von Beispielen erklärt. Der neue Algorithmus ist nach der Algorithmen-Klassifizierung, die in Abschnitt 3.3 beschrieben wurde, in Klasse 2 einzuordnen. Deswegen bietet TERA einen Mittelweg zwischen den Distanz-Vektor und Link-State Algorithmen und vereint die verteilte Routenberechnung der DV-Algorithmen mit der effektiven Schleifenvermeidung und Erweiterbarkeit von Link-State Algorithmen. Das Kapitel endet mit einer Simulation, die einen Vergleich zwischen der bisher verwendeten Pfadsuche und dem verbesserten Algorithmus bietet. In der Simulation wird die Anzahl der funktionsfähigen Routen bei langsamer und sehr schneller Topologieänderung untersucht. Die Ergebnisse zeigen, dass TERA besonders bei schnellen Topologieänderungen mehr funktionsfähige Routen erzeugt.

Der erweiterte Routingalgorithmus wurde bereits in [JF01] veröffentlicht. Die Simulationen entstanden im Rahmen einer Diplomarbeit und wurden in [Küh00] vorgestellt.

5.2 Einfache Pfadsuche

Schon in der Einführung der Basisalgorithmen wurde gezeigt, dass der Distanz-Vektor Algorithmus wegen des count-to-infinity Problems den schnellen Topologieänderungen nicht folgen kann und daher nicht für Ad-hoc Netzwerke geeignet ist. Der Link-State Algorithmus hat ebenfalls Schwierigkeiten mit Ad-hoc Netzwerken, er kann zwar den Topologieänderungen folgen, aber dazu sind regelmäßig geflutete Botschaften nötig, die ein Ad-hoc Netzwerk mit begrenzter Bandbreite zu sehr belasten.

In der Einführung wurde eine brauchbare Lösung für Ad-hoc Netze aufgezeigt, die Pfadsuche. Bei der Pfadsuche wird ein Distanz-Vektor Algorithmus so erweitert, dass die benutzten Pfade zu jedem Ziel vollständig aus der Tabelle ablesbar sind. Durch die Rückverfolgung der Pfade können Schleifen erkannt und verhindert werden, damit ist das count-to-infinity Problem gelöst. Diesen Ansatz verfolgt beispielsweise das in Abschnitt 3.3.9 vorgestellte WRP-Routingprotokoll.

Für die Algorithmusbeschreibung wird ein Netzwerk aus Stationen verwendet, wobei jede Station einen eindeutigen Bezeichner k erhält. Die Stationen können miteinander über Verbindungen kommunizieren, jeder Verbindung ist ein Kostenwert zugeordnet, der auch als Distanz bezeichnet wird. Die Distanz zwischen einer Ausgangsstation und einer Zielstation wird durch den Wert d_{ij} angegeben und muss positiv sein. Wenn es keine direkte Verbindung zwischen den Knoten i, j gibt wird die Distanz auf unendlich gesetzt: $d_{ij} = \infty$. Knoten mit $d_{ij} < \infty$ sind Nachbarn. Die Distanz eines Knotens zu sich selbst ist immer Null: $d_{ii} = 0$.

5.2.1 Datenstrukturen

Jede Station besitzt einen eindeutigen Bezeichner k und der Zustand einer Station wird durch Tabellen beschrieben:

- Die Routingtabelle D_i^k, die für jedes Ziel i die bestmögliche Distanz angibt.
- Die Nachfolgertabelle N_i^k beschreibt, über welchen Nachbarn das Ziel zu erreichen ist.
- Die Vorgängertabelle V_i^k, die den jeweils vorletzten Knoten im Pfad angibt.

In zwei weiteren Tabellen sammelt jeder Knoten alle von den Nachbarn erhaltenen Informationen. Jeder Knoten besitzt damit die vollständigen Informationen über die Routingtabellen seiner Nachbarn und errechnet daraus seine eigenen Routingtabellen.

- In $ND_{i\ j}^k$ wird die beste Distanz zu einem Ziel i angegeben, die der Knoten j errechnet und übermittelt hat.

– Zu der Tabelle der besten Distanzen muss auch eine Tabelle der entsprechenden vorletzten Knoten im Pfad gespeichert werden. In $NV_{i\,j}^{k}$ wird zu jedem Ziel der vorletzte Knoten angegeben, den der Nachbar j übermittelt hat.

Die Tabelle N_i ist redundant, da sie sich jeweils durch Zurückverfolgung der Pfade aus V_i errechnen lässt. Deswegen braucht dieser Tabelleninhalt auch nicht an die Nachbarn übermittelt zu werden.

5.2.2 Initialisierung

Die Kommunikation zwischen den Knoten geschieht durch Botschaften, die den Zustand eines Knotens verändern. Der Zustand wird durch den Inhalt der Tabellen $D_i^k(t), N_i^k(t), V_i^k(t), ND_{i\,j}^k(t)$ and $NV_{i\,j}^k(t)$ festgelegt. Zu Anfang, bei $t = 0$, initialisiert jeder Knoten seine Tabellen:

– $D_i^k(0) := \infty \; \forall \; i,k \;\; i \neq k$,
– $D_k^k := 0 \; \forall \; k$
– $N_i^k(0) :=$ *"none"* $\forall \; i,k$
– $V_i^k(0) :=$ *"none"* $\forall \; i,k$
– $ND_{i\,j}^k(0) := \infty \; \forall \; i,j,k$
– $NV_{i\,j}^k(0) :=$ *"none"* $\forall \; i,j,k$

5.2.3 Botschaften an Nachbarn

Nachdem alle Knoten ihre Tabelle initialisiert haben, beginnt die Iteration. Im ersten Schritt werden die Tabellen D_i^k, V_i^k (die anfangs nur ein Element enthalten) an alle Nachbarn gesendet. Jeder Knoten verarbeitet die erhaltenen Informationen und sendet im zweiten Schritt seine neu berechneten Tabellen wieder an seine Nachbarn. Auf diese Weise breitet sich die Information mit jeder Iteration um einen Hop weiter aus.

Jeder Knoten speichert die empfangene Information in seinen Tabellen $ND_{i\,j}^k$ und $NV_{i\,j}^k$. Wenn Knoten k die Tabelleneinträge $D_i^j(t-1)$ und $V_i^j(t-1)$ für die Ziele i von Nachbar j empfängt, speichert er die darin enthaltenen Daten für die anschließende Routenberechnung:

– Zu allen von Nachbar j angegebenen Distanzen muss noch die Distanz von k zu Nachbar j hinzuaddiert werden: $ND_{i\,j}^k(t) := D_i^j(t-1) + d_{k\,j} \; \forall \; i$.
– Die Vorgängertabelle bleibt unverändert $NV_{i\,j}^k(t) := V_i^j(t-1) \; \forall \; i$.

Eventuell muss noch der Eintrag in der Vorgängertabelle, der sich auf die Kante von Knoten k zu Knoten j bezieht, korrigiert werden, denn die von j gesendete Tabelle besitzt noch keinen Vorgänger für j. Hier wird Knoten k als Vorgänger eingetragen.

- $NV^k_{j\,j}(t) := k$

Geht eine Verbindung zu einem Nachbarn j verloren, dann werden die Einträge für diesen Nachbarn gelöscht:

- $ND^k_{i\,j}(t) := \infty \; \forall \; i$
- $NV^k_{i\,j}(t) := \text{"none"} \; \forall \; i$

Ebenso entfallen alle Ziele i, die über j zu erreichen waren. Die Ziele sind eventuell über andere Nachbarn erreichbar, deswegen werden die Ziele zuerst aus den Tabellen entfernt, danach muss aber für jedes entfernte Ziel eine neue Routenberechnung durchgeführt werden.

- $D^k_i := \infty \; \forall \; i \mid N^k_i = j$
- $V^k_i := \text{"none"} \; \forall \; i \mid N^k_i = j$
- Routenberechnung $\forall \; i \mid N^k_i = j$

Für jeden von j gesendeten Tabelleneintrag $D^j_i(t-1), V^j_i(t-1)$ berechnet Knoten k für das Ziel i aus den nun vorliegenden Tabellen $ND^k_{i\,j}(t)$ und $NV^k_{i\,j}(t)$ seine neue Routingtabelle und speichert sie in den Tabellen $D^k_i(t)$, $N^k_i(t)$ und $V^k_i(t)$.

Unterscheiden sich die neu errechneten Tabelleneinträge von denen der letzten Iteration $D^k_i(t-1)$, $N^k_i(t-1)$ und $V^k_i(t-1)$, dann müssen alle veränderten Einträge an die Nachbarn gesendet werden.

5.2.4 Pfadberechnung und Prüfung

Die Berechnung wird bei der einfachen Pfadsuche, wie beim Distanz-Vektor Algorithmus, für jedes Ziel i einzeln durchgeführt. Da die gesamte Berechnung zu einem Zeitpunkt t stattfindet, werden in der folgenden Beschreibung die Indizes zur Zeitangabe t weggelassen.

Vor der Berechnung muss die alte Route aus der Tabelle gelöscht werden:

- $D^k_i := \infty$
- $V^k_i := \text{"none"}$

- $N_i^k := \text{"none"}$

Danach sucht die Station k aus den von den Nachbarn j angebotenen Distanzen $ND_{i\,j}^k$ das günstigste zum Ziel i heraus und speichert den Nachbarn mit dem besten Angebot in b:

$$b = j \mid ND_{i\,j}^k = Min\{ND_{i\,j}^k, \forall j\}$$

Im Distanz Vektor Algorithmus wird der so ermittelte Nachbar als Nachfolger zu Ziel i benutzt. Bei der Pfadsuche muss aber die Schleifenfreiheit gesichert sein, deswegen ist vor der Eintragung noch eine Prüfung notwendig. Die Prüfung verfolgt den Pfad von Ziel i zum Ausgangsknoten b zurück. Nur wenn die Prüfung erfolgreich ist, darf der Pfad eingetragen werden, andernfalls wird das Ziel als unerreichbar gekennzeichnet.

Zur Überprüfung des Pfades wird der Pfad, bestehend aus den Knoten n_1 bis n_l, abgesucht. Der Wert l ist anfangs noch unbekannt, er darf aber nie größer als $|N|$ werden, denn ein Pfad mit $l > |N|$ deutet auf eine Schleife hin, da mindestens ein Knoten doppelt enthalten sein muss. Ebenso darf Knoten k nicht im Pfad enthalten sein, dies würde ebenfalls zu einer Schleife führen. Der Pfad wird mit einer Rekursion ermittelt:

- $n_1 = i$
- $n_2 = NV_{i\,b}^k$
- $n_{l+1} = V_{n_l}^k$

Die Rekursion wird abgebrochen, sobald $n_l = b$ oder $n_l = k$ erreicht ist oder $l > |N|$ wird. Der untersuchte Pfad ist nutzbar, wenn $n_l = b$ ist. Dann wird der Knoten b als Nachfolger zu i gesetzt:

- $D_i^k := ND_{i\,b}^k$
- $N_i^k := b$
- $V_i^k := NV_{i\,b}^k$

Die Funktionsweise der Schleifensperre wird deutlich, wenn das bereits in Abbildung 3.4 und Tabelle 3.3 vorgestellte Beispiel nochmals mit der Pfadsuche durchgerechnet wird.

Tabelle 5.1 zeigt für das ganze Netz den Inhalt der Distanztabellen und die Angabe der Vorgänger zur Station A. Aus der Tabelle ist zu erkennen, dass sofort nach dem Verbindungsverlust zu A die Route schnellstmöglich abgebaut wird. Die Stationen können aus ihren Tabellen die gesamte Route zu A nachvollziehen, daher versuchen sie nicht über ihren zweiten Nachbarn eine Ersatzroute aufzubauen. Um das Vorgehen näher zu erläutern, werden für Station C die einzelnen Berechnungsschritte bis zum Löschen der Route vorgestellt.

D_A^A	V_A^A	D_A^B	V_A^B	D_A^C	V_A^C	D_A^D	V_A^D	D_A^E	V_A^E	
0	-	1	A	2	B	3	C	4	D	Startwert
		∞	-	2	B	3	C	4	D	Nach einer Iteration
		∞	-	∞	-	3	C	4	D	Nach zwei Iterationen
		∞	-	∞	-	∞	-	4	D	Nach drei Iterationen
		∞	-	∞	-	∞	-	∞	-	Nach vier Iterationen

Tabelle 5.1: Routenabbau bei der Pfadsuche

D_A^C	V_A^C	N_A^C	V_B^C	$ND_{A\,B}^C$	$NV_{A\,B}^C$	$ND_{A\,D}^C$	$NV_{A\,D}^C$	
2	C	B	C	2	B	4	C	Startwert
2	C	B	C	2	B	4	C	Nach einer Iteration
2	C	B	C	∞	-	3	C	Schritt 1
∞	-	-	C	∞	-	3	C	Schritt 2
∞	-	-	C	∞	-	3	C	Nach zwei Iterationen

Tabelle 5.2: Routenabbau in der Station C

Die Tabelle 5.2 zeigt die Tabelleninhalte der Station *C*. In der ersten Zeile ist der Zustand mit einer funktionierenden Verbindung zu Station *A* dargestellt. Die Station *B* ist dabei als direkter Nachfolger für das Ziel *A* eingetragen. Die zweite Zeile zeigt den Tabellenzustand nach dem Verbindungsverlust zwischen Station *A* und *B*. Station *B* meldet nach der ersten Iteration eine Distanz von unendlich und keinen Nachfolger für *A*.

Die Station *C* speichert im ersten Schritt die Botschaft von *B* zuerst in $ND_{A\,B}^C$ und $NV_{A\,B}^C$ und startet die Routenberechnung. In der Berechnung werden zuerst die Einträge D_A^C, V_A^C, N_A^C gelöscht, dies ist in Schritt zwei dargestellt.

Zur Neuberechnung der Route in Station *C* wird nun nach dem minimalen Distanzwert in $ND_{A\,j}^C$ gesucht. Da nur noch ein Eintrag von *D* vorhanden ist, wird dieser verwendet. Diese Route kann aber nicht eingetragen werden, da eine Rückverfolgung des Pfades nicht gelingt. Die Rückverfolgung scheitert bereits beim ersten Hop, weil Station *C* zuerst den Eintrag $NV_{A\,D}^C$ ausliest und dabei *C* erhält. Damit ist die Abbruchbedingung bereits erfüllt und das Ziel wird als unerreichbar gekennzeichnet.

5.2.5 Kritische Betrachtung

Die hier vorgestellte einfache Pfadsuche ist ein verteilter Algorithmus, der die Routenberechnung lokal durchführt. Jede Station sendet nur die unbedingt nötigen Informationen an ihre Nachbarn. Daher ist der Algorithmus in seinem Bandbreitenbedarf verhältnismäßig effizient. Die Botschaften müssen allerdings zuverlässig zwischen den Nachbarn ausgetauscht werden. Das ist entweder durch ein verlässliches Übertragungsprotokoll zu erreichen, oder der Algorithmus muss so erweitert werden, dass jede Station seine Tabellen mit der Nachbarinformation $ND_{i\,j}^k$ und $NV_{i\,j}^k$ regelmäßig mit den Nachbarn abgleicht.

Der Speicherbedarf des Algorithmus ist durch das Vorhalten der Nachbarinformation relativ hoch. Ein Verzicht auf diese Tabellen ist möglich, jedoch nur auf Kosten der Bandbreitennutzung. Wenn auf die Tabellen $ND^k_{i\ j}$ und $NV^k_{i\ j}$ verzichtet wird, dann ist bei einem Verbindungsverlust zu einem Nachbarn keine sofortige Neuberechnung der verlorenen Routen möglich, die fehlende Information muss dann von sämtlichen Nachbarn neu eingeholt werden.

Der Bedarf an Rechenleistung ist mit dem des DV-Algorithmus vergleichbar, jede Routenberechnung besteht nur aus dem Ermitteln eines Minimums und der anschließenden Pfadüberprüfung. Gerade die Routenberechnung einzelner Ziele stellt aber auch das Hauptproblem der einfachen Pfadsuche dar. Die Ergebnisse des Algorithmus sind in einigen Fällen von der Reihenfolge der empfangenen Botschaften abhängig. Die eingesetzte Pfadüberprüfung verlangt immer einen gültigen Pfad, bevor eine Route eingetragen werden kann. Sind die Botschaften zur Beschreibung zweier neuer Stationen so vertauscht, dass zuerst die weiter entfernte Station bearbeitet wird, dann kann diese Station nicht eingetragen werden, wenn der Pfad über die näher gelegene Station noch nicht existiert. Die Auswirkungen dieses Problems werden später durch eine Simulation in Abschnitt 5.4 genauer untersucht.

5.3 Pfadsuche mit TERA

5.3.1 Funktionsprinzip

Im vorangehenden Abschnitt wurde die Pfadsuche als Erweiterung des DV-Algorithmus vorgestellt. Der Algorithmus ist in dieser Form bereits für Ad-hoc Netzwerke verwendbar, allerdings sind die Berechnungsvorgänge zwischen den Stationen oft nur schwer nachvollziehbar, da der Routenaufbau durch mehrmaliges Austauschen einzelner Elemente einer Routingtabelle geschieht. Aus diesem Grund wird im neuen Ansatz ausschließlich mit Spannbäumen gearbeitet. Der Informationsaustausch zwischen den Stationen ist ein Austausch von kompletten Bäumen inklusive der Distanzinformationen. Zur besseren Unterscheidung gegenüber der bereits vorgestellten Pfadsuche wird der neue Algorithmus daher „Tree Exchange Routing Algorithm“ (TERA) genannt.

Durch die ausschließliche Betrachtung der Spannbäume vereinfacht sich das Verständnis für die Vorgänge in den Stationen wesentlich. Der Bandbreitenbedarf des Algorithmus steigt in Praxis dennoch nicht wesentlich an, wenn nicht jedes Mal ein kompletter Baum übertragen wird, sondern nur die Differenzinformation zum vorhergehenden Baum. Dann beschränkt sich die Übertragung auf die neuen Tabelleneinträge und ist damit von Aufwand her identisch zu dem bereits vorgestellten Pfadsuchealgorithmus. Die Arbeitsweise von TERA wird nun durch ein einfaches Beispiel mit vier Stationen exemplarisch beschrieben.

Abbildung 5.1 zeigt oben das zugrunde liegende Netzwerk und die daraus resultierenden Bäume. Jede Station muss in einem Ad-hoc Netzwerk ihre Nachbarn kennen. Im Beispielnetzwerk hat jede Station zwei Nachbarn. Die Nachbarschaftsbeziehungen werden für jede Station als Baum mit der Tiefe 1 dargestellt. Die Wurzel jedes Baumes ist die Station selbst, die Blätter repräsen-

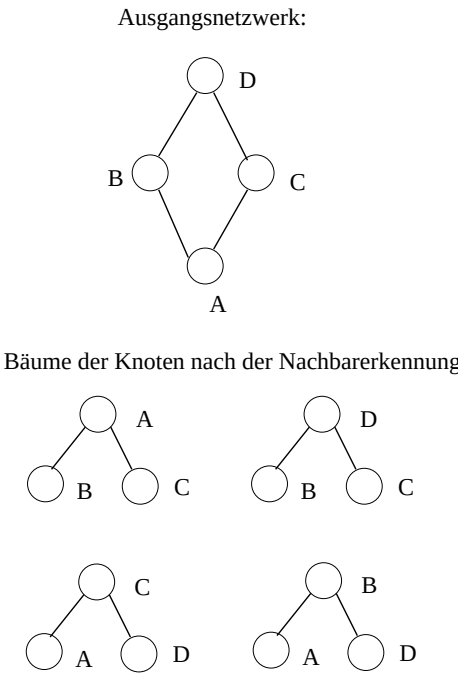


Abbildung 5.1: Beispielnetz und daraus resultierende Bäume

tieren die vorhandenen Nachbarn.

Abbildung 5.2 beschreibt die Berechnung der Routen für die Station *A*. In TERA werden immer die kompletten Bäume unter den Nachbarn ausgetauscht, so erhält Station *A* die Bäume von *B* und *C*. Diese Bäume werden von, wie in der Abbildung durch die beiden dicken Pfeile dargestellt, an den eigenen Baum mit der Tiefe 1 angehängt. Der so entstehende konkatenierte Graph ist im Bild unten links dargestellt. Die Station *A* ist in diesem Graphen drei mal vorhanden. Zur Berechnung der Routen wird dann der Algorithmus von Dijkstra verwendet, der die kürzesten Pfade von allen Stationen zu *A* berechnet. Das Ergebnis ist im Bild unten rechts dargestellt. Die Berechnung durch Dijkstra garantiert für das Ergebnis Schleifenfreiheit und minimale Pfadlängen. Der neu berechnete Baum muss anschließend von der Station *A* an die Nachbarn gesendet werden. Auf diese Weise verbreitet sich, wie beim DV-Algorithmus, die Information über neue oder verschwundene Stationen im ganzen Netzwerk.

5.3.2 Austausch von Bäumen

In TERA werden zwischen den Stationen nur vollständige Bäume ausgetauscht. Das Senden aller vollständigen Bäume verbraucht aber sehr viel Kapazität des Funknetzwerks. Um die Kommunikation gering zu halten, ist es sinnvoll, bei Veränderungen lediglich die neuen Tabellenteile an die Nachbarn zu senden. Das Einbauen dieser Informationen ist dann jedem Knoten leicht möglich.

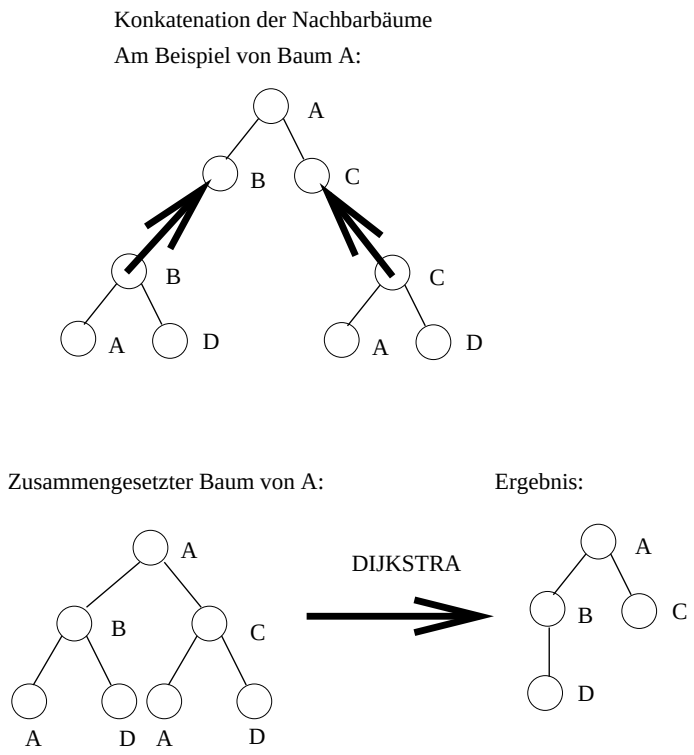


Abbildung 5.2: Routenberechnung durch Konkatenation von Bäumen

Aufgrund der unzuverlässigen Funkübertragung entstehen mitunter Situationen, in denen einige Tabellenteile die Nachbarn erreichen, andere jedoch verloren gehen. Akzeptiert ein Nachbar die nur teilweise eintreffenden Tabellenteile und baut er diese in seine Tabelle ein können unvollständige Bäume, Schleifen und widersprüchliche Tabelleninhalte entstehen. Das Problem lässt sich lösen, indem jede Botschaft an einen Nachbarn eine eindeutige Seriennummer erhält, die es diesem erlaubt, die Vollständigkeit der eintreffenden Botschaften zu kontrollieren. Fehlende Teile müssen dann entsprechend nachgefordert werden. Dies benötigt weiteren Speicherplatz, denn ein Sender muss die letzten Botschaften für den Fall einer Wiederholungsanfrage zwischenspeichern.

Eine elegantere Lösung ist der Einsatz von sogenannten Hash-Werten. Ein Hash-Algorithmus ermittelt eine Art Quersumme über einen Datensatz, der in diesem Fall dem zu versendenden Baum entspricht. Wird der Hashwert eines neuen Baumes an die versendeten Tabellenteile angehängt, dann kann ein Empfänger sofort kontrollieren, ob der vorliegende Baum nach Einsetzen der neuen Tabellenteile vollständig identisch mit dem des Senders ist. Ist dies nicht der Fall, dann wird der ganze Baum nachgefordert. Dies erfordert zwar etwas mehr Bandbreite, garantiert aber dafür zwischen den Nachbarn die vollständige Konsistenz der Bäume.

Eine weitere Verbesserungsmöglichkeit betrifft die Nachforderung ganzer Bäume. Große Bäume können dabei vor dem Senden in Teilbäume zerlegt werden, wobei für die Teilbäume eigene Hashwerte berechnet werden. Die Empfangsstation bekommt zuerst nur eine Liste der Hashwerte und fordert dann gezielt die Teilbäume nach, die nicht mit ihren Hashwerten übereinstim-

men.

Da alle Stationen in einem Ad-hoc Netzwerk mit proaktiven Algorithmen regelmäßig ihre Nachbarschaftsbeziehungen kontrollieren müssen, senden die Stationen gelegentlich sogenannte „Hallo"-Pakete aus, die neue Nachbarn über die Kennung des Senders informieren und bestehenden Nachbarn die Kontrolle der Verbindung ermöglichen. Es bietet sich an, an mit diesem „Hallo"-Paket neben der Kennung auch den Hashwert des eigenen Baumes zu versenden. Die Nachbarn werden damit regelmäßig über den aktuellen Hashwert informiert und können dann umgehend eine Nachsendung anfordern, wenn ihr gespeicherter Hashwert nicht mehr übereinstimmt.

5.3.3 Algorithmusbeschreibung von TERA

Beim TERA werden Botschaften in der gleichen Weise wie beim DV-Routing ausgetauscht. Die Botschaften sind, wie beim DV, Vektoren, allerdings mit einer Erweiterung, die das Rückverfolgen einzelner Pfade zu den Zielen erlaubt. Damit werden die Hauptprobleme des DV umgangen: Es tritt kein Count-to-Infinity Problem auf und Schleifen lassen sich erkennen und verhindern.

Für die Algorithmusbeschreibung werden die selben Definitionen verwendet wie sie bereits bei der einfachen Pfadsuche in 5.2 beschrieben sind. Jede Station erhält einen eindeutigen Bezeichner k. Die Distanz zwischen einer Ausgangsstation und einer Zielstation wird durch den Wert d_{ij} angegeben und muss positiv sein.

Datenstrukturen

Auch die verwendeten Tabellen sind mit der einfachen Pfadsuche identisch. Jede Station k definiert ihren Zustand durch fünf Tabellen:

- Die Routingtabelle D_i^k, die für jedes Ziel i die bestmögliche Distanz angibt.
- Die Nachfolgertabelle N_i^k, die beschreibt über welchen Nachbarn das Ziel zu erreichen ist.
- Die Vorgängertabelle V_i^k, die den jeweils vorletzten Knoten im Pfad angibt.
- In $ND_{i\ j}^k$ wird die beste Distanz zu einem Ziel i angegeben, die der Knoten j errechnet und übermittelt hat.
- Zu der Tabelle der besten Distanzen muss auch eine Tabelle der entsprechenden vorletzten Knoten im Pfad gespeichert werden. In $NV_{i\ j}^k$ wird zu jedem Ziel der vorletzte Knoten angegeben, den der Nachbar j übermittelt hat.

Initialisierung

Die Kommunikation zwischen den Knoten geschieht durch Botschaften, die den Zustand eines Knotens verändern. Der Zustand wird durch den Inhalt der Tabellen $D_i^k(t), N_i^k(t), V_i^k(t), ND_{i\,j}^k(t)$ und $NV_{i\,j}^k(t)$ festgelegt. Zu Anfang, bei $t = 0$ initialisiert jeder Knoten seine Tabellen:

- $D_i^k(0) := \infty \; \forall \; i,k \;\; i \neq k$, und $D_k^k := 0 \; \forall \; k$
- $N_i^k(0) :=$ *"none"* $\forall \; i,k$
- $V_i^k(0) :=$ *"none"* $\forall \; i,k$
- $ND_{i\,j}^k(0) := \infty \; \forall \; i,j,k$
- $NV_{i\,j}^k(0) :=$ *"none"* $\forall \; i,j,k$

Botschaften an die Nachbarn

Nachdem alle Knoten ihre Tabelle initialisiert haben beginnt die Iteration. Im ersten Schritt werden die Tabellen D_i^k, V_i^k (die anfangs nur ein Element enthalten) an alle Nachbarn gesendet. Jeder Knoten verarbeitet die erhaltenen Informationen und sendet im zweiten Schritt die neuen Informationen wieder an seine Nachbarn. Auf diese Weise breitet sich die Information mit jeder Iteration um einen Hop weiter aus.

Jeder Knoten speichert die empfangene Information in seinen Tabellen $ND_{i\,j}^k$ und $NV_{i\,j}^k$. Wenn Knoten k die Tabellen $D_i^j(t-1), V_i^j(t-1)$ von Nachbar j empfängt, speichert er den darin enthaltenen minimal spannenden Baum für die anschließende Routenberechnung:

Zuerst werden alle alten Einträge gelöscht.

- $ND_{i\,j}^k(t) := \infty \; \forall \; i$
- $NV_{i\,j}^k(t) :=$ *"none"* $\forall \; i$

Danach wird der neue Baum eingefügt

- Zu allen von Nachbar j angegebenen Distanzen muss noch die Distanz von k zu Nachbar j hinzuaddiert werden: $ND_{i\,j}^k(t) := D_i^j(t-1) + d_{k\,j} \; \forall \; i$.
- Die Vorgängertabelle bleibt unverändert $NV_{i\,j}^k(t) := V_i^j(t-1) \; \forall \; i$.

Schließlich muss noch der Eintrag in der Vorgängertabelle, der sich auf den letzten Hop (also die Kante von Knoten k zu Knoten j) bezieht, korrigiert werden, denn die Wurzel des empfangenen minimal spannenden Baumes besitzt noch keinen Vorgänger. Nun wird Knoten k als Vorgänger eingetragen.

- $NV^k_{j\,j}(t) := k$

Aus den nun vorliegenden Tabellen $ND^k_{i\,j}(t)$ und $NV^k_{i\,j}(t)$ errechnet Knoten k nun seinen eigenen minimal spannenden Baum und speichert ihn in die Tabellen $D^k_i(t)$, $N^k_i(t)$ und $V^k_i(t)$.

Wenn das Ergebnis dieser Berechnung sich von der letzten Iteration unterscheidet, dann müssen die Tabellen $D^k_i(t)$, $V^k_i(t)$ erneut an alle Nachbarn gesendet werden.

Alle Distanz-Vektor Algorithmen nutzen diese Art der Iteration, dabei werden immer nur neue und nur relevante Informationen an den Nachbarn weitergereicht, dies erspart gegenüber den Link-State Algorithmen viel Kommunikationsaufwand.

Berechnung lokaler minimal spannender Bäume

Nach dem Empfang neuer Informationen von einem Nachbarn errechnet ein Knoten für sich einen neuen minimal spannenden Baum. Die Berechnung geschieht durch einen speziell angepassten Dijkstra-Algorithmus.

Jeder Knoten k hat sämtliche minimal spannenden Bäume der Nachbarn in seinen Tabellen $ND^k_{i\,j}$ und $NV^k_{i\,j}$ gespeichert, wobei die Bäume nach dem Empfang so modifiziert wurden, dass diese alle Unter-Bäume der Wurzel k sind. Der Algorithmus von Dijkstra kann darauf direkt angewendet werden und liefert dann den minimal spannenden Baum bezüglich k.

Da der Algorithmus in jedem Knoten einzeln ausgeführt wird und während der Berechnung keine Kommunikation mit Nachbarn notwendig ist, werden im Folgenden die Angaben zu den Iterationsschritten t weggelassen.

Vor Beginn werden die Tabellen D^k_i, N^k_i, V^k_i von Station k gelöscht: $D^k_i := \infty \; \forall \; i$,
$N^k_i :=$ *"none"* $\forall \; i$ und $V^k_i(0) :=$ *"none"* $\forall \; i$.

Die Tabelle D^k_i beschreibt die vorläufig angenommene Distanz zu allen Zielen. Diese Werte werden immer dann, wenn bessere Distanzen gefunden werden, entsprechend angepasst. Alle Ziele, deren Distanz bereits feststeht, sind in der Tabelle P^k aufgelistet. Der Algorithmus benötigt zusätzlich noch eine Kandidatenliste C^k, von der schrittweise jeweils ein Element in die Liste mit feststehenden (permanenten) Distanzen wechselt.

Vor dem ersten Durchlauf wird die Distanz der Wurzel k auf Null gesetzt $D^k_k := 0$ und permanent gemacht $P^k := \{k\}$. Die Liste der Kandidaten wird mit den Nachbarn von k gefüllt: $C^k := C^k \cup \{i \mid d_{ik} < \infty\}$ und die entsprechenden Distanzen werden eingetragen: $D^k_i := d_{ik} \; \forall \; \{ \; i \mid d_{ik} < \infty \; \}$.

Zusätzlich müssen alle Nachbarn als mögliche Nachfolger eingetragen werden: $N_i^k := i \; \forall \; \{ \; i \mid d_{ik} < \infty \; \}$ und jeder Nachbar nutzt die Wurzel k als Vorgänger $V_i^k := k \; \forall \; \{ \; i \mid d_{ik} < \infty \; \}$.

Die Berechnung der kürzesten Pfade erfolgt in einer Schleife, die solange wiederholt wird, bis keine weiteren Knoten mehr der Tabelle P^k hinzugefügt werden können.

- Der Kandidat mit der kleinsten Entfernung wird gesucht:
  $BD^k := min[D_i^k] \; \forall \; \{ \; i \mid i \in C^k \; \}, \quad BI^k :=$ Kennung des Knotens mit der Distanz BD^k.
- Der beste Kandidat wird permanent: $P^k := P^k \cup \{BI^k\}$
- Der Kandidat wird aus der Kandidatenliste entfernt: $C^k := C^k \setminus \{BI^k\}$
- Alle Nachfolger von BI^k werden nun Kandidaten:

$$C^k := C^k \cup \; \{i \mid i \notin P^k \; und \; NV^k_{i \; (N^k_{BI^k})} = BI^k\}$$

- Die Distanz der neuen Kandidaten wird errechnet:

$$D_i^k := DV^k_{i \; (N^k_{BI^k})} \forall \; \{ \; i \mid i \notin P^k \; und \; NV^k_{i \; (N^k_{BI^k})} = BI^k \; \}$$

- Die Nachfolger der neuen Kandidaten werden eingetragen:

$$N_i^k := N^k_{(BI^k)} \forall \; \{ \; i \mid i \notin P^k \; und \; NV^k_{i \; (N^k_{BI^k})} = BI^k \; \}$$

- Die Vorgänger der neuen Kandidaten werden eingetragen:

$$V_i^k := NV^k_{i \; (N^k_{BI^k})} \forall \; \{ \; i \mid i \notin P^k \; und \; NV^k_{i \; (N^k_{BI^k})} = BI^k \; \}$$

Nach Beendigung der Schleife steht ein neuer minimal spannender Baum für Knoten k in den Tabellen $D_i^k(t), N_i^k(t), V_i^k(t)$ zur Verfügung. Dieser neue Baum wird nun mit seinem Vorgänger verglichen, um neue Knoten, veränderte Distanzen oder Positionswechsel zu erkennen.

Kriterien für ein Update

Da jede Veränderung der eigenen Routingtabellen weiterzumelden ist, müssen die Tabellen $D_i^k(t)$ and $V_i^k(t)$ jedes Mal an die Nachbarn weitergeleitet werden, wenn eine der folgenden Bedingungen zutrifft:

$$D_i^k(t) \neq D_i^k(t-1) \; \exists \; i \quad \vee \quad N_i^k(t) \neq N_i^k(t-1) \; \exists \; i \quad \vee \quad V_i^k(t) \neq V_i^k(t-1) \; \exists \; i \tag{5.1}$$

Routenpflege

Damit die Routen langfristig funktionieren, müssen jedes Mal beim Auftauchen neuer Ziele und beim Verlust von Zielen die Routen angepasst werden.

Neue Ziele Wenn ein neuer Nachbar von Knoten k entdeckt wird, dann sind Korrekturen an den Tabellen $ND^k_{i\,j}$ und $NV^k_{i\,j}$ notwendig, um den neuen Knoten in die Routenberechnung mit einzubeziehen. Dazu wird zuerst der Nachbar mit seiner Kennung n in $ND^k_{n\,n} := d_{kn}$ eingetragen und k als sein Vorgänger $NV^k_{n\,n} := k$ gesetzt. Damit ist der neue Nachbar als Unterbaum unter der Wurzel k eingetragen und danach kann mit Hilfe des oben beschriebenen Algorithmus in Kapitel 5.3.3 ein neuer minimal spannender Baum berechnet werden. Wenn der neue Nachbar bessere Distanzen anbietet, dann wird die Überprüfung der Kriterien, wie sie in Abschnitt 5.3.3 beschrieben wurde, dies erkennen und die Informationen an die Nachbarn weitergeben.

Entfallene Ziele Ein Verlust einer Verbindung von Knoten k zu Nachbar n erfordert, dass die Distanz d_{kn} auf unendlich gesetzt wird. Knoten k muss dann prüfen, ob es Routen gibt, die diese Verbindung nutzen. Die Tabelle $ND^k_{i\,n}$ muss für alle i auf ∞ gesetzt werden und Tabelle $NV^k_{i\,n}$ muss für alle i auf "*none*" gesetzt werden. Dann wird ebenfalls ein neuer Baum wie in 5.3.3 beschrieben berechnet und eine Prüfung der Kriterien 5.3.3 durchgeführt. Treffen die Kriterien zu, dann werden die Nachbarn über die Veränderungen informiert.

5.4 Vergleich der einfachen Pfadsuche mit TERA

Die einfache Pfadsuche und TERA beruhen auf grundsätzlich unterschiedlichen Ideen. Während die einfache Pfadsuche immer darum bemüht ist, möglichst viel einer Routingtabelle beizubehalten, berechnet TERA jedes mal den kompletten Spannbaum neu. Die einfache Pfadsuche benötigt so relativ wenige Berechnungen, da sie immer nur einzelne Ziele in ihre Tabelle einfügt und für jedes Ziel nur die Gültigkeit des Pfades prüfen muß. Dagegen verursacht TERA einen vielfach höheren Berechnungsaufwand, denn jede Veränderung erzwingt die komplette Neuberechnung aller Routen. Durch diese Vorgehensweise werden jedoch immer die neuesten Informationen für alle Ziele genutzt, damit werden mehr Ziele erreichbar.

Das wird in Abbildung 5.3 deutlich, in der die Ergebnisse von Simulationen mit 50 Knoten in einem Raum der Größe 1000×1000 Meter mit adaptiver Verbindungserkennung bei einer Sendereichweite von 200 Metern darstellt sind. Die Ergebnisse stammen dabei von dem im Rahmen dieser Arbeit entwickelten Simulator, wobei ein mal pro simulierter Sekunde sämtliche Routen des Netzwerks überprüft wurden. Mit der einfachen Pfadsuche und für TERA wurden die Simulationsläufe bei verschiedenen Geschwindigkeiten der Knoten durchgeführt, wobei jedes Mal eine Zeitspanne von einer Stunde simuliert wurde. Die Knotengeschwindigkeiten sind über der x-Achse aufgetragen, während über der y-Achse jeweils die nutzbaren bzw. nicht nutzbaren Routen für den jeweiligen Algorithmus zu sehen sind.

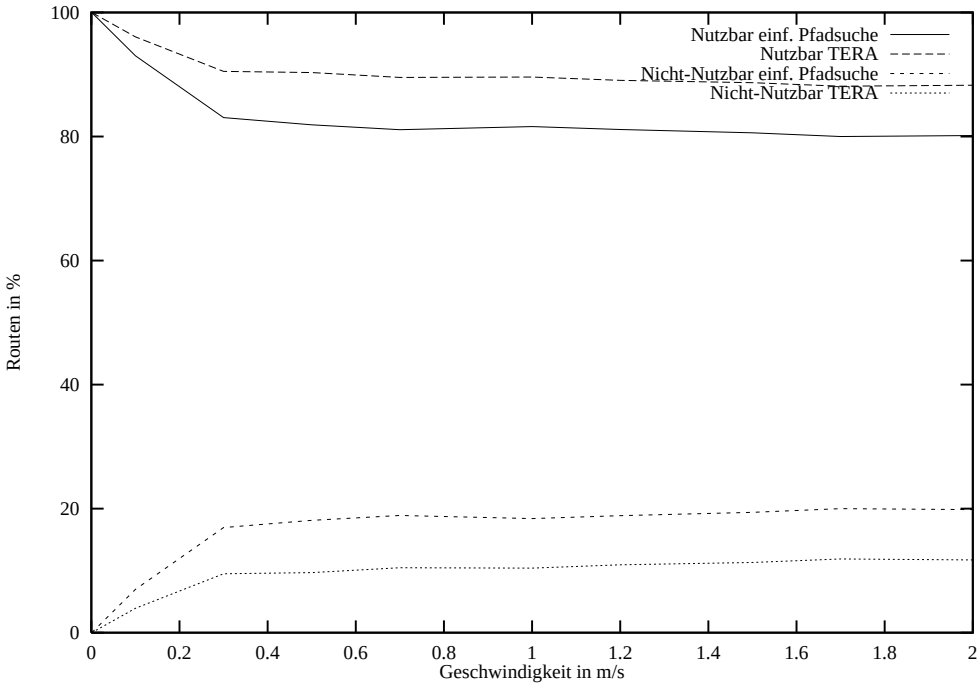


Abbildung 5.3: Vergleich von TERA mit der einfachen Pfadsuche

Die Auswertung der Simulation beruht auf Stichproben, die gemittelt werden. Die an der y-Achse aufgetragene Prozentzahl gibt dabei für jeden Simulationsdurchgang entweder den Anteil der Routen an, die zum gewünschten Ziel führen oder, bei den nicht nutzbaren Routen, den Anteil der Routen, bei denen dies nicht gelang. Bei der sehr kleinen Geschwindigkeit von 0,001 m/s, d.h. so gut wie keiner Veränderung der Netzwerktopologie, wird eine Erfolgsquote von 100 % erreicht, entsprechend ist die Fehlerqoute 0 %. Bei dieser geringen Geschwindigkeiten haben beide Algorithmen genügend Zeit, sich auf die Topologie einzustellen.

Während der Simulation besteht immer die Gefahr, dass der Netzwerkgenerator zufällig ein partitioniertes Netz erzeugt. Da in einem solchen Netzwerk die Stichproben auch Stationen erfassen, die in verschiedenen Partitionen liegen, werden so Fehler gezählt, obwohl der Algorithmus richtig arbeitet. Die Wahrscheinlichkeit, für das Auftreten dieses Falls kann mit der Gleichung 4.5 auf Seite 61 bestimmt werden. Mit den oben angegebenen Parametern errechnet sich so eine Wahrscheinlichkeit für die Erreichbarkeit von 0.997, damit ist die Fehlerwahrscheinlichkeit $1 - 0.997 = 0.003$.

Selbst wenn dieser sehr unwahrscheinliche Fehler auftritt, dann sind beide Vergleichsalgorithmen gleichermaßen davon betroffen. Dies ist durch den Aufbau des Simulators garantiert, denn die Topologieinformation wird vorab durch die erste Stufe des Simulators erzeugt.

Steigt die Bewegungsgeschwindigkeit der Knoten an, dann müssen die Routingalgorithmen immer häufiger auf Topologieveränderungen reagieren. Dabei wird es für die Routingalgorithmen bei steigender Geschwindigkeit immer schwieriger, die Informationen über die augenblickliche

Netztopologie an alle Knoten zu verteilen, damit diese sie in ihre Berechnung der Routingtabellen einfließen lassen können. Durch die fehlenden oder veralteten Informationen entstehen dann fehlerhafte Routen, die mit den Stichproben gemessen werden.

Es ist gut zu erkennen, dass TERA deutlich mehr funktionierende Routen produziert als die einfache Pfadsuche. Um die Ursache hierfür festzustellen, wurden die nicht funktionierenden Routen genauer analysiert. Dabei wurden die nicht nutzbaren Routen in defekte Routen (ein verwendeter Link existiert nicht), Schleifen (das Datenpaket wird im Kreis geschickt) und Fälle, in denen Knoten unerreichbar sind, unterteilt. Während bei den Schleifen und den defekten Routen kein großer Unterschied feststellbar ist, ist der Unterschied bei den unerreichbaren Knoten jedoch sehr deutlich. Das Ergebnis der Simulation ist in Abbildung 5.4 zu sehen.

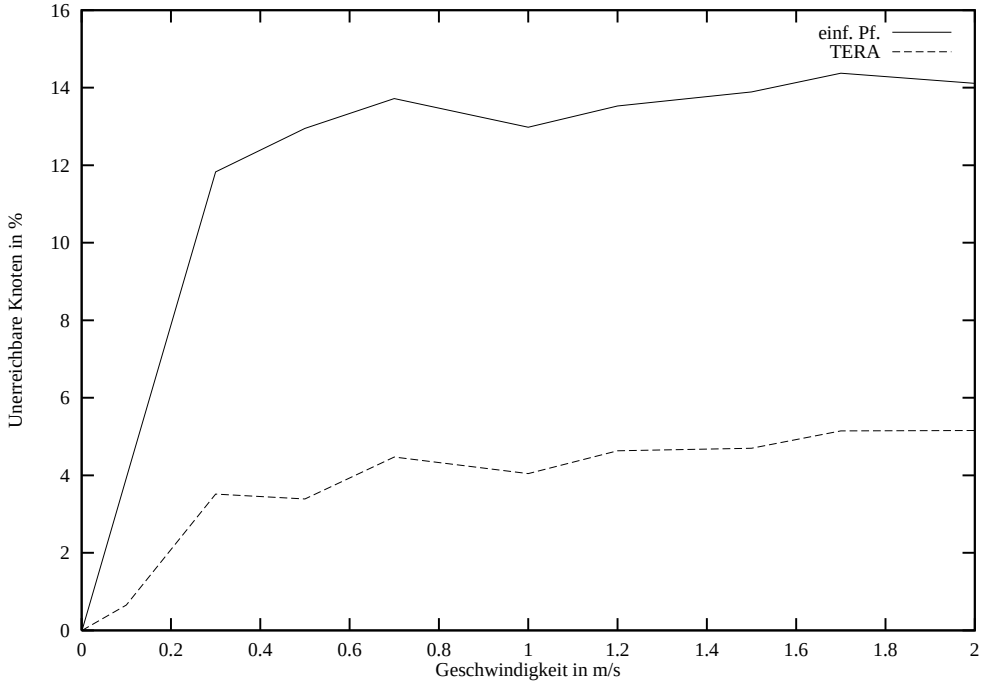


Abbildung 5.4: Unerreichbare Knoten von TERA und der Pfadsuche

Zur Erklärung dieses Unterschiedes wird im folgenden ein Beispiel vorgestellt, in dem die einfache Pfadsuche einige neue Ziele in ihre Tabellen aufnehmen muss.

Es wird dabei das Beispielnetzwerk aus Abbildung 5.5 unter der Annahme betrachtet, dass noch keine Verbindung zu Knoten A existiert, weder von B noch von C. Wenn die Verbindung zwischen den Knoten A und B jetzt neu entsteht, tauschen die beiden ihre Routingtabellen aus. B teilt dabei A mit, dass er die Knoten F und G erreichen kann, allerdings kommen diese Pakete in umgekehrter Reihenfolge bei A an, so dass A zuerst versucht, Knoten G in seinen minimal spannenden Baum einzufügen. Dies gelingt jedoch nicht, da der Vorgänger von G, nämlich F, noch nicht im Baum enthalten ist. Im nächsten Schritt wird dann der Knoten F in den Baum eingefügt. Knoten G wird jedoch nicht noch einmal bearbeitet, so dass er für A unerreichbar bleibt. Für die einfache Pfadsuche ist also die Reihenfolge der eintreffenden Routingbotschaften

entscheidend. Dieses Problem tritt nicht häufig auf, denn selten werden weiter entfernte Knoten vor ihren Vorgängern gemeldet. Allerdings reicht dieses Problem aus, um die Unterschiede in der Erreichbarkeit bei vielen Knoten und großer Geschwindigkeit zu erklären.

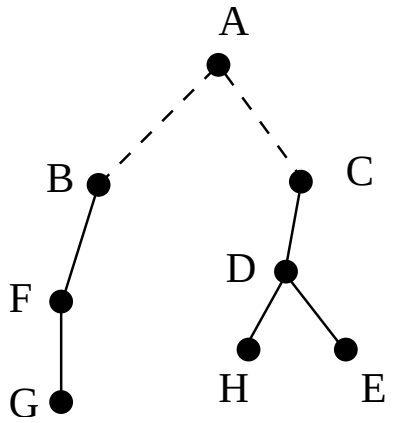


Abbildung 5.5: Beispielnetz und minimal spannender Baum für Knoten A

TERA leidet hingegen nicht an diesem Problem, da bei jeder Veränderung der komplette Spannbaum neu erzeugt wird. Damit werden dann auch Routen zu Zielen eingefügt, die vorher wegen unvollständiger Pfadinformation nicht erreichbar waren. In dem oben beschriebenen Beispiel würde Knoten A also beim ersten Versuch G nicht einfügen können. Wenn jedoch die Nachricht eintrifft, dass auch Knoten F erreicht werden kann, wird die komplette Routingtabelle neu berechnet und dabei auch der Pfad nach G über F gefunden, so dass sowohl F als auch G erreichbar werden.

5.5 Zusammenfassung

Das Kapitel beschreibt im Detail die Routenberechnung durch Pfadsuche. Die Beschreibung beginnt mit dem Algorithmus für die einfache Pfadsuche. Die dabei verwendeten Tabellen und die einzelnen Operationen wurden ausführlich erklärt. Ein Beispiel erläutert die Funktionsweise der Schleifensperre, die der Hauptvorteil gegenüber normalen DV-Algorithmen ist. Daran schließt sich eine kritische Betrachtung des Algorithmus an, die noch bestehende Probleme der einfachen Pfadsuche aufzeigt.

Anschließend wurde das Funktionsprinzip eines neuen Algorithmus vorgestellt, der auf dem Austausch von Bäumen basiert und deshalb auch Tree Exchange Routing Algorithm (TERA) genannt wird. Die Routenberechnung mit TERA wurde anhand eines Beispieles erklärt. Danach wurde auf den vermeintlich höheren Kommunikationsbedarf von TERA eingegangen und eine Methode vorgestellt, die den Kommunikationsbedarf auf ein ähnliches Niveau wie bei der einfachen Pfadsuche senkt. Eine Spezifikation des neuen Algorithmus wurde angegeben, dabei wurden die Unterschiede zur einfachen Pfadsuche erläutert.

Das Kapitel endete mit einem Vergleich der einfachen Pfadsuche und TERA. Die beiden Algorithmen wurden bei unterschiedlicher Knotenmobilität miteinander verglichen. Dabei zeigte sich, dass TERA und die einfache Pfadsuche bei einem Netzwerk ohne Topologieveränderungen identische Erreichbarkeit erzielen. Bei erhöhter Knotenmobilität errechnet TERA jedoch immer mehr funktionsfähige Routen als der Vergleichsalgorithmus. Eine Analyse der Unterschiede zeigte, dass TERA durch seine umfangreicheren Berechnung Vorteile gegenüber der einfachen Pfadsuche erzielt. Da TERA immer ganze Bäume neu berechnet, nutzt er die vorhandenen Informationen besser aus und erkennt so in einigen Fällen mehr funktionierende Routen.

Kapitel 6

Steigerung der Bandbreiteneffizienz

6.1 Einleitung

Dieses Kapitel beschreibt zwei Verfahren zur verbesserten Ausnutzung der Funkbandbreite beim Routing. Das erste Verfahren beschäftigt sich mit der Nachbarschaftserkennung, die von den meisten Routingalgorithmen als vorhandener Dienst einfach vorausgesetzt wird. Das zweite Verfahren erweitert den im vorigen Kapitel beschriebenen Routingalgorithmus.

Im ersten Abschnitt wird die adaptive Nachbarschaftserkennung vorgestellt. Obwohl dieser Dienst nur wenig Beachtung findet, hat er einen großen Einfluss auf die Qualität des Routing. Eine schlechte Nachbarerkennung führt zu fehlerhaften Routen und damit zu Problemen, die sich über das ganze Netzwerk ausdehnen. Durch das Abschätzen der Mobilität von Stationen wird hier die Erkennung von neuen Nachbarn und Verbindungsverlusten verbessert. Dies geschieht durch ein Anpassen der regelmäßig gesendeten Kennungen an die Mobilität im Netzwerk. In Netzwerken mit wenig Mobilität lässt sich durch Reduktion der gesendeten Kennungen Bandbreite sparen, bei hoher Mobilität steigt zwar der Bandbreitenbedarf wieder an, dafür wird die Nachbarerkennung aber wesentlich schneller. Die Beschreibung endet mit einer Reihe von Simulationen, in denen die Eigenschaften der adaptiven Nachbarerkennung untersucht werden.

Der zweite Abschnitt des Kapitels beschreibt einen Verbesserungsvorschlag für TERA, um dessen Bandbreitenbedarf weiter zu reduzieren. Dabei wird von der bisher verwendeten Methode abgewichen, immer die kürzesten Pfade zu berechnen. Dem Algorithmus wird ein gewisser Spielraum eingeräumt, so dass er nicht mehr unbedingt den kürzesten Pfad suchen muss. Die dabei erzielten Einsparungen werden ebenfalls durch eine Reihe von Simulationen belegt.

Die Simulationen zur adaptiven Nachbarschaftserkennung wurden in [JF98] vorgestellt, die Erweiterung von TERA wurde in [JF01] veröffentlicht.

6.2 Adaptive Nachbarschaftserkennung

Die Grundlage für einen kontinuierlichen Routingvorgang sind Listen über Nachbarschaftsbeziehungen, die in allen Knoten vorhanden sein müssen, die am Routingvorgang beteiligt sind. Da sich Nachbarschaftsbeziehungen unter Umständen sehr schnell ändern können, müssen die Listen ständig überprüft und gegebenenfalls auf den aktuellen Stand gebracht werden. Diese kontinuierliche Wartung der Listen verursacht Kosten durch Energieverbrauch und Belegung von Bandbreite. Im folgenden Abschnitt werden die Qualität der Wartung und die dabei entstehenden Kosten näher untersucht.

In einem drahtlosen Netzwerk wird ein neuer Nachbar erst dann bemerkt, wenn er ein Signal aussendet. Ebenso wird der Verlust eines Nachbarn nur dann bemerkt, wenn eine Kommunikation zu dem Nachbarn nicht mehr möglich ist. Der Zustand einer Nachbarschaftsbeziehung lässt sich nur durch aktive Signalisierung und Prüfsignalisierung feststellen, welche jedes Mal Energie und Bandbreite verbraucht.

Die einfachste Methode zur Überprüfung der Verbindungen zu den Nachbarn ist die regelmäßige Aussendung eines sogenannten „Hallo“-Signales (Beaconing). Jeder Knoten sendet dazu periodisch ein Hallo-Signal aus, um seine Nachbarn von seiner Gegenwart zu informieren. Das gesendete Hallo-Signal wird gleichzeitig von allen Nachbarn in Reichweite empfangen. Diese aktualisieren anschließend ihre Nachbarlisten. Wird ein Hallo-Signal mit einer bisher unbekannten Kennung empfangen, dann wird der neue Nachbar in die Liste eingefügt. Bleiben die regelmäßigen Hallo-Signale eines bekannten Nachbarn aus, dann wird der entsprechende Nachbar aus der Liste entfernt.

Um die Kosten für die Nachbardetektion gering zu halten, werden die regelmäßigen Signalisierungen möglichst selten durchgeführt, was zwangsläufig zu größeren Verzögerungen bei der Aktualisierung der Listen führt. Dabei bestimmt die gerade noch akzeptable Ungenauigkeit der Nachbarlisten die Grenzen dieser Einsparung.

Algorithmen, die regelmäßig den Status einer Verbindung abfragen, werden in drahtlosen Netzwerken häufig eingesetzt. Bei einem sehr geringen Abstand der Signale (z.B. 1 Sekunde) wird eine Veränderung durchschnittlich innerhalb von 0.5 Sekunden entdeckt. Diese Zeit reicht für die Anforderungen vieler Anwendungen vollkommen aus. Allerdings ist die dabei belegte Bandbreite relativ hoch. Sind die Anforderungen an die Erkennungszeit genau bekannt, dann kann problemlos eine Intervallzeit bestimmt werden, die diese Anforderungen erfüllt. Wenn sich die Knoten allerdings mit sehr unterschiedlichen Geschwindigkeiten bewegen, dann ist eine einheitliche Intervallzeit sehr ungünstig. In diesem Fall muss die Intervallzeit so gewählt werden, dass sie die Anforderungen des schnellsten Knotens noch erfüllt, wobei allerdings überhöhte Kosten für die langsameren Knoten entstehen.

In einem Netzwerk mit sehr unterschiedlichen Bewegungsgeschwindigkeiten führt ein festes Signalisierungsintervall entweder zu einer erhöhten Fehlerrate oder zu einer hohen Bandbreitenbelegung. Im Folgenden werden die Möglichkeiten untersucht, flexible Intervallzeiten einzusetzen.

6.2.1 Funktionsprinzip

Ein erster Ansatz für flexible Intervallzeiten ist die Einführung einer sofortigen Antwort auf neu entdeckte Nachbarn. Sobald ein Knoten das Hallo-Signal eines neuen Nachbarn empfängt, antwortet dieser Knoten mit einem Hallo-Signal unabhängig von seiner eigenen Intervallzeit. Durch diese Veränderung verkürzt sich die Erkennungszeit für einen neuen Nachbarn.

Zur Ermittlung der Erkennungsgenauigkeit kann die folgende Berechnung verwendet werden: Ein Knoten mit der Intervallzeit I sendet alle I Zeiteinheiten ein Signal aus. Kommt der Knoten in die Reichweite eines neuen Nachbarn, dann wird er innerhalb des Zeitraumes zwischen 0 und I entdeckt. Da alle Zeitpunkte gleichwahrscheinlich sind, kann als Erwartungswert der Mittelwert verwendet werden. Deswegen ist im einfachsten Fall der erwartete Zeitraum bis zur Entdeckung:

$$\frac{\int_0^I x\,dx}{I} = \frac{1}{2} \cdot I \tag{6.1}$$

Für den Fall einer sofortigen Antwort wird der zuerst aussendende Knoten entdeckt. Dieser sendet sofort eine Antwort und wird deshalb ebenfalls zu diesem Zeitpunkt entdeckt. Zur Berechnung der erwarteten Entdeckungszeit werden hier zwei voneinander unabhängige Zeitpunkte angenommen, die sich jeder Knoten als Aussendezeitpunkt gewählt hat. Die Entdeckungszeit ist dann das Minimum der beiden Zeitpunkte. Da die Zeitpunkte voneinander unabhängig sind, kann der Durchschnitt einfach errechnet werden:

$$\frac{\int_0^I \left(\int_0^I \min(x,y)\,dx \right) dy}{I^2} = \frac{\int_0^I \left(\int_0^y x\,dx + \int_y^I y\,dx \right) dy}{I^2} = \frac{1}{3} \cdot I \tag{6.2}$$

Die Gleichung 6.2 zeigt, dass durch Kooperation der Erwartungswert für das Entdecken eines neuen Knotens verbessert wird, leider ist die Entdeckung eines Verbindungsverlustes nicht auf diese Weise zu beschleunigen. Hier gilt weiterhin der Wert, der in der Gleichung 6.1 bestimmt wurde. Da sich die Topologieänderungen eines Netzwerks aus dem Auf- und Abbau von Verbindungen zusammensetzen, liegt die Zeit für die Entdeckung einer Netzwerkveränderung zwischen den beiden genannten Werten, also zwischen $\frac{I}{2}$ und $\frac{I}{3}$.

Bei einer konstanten Bewegungsgeschwindigkeit der Knoten ist die Entdeckungszeit proportional zum Fehler, da die erkannte Topologie dem aktuellen Stand hinterherläuft. Der Fehler wird nun aber durch die schnellere Erkennung reduziert.

Die Anzahl der benötigten Botschaften ist für ein Signalisierungsverfahren mit konstanten Intervallen und einer sofortigen Antwort nur noch für unbewegte Knoten konstant. Schnelle Bewegungen können die Zeiten zwischen den Signalen erheblich reduzieren. Diese Eigenschaft wird im Anschluss durch Simulationen genauer untersucht.

Die Einführung einer sofortigen Antwort ist aber nur ein erster Schritt hin zu einer vollständigen Adaption. Eine vollständig adaptive Lösung passt die Signalisierungsintervalle den Bedürfnis-

sen im Netzwerk an. Dabei setzt jeder Knoten sein Signalisierungsintervall selbst fest. Er schätzt aus den verfügbaren Daten seine Bewegungsgeschwindigkeit und regelt dementsprechend seine Intervalle. Um aber die Geschwindigkeit schätzen zu können, müssen Daten aus vorangegangen Signalisierungen gesammelt werden. Wird in einem solchen adaptiven System die Regel zur sofortigen Antwort eingesetzt, dann verkürzen sich die Zeiten für die Regelung und die Adaption wird beschleunigt. Die Adaption arbeitet allerdings nur dann korrekt, wenn die gesammelten Daten richtig sind. Wie bereits vorher erläutert, bewirkt eine Erhöhung der Intervalle nur bei den Nachbarn eine verkürzte Erkennungszeit. Die Adaption funktioniert nur dann richtig, wenn die Nachbarn diesen Vorteil durch sofortiges Antworten an den Sender zurückgeben, der dann seinerseits aktuelle Daten sammeln kann.

Wird die Anzahl der Verbindungsänderungen in der Umgebung eines Knotens erfasst, dann kann ein Signalisierungsintervall für diesen Knoten bestimmt werden, das eine Fehlergrenze sicher einhält. Die gemessenen Veränderungsraten werden benutzt, um die nächsten Intervalle zu bestimmen. Dadurch passen sich die Intervalle den Veränderungsraten der Umgebung an. Wenn sich ein Knoten schnell bewegt, dann wird er eine größere Menge von Veränderungen in seiner Umgebung feststellen. Als Reaktion darauf erhöht er seine Signalisierungsrate und bewirkt damit eine reduzierte Erkennungszeit.

Jeder Knoten schätzt also aus den gemessenen Verbindungsänderungen seine relative Geschwindigkeit zu seiner Umgebung. Er kann auch die Intervalle seiner Nachbarn messen und daraus Rückschlüsse über die Geschwindigkeit ziehen. Die Verwendung von solchen indirekten Parametern führt oft zu unvorhersehbaren Ergebnissen, da sich die Knoten dadurch immer wieder gegenseitig beeinflussen. Deswegen sollten indirekte Parameter nur eine untergeordnete Rolle bei der Berechnung spielen. In den folgenden Simulationen wurden nur direkte Parameter verwendet. Ob indirekte Parameter mehr Vor- oder Nachteile zeigen, wurde in der vorliegenden Arbeit nicht mehr untersucht.

In drahtlosen Netzen ist die Erkennung eines Verbindungsabbruches nicht trivial. Wenn eine Verbindung zwischen zwei Knoten zerbricht, dann entstehen keine Verlustmeldungen, nur die fehlenden Signale der Nachbarn zeigen den Verlust an. Da drahtlose Verbindungen oft hohe Fehlerraten haben, ist ein fehlendes Signal aber kein eindeutiger Beweis für eine Verbindungsunterbrechung, das Signal könnte auch durch eine kurzfristige Störung verlorengegangen sein. Um eine Unterbrechung zu überprüfen, kann ein einfaches Testsignal benutzt werden, auf das der Nachbar mit hoher Priorität antworten sollte. Wenn solche Signale nicht innerhalb einer kurzen Zeit beantwortet werden, dann kann die Verbindung als unterbrochen angesehen werden.

Dynamische Intervalle erschweren die Erkennung von Unterbrechungen, da ein ausbleibendes Signal auch durch vergrößerte Intervalle ausgelöst werden kann. Deswegen wird in den durchgeführten Simulationen jedem Signal eine Zeitangabe beigefügt, die angibt, wann das nächste Signal zu erwarten ist.

Um genauere Aussagen über die Eigenschaften der vorgeschlagenen Verbesserungen machen zu können, wurden sie durch Simulationen getestet. Der Algorithmus zur dynamischen Adaption zählt die aufgetretenen Veränderungen der Netzwerktopologie. Die Veränderungsrate ist dabei der zentrale Parameter zur Berechnung der Signalisierungsrate. Bei der Simulation wird

für jeden Knoten ein eigenes Objekt angelegt. Dieses Objekt sendet und empfängt die Signale wie eine reale Station. Die Signale laufen über einen zentralen Verteiler, der die Position der Stationen kennt und Signale nur an Stationen innerhalb der Reichweite des Senders weitergibt. Jede Station legt sich eine eigene Liste an, in der die registrierten Veränderungen und die dazugehörigen Zeiten eingetragen werden. Deswegen hat jede Station ihre eigene Sicht auf die Umgebung. Aus dieser Sicht heraus wird dann die eigene Signalisierungsrate berechnet.

Ein Signal, welches zwischen zwei Stationen übertragen wird, enthält folgende Kenndaten:

- die Kennummer (ID) des Senders,
- die Zeitspanne bis das nächste Signal zu erwarten ist,
- Statusinformationen über den Sender.

Jede Station legt eine Liste mit einem Eintrag für jeden Nachbarn an. In dieser Liste werden die Daten des letzten Signales des Nachbarn festgehalten. Wenn ein neuer Nachbar entdeckt wird, dann legt die Station einen neuen Eintrag an. Geht die Verbindung zum Nachbarn verloren, dann wird der Nachbar aus der Liste entfernt. Das geschieht immer dann, wenn die Zeit für das erwartete nächste Signal abgelaufen ist.

Jedes Anlegen oder Löschen eines Listeneintrages ist ein Ereignis, das in einer weiteren Liste festgehalten wird. Diese Ereignisliste sammelt so die Zeitpunkte vergangener Topologieänderungen. Da die Liste eine begrenzte Größe hat, müssen die ältesten Einträge gelöscht werden, wenn die Liste voll ist und neue Einträge hinzukommen. Aus der Liste wird die durchschnittliche Zeit zwischen Topologieänderungen bestimmt. Dazu wird mindestens ein Eintrag benötigt. Der Durchschnitt wird mit veränderlichen Gewichten berechnet, so dass weiter zurückliegende Ereignisse weniger Einfluss auf das Ergebnis haben als aktuelle Werte. Die Funktionsweise der automatischen Anpassung ist in Abbildung 6.1 dargestellt.

Die Berechnung kommt mit wenigen Parametern aus. Jede Station hat eine Ereignistabelle der Länge N mit Einträgen, die die Dauer zwischen den Veränderungen beschreiben. Diese Tabelle ist zu Anfang leer und kann dann bis zur Größe von N_{max} wachsen. Die Tabelleneinträge werden mit T_1 bis T_N bezeichnet. Ein neuer Eintrag wird immer als T_1 in die Tabelle eingefügt. Der Index aller weiteren Einträge erhöht sich jeweils um 1. Wenn die maximale Größe der Tabelle bereits erreicht ist, wird der älteste Eintrag T_N gelöscht und durch seinen Vorgänger ersetzt.

Die Gewichte W_1 bis W_N sind eine Liste von Integer-Werten, mit denen festgelegt wird, welche Einträge welchen Einfluss auf das Ergebnis haben. Die Gewichte bestimmen, wie schnell der Algorithmus sich neuen Situationen anpasst.

Die durchschnittliche Zeit zwischen den Ereignissen CT_{avg} wird mit der Formel

$$CT_{avg} = \frac{\sum_{i=1}^{N} T_i \cdot W_i}{\sum_{i=1}^{N} W_i} \tag{6.3}$$

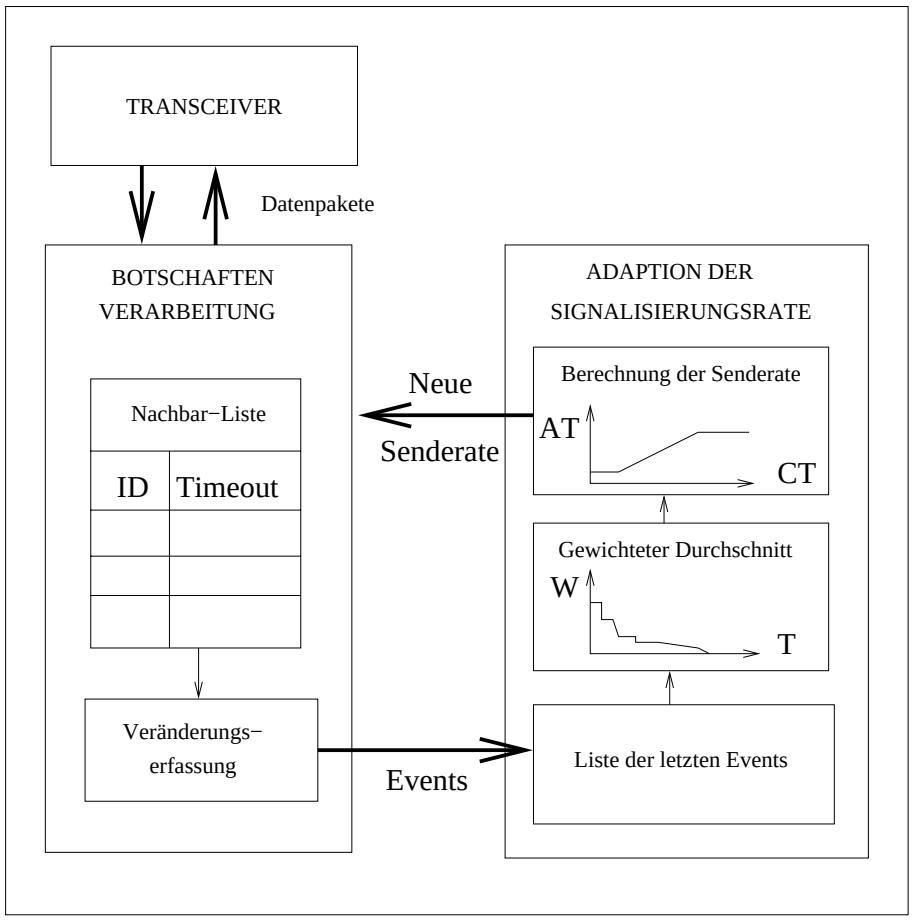


Abbildung 6.1: Funktionsprinzip der adaptiven Topologieerkennung

berechnet. Damit eine verlässliche Erkennung auch von sehr langsamen Veränderungen möglich ist, wird eine untere Grenze eingeführt, die dafür sorgt, dass Veränderungen auch nach einer sehr langen Stillstandsperiode (wenn auch mit Verzögerungen) erkannt werden. Zusätzlich wird eine obere Grenze eingeführt, die verhindert, dass eine Adaption überreagiert, wenn viele Veränderungen innerhalb einer kurzen Zeit auftreten. Die Parameter AR_{min} und AR_{max} beschreiben diese untere und obere Grenze, zwischen denen sich die Signalisierungsrate bewegen kann. Die Adaption AT_{ad} wird zusätzlich von einem Faktor AF gesteuert, der bestimmt, welche Signalrate für das ermittelte CT_{avg} verwendet wird. Sie wird mit der Formel 6.4 errechnet.

$$AT_{ad} = min(AR_{max}, max(AR_{min}, CT_{avg} \cdot AF)) \tag{6.4}$$

6.2.2 Simulationsergebnisse

Die Eigenschaften der festen und dynamischen Signalisierung werden im Folgenden durch einige Simulationsserien analysiert. Die Simulationen wurden mit Modulen aus dem Simulationssystem durchgeführt, das in Kapitel 4.4 dokumentiert ist.

Für die Simulationen wurde eine Umgebung mit folgenden Parametern angenommen:

- 100 Stationen,
- ein Grundfeld von 1000×1000 Metern,
- variable Bewegungsgeschwindigkeiten,
- eine Reichweite von 400 Metern.

Die Stationen bewegen sich auf dem Feld mit einer Geschwindigkeit zwischen 0 und 2 Metern pro Sekunde und haben die angegebene Reichweite für das Senden und Empfangen der Signale. Alle Verbindungen in den Simulationen sind verlustfrei. Sie entstehen und verschwinden nur durch die Mobilität der Stationen.

Größe der Ereignistab.	N	3
Min. Signalisierungsrate	AR_{min}	1 sec
Max. Signalisierungsrate	AR_{max}	60 sec
Signalisierungsfaktor	AF	1
Gewichtung	W_1, W_2, W_3	75,15,10

Tabelle 6.1: Parameter der Adaption

Die Simulationen analysieren zwei verschiedene Aspekte: Zum einen wird die Genauigkeit ermittelt, mit denen die Algorithmen die Netzwerktopologie erkennen, zum anderen wird der

Bandbreitenbedarf der Algorithmen bestimmt. Der Simulator kennt grundsätzlich alle Positionen der Stationen und ihre Verbindungen. Zur Messung der Genauigkeit der Topologieerkennung vergleicht der Simulator andauernd die von den Algorithmen errechnete Topologie mit dem aktuellen Netzwerk.

Alle Parameter, die für die Adaption verwendet werden, finden sich in Tabelle 6.1.

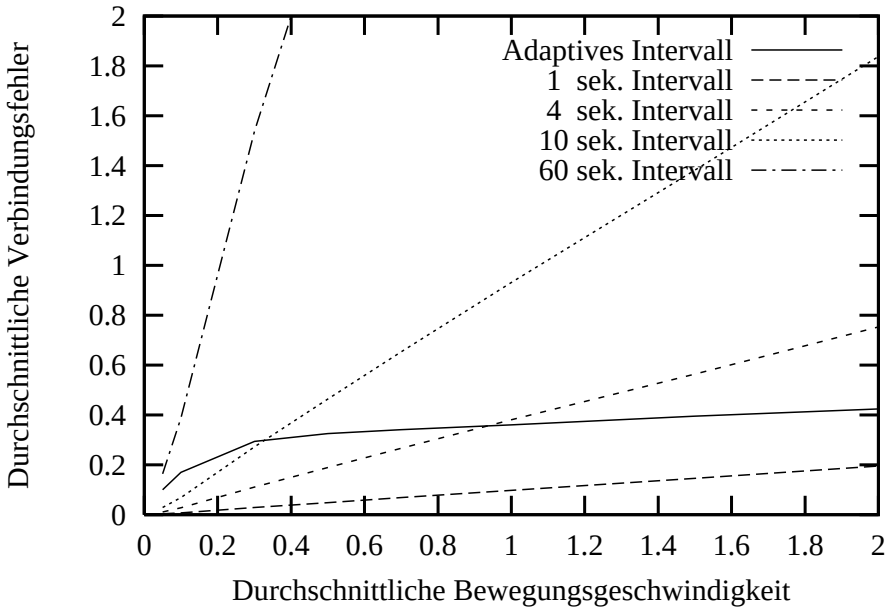


Abbildung 6.2: Fehlerrate der Nachbarerkennung

Das Bild 6.2 zeigt die Ergebnisse von fünf Simulationsserien. Auf der x-Achse sind die durchschnittlichen Knotengeschwindigkeiten in Metern pro Sekunde aufgetragen, mit denen die Stationen in der Simulationsserie bewegt wurden, auf der y-Achse ist die Fehlerrate der Erkennung aufgetragen, die durch die durchschnittliche Anzahl von falschen Verbindungen pro Knoten angegeben ist.

Jede Serie simuliert eine Stunde Netzwerkaktivität, in der sich die Stationen bewegen und Signale austauschen. In der Abbildung wird ein adaptiver Algorithmus mit vier Algorithmen mit festen Intervallzeiten von 1 bis 60 Sekunden verglichen.

Im Verlauf der Simulation verändert sich die Topologie des Netzes, manche Veränderungen werden von den Algorithmen nicht entdeckt. Diese Erkennungsfehler werden vom Simulator gezählt und als Messwert für die Leistungsfähigkeit des Algorithmus verwendet. Der Simulator zählt dabei verlorengegangene Verbindungen, die noch als verfügbar angesehen werden, genauso wie existierende Verbindungen, die nicht entdeckt wurden. Der Messwert wird über die Laufzeit der Simulation gemittelt und beschreibt, wie viele fehlerhaft erkannte Nachbarn ein einzelner Knoten in seinen Tabellen hat.

Die Abbildung 6.2 zeigt den Zusammenhang zwischen Fehlerrate und der Bewegungsgeschwin-

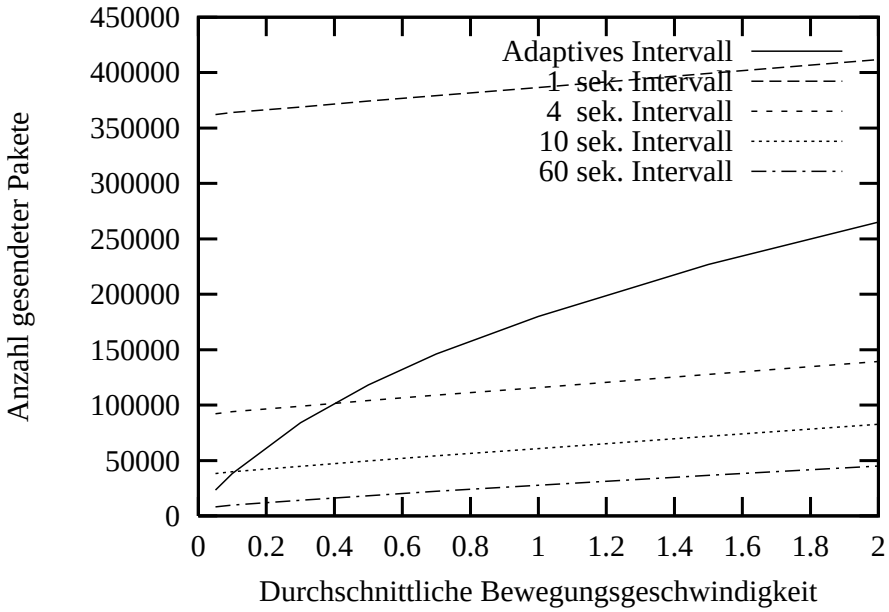


Abbildung 6.3: Anzahl gesendeter Pakete

digkeit bei allen konstanten Signalisierungsalgorithmen. Der adaptive Algorithmus versucht, die Fehlerrate unter einem bestimmten Limit zu halten. Das Bild zeigt, dass dieses Ziel erreicht wird, wenn dazu auch bei höheren Geschwindigkeiten immer mehr Signale nötig sind. Diese Eigenschaft lässt sich in Bild 6.3 ablesen. Hier ist der Bandbreitenbedarf durch die Anzahl der gesendeten Signale in Abhängigkeit von der Bewegungsgeschwindigkeit aufgetragen. Dazu werden alle Signale gezählt, die im Verlauf der Simulation ausgesendet wurden.

Der Bandbreitenbedarf von konstanten Signalisierungsalgorithmen ist unabhängig von deren Bewegungsgeschwindigkeit. Trotzdem ist in Bild 6.3 ein leichter Anstieg des Bedarfs bei ansteigender Bewegungsgeschwindigkeit zu erkennen. Dies wird durch die sofortige Aussendung einer Antwort an einen neuen Nachbarn verursacht. Je schneller sich die Knoten bewegen, um so mehr Antworten müssen gesendet werden. Der Bandbreitenbedarf des adaptiven Algorithmus steigt merklich mit der Bewegungsgeschwindigkeit. Er bleibt aber auf einem akzeptablen Niveau, wenn er mit dem Bedarf der konstanten Algorithmen mit ähnlicher Fehlerrate verglichen wird.

Der adaptive Algorithmus wurde auch in Netzwerken mit stark voneinander abweichenden Bewegungsgeschwindigkeiten getestet. In einem Netz mit gleichmäßig verteilten Geschwindigkeiten zwischen 0 und 2 m/s wurde eine Fehlerrate von 0.38 erreicht, wobei ungefähr 200000 Signale versendet wurden. Ein weiterer Test, bei dem sich die meisten Stationen nur langsam bewegen und nur ein Anteil von 10 % sich mit hoher Geschwindigkeit bewegt, hat die gleiche Fehlerrate, aber eine Aussendung von 130000 Signalen ergeben.

Die Ergebnisse der Simulationen zeigen deutlich, dass ein adaptiver Algorithmus für die Topologieerkennung gut geeignet ist. Durch die Adaption kann die Fehlerrate der Nachbarschaftser-

kennung reduziert werden. Die Adaption verbraucht dann zwar eventuell mehr Bandbreite, dies ist aber auch bei anderen Algorithmen nicht zu vermeiden. Die Anpassung der Signalrate an die Bewegungsgeschwindigkeit ist dagegen ein Vorteil bei Netzwerken, die wenige Veränderungen erfahren. Sie sparen wertvolle Bandbreite ein und können diese für zusätzliche Kommunikationsaufgaben nutzen.

6.3 Lokales Routing

Bereits in Kapitel 3.3 wurde eine Klassifizierung eingeführt, die Ad-hoc Routingalgorithmen in reaktive und proaktive Varianten unterteilt. Bei der Vorstellung der Eigenschaften von reaktiven Algorithmen wurde erläutert, dass diese Algorithmenklasse nicht die kürzesten Pfade nutzt. Dies ist durch das Suchen der Pfade per Fluten begründet, das nicht die Distanzen zwischen den Knoten misst, sondern die schnellste Botschaft bevorzugt. Diese Methode der Pfadermittlung verhindert auch, dass Pfade nachträglich optimiert werden. Ein neuer Pfad erfordert jedes Mal ein neues Suchen per Fluten. Der Verzicht auf das Suchen der kürzesten Pfade und auf jede Art von Pfadoptimierung bringt den reaktiven Algorithmen Vorteile hinsichtlich der benötigten Bandbreite ein. Dieser Abschnitt widmet sich der Frage, ob auch ein proaktiver Algorithmus durch einen Verzicht auf einige Pfadoptimierungen und die Nutzung von funktionierenden (statt kürzesten) Pfaden ähnliche Einsparungen erzielen kann.

Es erfordert relativ viel Netzwerkkapazität, um ständig neue Pfade zu berechnen. Die für die Nutzer verfügbare Bandbreite wird dabei zwangsweise reduziert. Besonders Netzwerke mit ohnehin niedriger Bandbreite und häufigen Veränderungen werden dabei stark belastet. In solchen Netzwerken ist es empfehlenswert, auf die Optimierung von Pfaden zu verzichten und eine Berechnung nur dann einzuleiten, wenn neue Pfade benötigt werden oder unbrauchbare Pfade repariert werden müssen.

In einem Netzwerk mit häufigen Veränderungen ist die Lebensdauer eines Pfades oft nicht groß genug, damit sich eine Optimierung lohnt. Die Kosten für die Optimierung eines Pfades sind bei geringer Lebensdauer höher als die tatsächliche Einsparung durch die spätere Nutzung des verkürzten Pfades.

Es wurden bereits einige Versuche unternommen, um die benötigte Bandbreite für einen Algorithmus zu reduzieren. Ein Versuch war der Fisheye-Algorithmus [GPH00], bei dem die Weitergabe von Update-Botschaften mit dem Abstand vom Sender immer stärker verzögert wird. Der Star-Algorithmus [GLAS01] ist ein Link-State Routing, das ebenfalls eine vom Nutzer einstellbare Optimierung für die Pfade enthält. Ein anderer Ansatz zur Einsparung von Bandbreite kommt von C.K. Toh [Toh97b], der mit bedarfsorientiertem Routing die langlebigsten Pfade belegen möchte, indem Stationen mit geringer Mobilität bevorteilt werden. In diesem Ansatz verkehrt sich die obige Effizienzbetrachtung ins Gegenteil, denn bedarfsorientiertes Routing führt im Normalfall keine Optimierung von Routen durch, es ist also bereits sehr effizient. Versucht man nun besonders stabile Pfade zu finden, muss die Mobilität der Stationen gemessen werden, wofür wieder Bandbreite verbraucht wird. Die Suche ist also nur dann erfolgreich,

wenn besonders langsame Stationen zu finden sind.

6.3.1 Neue Updatekriterien

Um die möglichen Einsparungen durch Verzicht auf Pfadoptimierung zu ermitteln, wird der in 5.3.3 beschriebene TERA modifiziert. Durch die Änderungen erhält der Algorithmus die Fähigkeit, eine Pfadoptimierung vorzeitig zu stoppen. Anschließend wird der ursprüngliche TERA mit dem modifizierten Algorithmus durch Simulationen verglichen.

Die notwendigen Änderungen sind gering, der erweiterte Algorithmus muss nun zwischen dem an die Nachbarn gemeldeten Baum und dem aktuellen Baum unterscheiden. Dies geschieht in der folgenden Beschreibung durch den Iterationsindex t. Der aktuelle Baum wird in $D_i^k(t)$, $V_i^k(t)$ und $N_i^k(t)$ abgelegt. Der an die Nachbarn gemeldete Baum ist in $D_i^k(t-1)$, $V_i^k(t-1)$ und $N_i^k(t-1)$ gespeichert. Ein Update wird erst dann an die Nachbarn weitergemeldet, wenn die Veränderungen im Baum so groß sind, dass sie einen festgelegten Grenzwert übersteigen.

In der ursprünglichen Algorithmusbeschreibung in Kapitel 5.3.3 wurden die Tabellen $D_i^k(t)$ and $V_i^k(t)$ jedesmal an die Nachbarn weitergeleitet, wenn eine der Bedingungen zutrifft:

$$D_i^k(t) \neq D_i^k(t-1)\ \exists\, i \quad \vee \quad N_i^k(t) \neq N_i^k(t-1)\ \exists\, i \quad \vee \quad V_i^k(t) \neq V_i^k(t-1)\ \exists\, i \tag{6.5}$$

Die Anzahl der übertragenen Botschaften kann reduziert werden, indem die vorliegenden Kriterien so geändert werden, dass nicht mehr jede kleine Änderung an die Nachbarn weitergeleitet wird. Es wird so ein Grenzwert ΔD bestimmt, der die maximal tolerierbare Veränderung definiert, die unterdrückt werden darf. Das Kriterium bezüglich der Distanz wird dann zu:

$$\frac{\mid D_i^k(t) - D_i^k(t-1) \mid}{D_i^k(t)} > \Delta D\ \ \exists i \quad \vee \quad N_i^k(t) \neq N_i^k(t-1)\ \ \exists i \tag{6.6}$$

Die Kriterien für den Nachfolger $N_i^k(t) \neq N_i^k(t-1)\ \exists i$ bleiben gleich, da jede Topologieänderung zwischen Nachbarn so wichtig ist, dass sie nicht ignoriert werden darf. Die Bedingung garantiert auch, dass neue oder entfallene Ziele sofort an die Nachbarn weitergemeldet werden.

6.3.2 Simulationsergebnisse

Zur Erprobung der vorgeschlagenen Verfahren wurden Simulationen durchgeführt, die mögliche Einsparungen und deren Auswirkungen auf die Erreichbarkeit auswerten. Die Simulationen verwenden 100 Knoten, die zufällig auf eine rechteckige Fläche von 1000×1000 Metern verteilt sind. Alle Knoten bewegen sich auf der Fläche und besitzen eine einheitliche Sendereichweite von 125 oder 180 Metern. Die Bewegungsgeschwindigkeit liegt zwischen 0.1 und 3 Metern pro Sekunde, was bis zu 10 Stundenkilometern entspricht.

Dieses Szenario entspricht beispielsweise einem Firmengelände, auf dem Personen tätig sind, die gelegentlich ihre Position verändern und andauernd miteinander kommunizieren müssen. Für die hohen Bewegungsgeschwindigkeiten sind beispielsweise Fahrzeuge für den Personentransport verantwortlich, oder es existieren bewegliche Maschinen wie z.B. Transportroboter, die über das Netzwerk gesteuert werden. Alle Personen arbeiten zusammen, dies erlaubt den Einsatz kooperativer Netzwerke zur Organisation der Personen und zur Steuerung von Maschinen.

Die Resultate der Simulationen sind in Tabelle 6.2 aufgelistet. Die Simulation zählt die Gesamtzahl der Pakete, die für das Routing notwendig waren einschließlich der Pakete für Beaconing („Hallo"-Pakete), um Nachbarn zu entdecken. Jede Simulation betrachtet eine Stunde, in der sich die Knoten mit der jeweils angegebenen Geschwindigkeit bewegen und das Netzwerk betreiben.

Um die Erreichbarkeit zu prüfen, werden in dieser Zeit andauernd zufällige Knoten als Ausgangspunkt und Ziel einer Route ausgewählt und die Funktionsfähigkeit der Route geprüft. Der Anteil an funktionsfähigen Routen ist jeweils im unteren Teil der Tabellen angegeben.

Die zweite Spalte zeigt die Ergebnisse, die bei vollständiger Optimierung der Routen erzielt werden. Dabei wird der Algorithmus aus Abschnitt 6.3.1 mit den Kriterien 6.5 eingesetzt. Alle Standard Distanz-Vektor Algorithmen benötigen diese Anzahl von Botschaften zur Routenberechnung.

Die Ergebnisse für $\Delta D > 0$ sind in den folgenden Spalten dargestellt und zeigen deutlich den Effekt der reduzierten Routenoptimierung durch die in Abschnitt 6.3.1 beschriebenen Kriterien 6.6. Beispielsweise sinkt bei der Bewegungsgeschwindigkeit von 3 m/s und einer Reichweite von 125 Metern die Anzahl von Botschaften von 8426375 für die volle Optimierung für $\Delta D = 0.5$ auf 4546803. Es ist somit möglich, die Anzahl der Botschaften fast zu halbieren. Dabei werden die Pfade maximal 50 Prozent länger, allerdings tritt dieser Fall praktisch nie ein. Die Länge der Pfade wurde ebenfalls untersucht, dabei ergibt sich, dass *alle* Simulationen mit einer Reichweite von 125 Metern eine durchschnittliche Pfadlänge von circa 5 Hops aufweisen und die Simulationen mit 180 Metern Reichweite etwa 3.2 Hops benötigen. Eine Verlängerung der Pfade durch Verzicht auf Pfadoptimierung ist in den Simulationen nicht feststellbar.

Die volle Pfadoptimierung zeigt nur einen kleinen Vorteil in den Simulationen, denn die Anzahl der erreichbaren Ziele sinkt um etwa 3 Prozent, wenn die Pfadoptimierung eingeschränkt wird. Die geringfügig reduzierte Erreichbarkeit ist ein Nebeneffekt der eingesparten Botschaften und liegt nicht an mangelhaftem Routing. Bei der vollen Optimierung werden vergleichsweise mehr Botschaften versandt. Damit ist die Wahrscheinlichkeit, einen Nachbarn früher zu entdecken etwas höher, da er öfter Botschaften aussendet. Da beide Routingalgorithmen neue Ziele gleich behandeln, profitiert der Algorithmus mit voller Optimierung hier von der schnelleren Nachbardetektion.

Die Simulationen zeigen allerdings auch, dass nur begrenzte Einsparungen möglich sind. Es gibt nur einen geringen Unterschied in der Anzahl der Botschaften zwischen $\Delta D = 0.5$ und $\Delta D = 0.7$. Es lassen sich hier keine Botschaften mehr einsparen, da das Kriterium für den

Sendereichweite 125 Meter

Geschw.	Anzahl der Botschaften			
	Volle Opt.	$\Delta D = 0.3$	$\Delta D = 0.5$	$\Delta D = 0.7$
0.1	299911	192395	168302	167495
0.5	1550582	958484	848312	857040
1.0	3038093	1879099	1667562	1673406
2.0	5925305	3638033	3211799	3204918
3.0	8426375	5112987	4546803	4559186
Geschw.	Erreichbare Ziele			
0.1	77.6%	76.0%	75.1%	74.9%
0.5	78.7%	76.8%	76.3%	76.3%
1.0	76.0%	74.2%	73.6%	73.7%
2.0	75.4%	73.2%	72.8%	72.7%
3.0	74.1%	72.0%	71.4%	71.4%

Sendereichweite 180 Meter

Geschw.	Anzahl der Botschaften			
	Volle Opt.	$\Delta D = 0.3$	$\Delta D = 0.5$	$\Delta D = 0.7$
0.1	487921	357846	321489	320580
0.5	2593525	1950157	1761412	1762888
1.0	5017479	3806848	3441140	3442982
2.0	9792599	7351917	6673566	6650802
Geschw.	Erreichbare Ziele			
0.1	96.28%	95.49%	95.01%	95.12%
0.5	95.12%	94.45%	94.13%	94.09%
1.0	94.42%	93.62%	93.38%	93.34%
2.0	93.5%6	92.74%	92.40%	92.44%

Tabelle 6.2: Simulationsergebnisse bei lokalem Routing

Nachfolger $N_i^k(t) \neq N_i^k(t-1)\ \exists i$ immer mehr Einfluss gewinnt. Mit diesem Kriterium wird gewährleistet, dass ein Netzwerk funktionsfähig bleibt. Eine weitere Erhöhung von ΔD kann die Anzahl der Botschaften nicht unter das von diesem Kriterium bestimmten Limit drücken.

6.4 Zusammenfassung

In diesem Kapitel wurden zwei Methoden zur besseren Ausnutzung der Funkbandbreite vorgeschlagen.

Die erste Methode zielt darauf ab, durch Anpassung der Senderate bei der Nachbarschaftserkennung Sendungen einzusparen. Dabei wird die (relative) Bewegungsgeschwindigkeit einer Station durch die Anzahl der auftretenden Verbindungswechsel abgeschätzt. Aus der so geschätzten Geschwindigkeit lässt sich eine Senderate errechnen, die eine vorgegebene durchschnittliche Fehlerrate einhält. Die adaptive Nachbarerkennung bringt so langsamen Stationen den Vorteil von wenigen Aussendungen und erlaubt ihnen damit z.B. ihre Batterien zu schonen. Für schnelle Stationen werden zwar viele Aussendungen ausgelöst, diese sind jedoch nötig, um die Topologie des Netzwerks besser erkennen zu können und somit ein funktionierendes Routing trotz schneller Bewegungen zu gewährleisten.

Die zweite Methode schlägt einen verbesserten Routingalgorithmus, den Tree Exchange Routing Algorithmus mit lokaler Arbeitsfähigkeit vor. Der Algorithmus kann statt kürzester Pfade auch nur funktionierende Pfade ermitteln und so einen erheblichen Teil der ansonsten zur Berechnung nötigen Botschaften einsparen. Die Intensität, mit der sich der Algorithmus um eine Pfadoptimierung bemühen soll, ist durch einen Parameter einstellbar. Der Algorithmus wird anschließend mit verschiedenen Intensitäten und Bewegungsgeschwindigkeiten durch Simulation getestet. Dabei zeigt sich, dass der Verzicht auf die Pfadoptimierung kaum Einschränkungen in der Erreichbarkeit, aber drastische Einsparung in der Anzahl der Botschaften bewirkt.

Kapitel 7

Mehrwege-Routing

7.1 Einleitung

Im folgenden Kapitel wird das Mehrwege-Routing vorgestellt, mit dem ein Routingalgorithmus auch mehrere Routen zwischen zwei Knoten eines Netzwerks berechnen kann.

Dafür wird zuerst das Konzept des Mehrwege-Routings vorgestellt, das aus den von Nachbarn gesendeten Informationen zusätzliche Routen errechnet. Werden so mehrere Pfade zu einem Ziel entdeckt, dann versucht der Algorithmus durch eine gleichmäßige Verteilung der Datenpakete auf die Pfade eine möglichst ausgeglichene Belastung des Netzwerks zu erreichen.

Die durch Mehrwege-Routing erzielbare Lastverteilung wird durch eine Reihe von Simulationen genauer untersucht. Anhand eines statischen Netzwerks wird ein Beispiel für eine optimale Lastverteilung erläutert. Anschließend wird ein wesentlich realistischeres Modell mit dynamischen Netzwerken behandelt. Für diese Netzwerke wird die Spitzenbelastung einzelner Knoten und die Verteilung der Belastung im Netzwerk untersucht.

Ein weiterer Abschnitt widmet sich dem Problem inkonsistenter Routingtabellen, die in Ad-hoc Netzwerken durch unterschiedliche Laufzeiten von Routing-Botschaften immer wieder auftreten können. Es wird ein modifiziertes Mehrwege-Routing vorgestellt, mit dem die Auswirkungen von inkonsistenten Tabellen auf die Erreichbarkeit deutlich reduziert werden.

Das Kapitel endet mit einer Analyse der Ausfallzeiten in Ad-hoc Netzwerken. Dabei wird besonders die Dauer und Häufigkeit der Ausfälle und deren Korrigierbarkeit betrachtet.

Die in diesem Kapitel vorgestellten Simulationen entstanden im Rahmen einer Diplomarbeit [Küh00].

7.2 Routing über parallele Pfade

Die bisher erläuterten Routingverfahren waren immer darauf ausgelegt, den kürzesten Pfad zwischen zwei Knoten zu berechnen, und nur dieser wurde anschließend für die Kommunikation zwischen den Knoten genutzt. Mit steigender Vermaschung eines Netzwerks (siehe auch Kapitel 4.3.1) stehen mehrere, oftmals sogar gleich lange Pfade im Netzwerk zur Verfügung. Beim Mehrwege-Routing (engl. multipath routing) werden auch mehrfache Routen zwischen zwei beliebigen Netzknoten ermittelt, d.h. es stehen neben der ursprünglichen kürzesten Route von Knoten A nach Knoten B noch andere Alternativ-Routen zur Verfügung.

Zur Nutzung von parallelen Routen wird die Routingtabelle so erweitert, dass für jedes Ziel mehrere Nachbarn als Vermittler eingetragen werden können. Zusätzlich wird ein Parameter eingeführt, der die Verteilung der ausgehenden Pakete auf die verschiedenen Routen steuert. Mit diesem Parameter ist für jede Route eine Wahrscheinlichkeit festgelegt, die bestimmt, wie häufig eine Route genutzt wird. Dieser Parameter errechnet sich in der Regel aus den Kosten für die jeweilige Route, z.B. der Länge des entsprechenden Pfades, dem Durchsatz einer Verbindung oder vergleichbaren Informationen. Zur besseren Veranschaulichung ist in Tabelle 7.1 eine so erweiterte Routingtabelle des Knotens E für das in Abbildung 7.1 dargestellte Netzwerk aufgeführt.

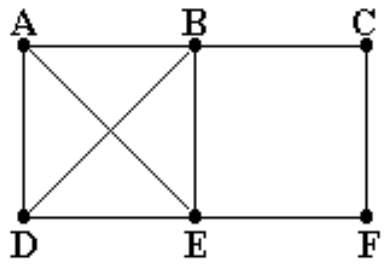


Abbildung 7.1: Beispielnetzwerk

Ziel	1. Pfad		2. Pfad		3. Pfad	
A	A	0.5	B	0.25	D	0,25
B	B	0,5	A	0,25	D	0,25
C	B	0,375	F	0,375	A	0,25
D	D	0,5	A	0,25	B	0,25
E	-	-	-	-	-	-
F	F	0,63	B	0,21	A	0,16

Tabelle 7.1: Routing-Tabelle des Knotens E

Wenn Knoten E ein Paket an Knoten A senden muss, so zieht er zuerst eine Zufallszahl zwischen 0 und 1. Je nach Wert der Zufallszahl wird dann die entsprechende Ausgangsleitung gewählt. Ist die Zufallszahl kleiner oder gleich 0.5, so wird das Paket direkt an Rechner A gesendet. Liegt sie zwischen 0.5 und 0.75, wird das Paket über Knoten B an A gesendet, und bei einer

Zahl zwischen 0.75 und 1 wird es über Knoten D an A gesendet. Die gezogenen Zufallszahlen müssen gleichverteilt sein, dann geben auch die in der Routingtabelle eingetragenen Zahlen genau die Wahrscheinlichkeit an, mit der eine bestimmte Leitung für ein bestimmtes Ziel benutzt wird.

Die Berechnung der Routingtabelle kann beim Mehrwege-Routing durch mehrfache Anwendung des bereits vom Shortest-Path Routing bekannten Algorithmus erfolgen. Dazu wird zuerst der kürzeste Pfad zu einem Ziel ermittelt und diese Route in die erste Spalte der Tabelle eingetragen. Dann wird diese Route (d.h. sämtliche Knoten und Kanten des Pfades mit Ausnahme des Start- und Zielknotens) aus dem Graphen gestrichen. Danach wird erneut der kürzeste Pfad errechnet, der jetzt in die zweite Spalte der Tabelle eingetragen wird. Dies wird solange wiederholt, bis entweder keine Routen mehr zu finden sind oder die Tabelle keinen weiteren Einträge mehr erlaubt. So werden vollkommen unabhängige Pfade errechnet. Diese Vorgehensweise zum Aufbau der Routingtabellen ist natürlich sehr rechenintensiv, da für jedes Ziel nicht nur eine, sondern n Routen berechnet werden müssen, weshalb diese Art des Mehrwege-Routings für den Einsatz in hochdynamischen Netzen nicht besonders geeignet ist.

In Netzwerken mit beschränkter Funkkapazität hat Mehrwege-Routing viele Vorteile. Beispielsweise verteilt sich bei parallelen Routen das Verkehrsaufkommen auf mehrere Wege. Damit werden einzelne Stationen weniger belastet und der mögliche Gesamtdurchsatz kann mit jeder zusätzlichen Route gesteigert werden. Die Effizienz des Mehrwege-Routings hängt natürlich davon ab, wie viele verschiedene Wege es in einem Netzwerk vom Start- zum Zielknoten gibt. Je vermaschter ein Netzwerk ist, um so mehr verschiedene Routen stehen zur Verfügung und um so effizienter arbeitet das Routing-Verfahren.

Die folgenden Untersuchungen zeigen auch, dass sich die Zuverlässigkeit einer Verbindung durch Mehrwege-Routing im begrenzten Umfang steigern lässt. Sind n unabhängige Pfade zu einem Ziel verfügbar, dann können $n-1$ Pfade ausfallen, bevor der Zielknoten nicht mehr erreichbar ist. In der Praxis bedeutet der Ausfall einer Route zu einem bestimmten Ziel dann keinen vollständigen Verbindungsverlust mehr. Allerdings gehen die Pakete verloren, die durch die Zufallsentscheidung dem defekten Pfad zugeteilt werden.

7.3 Mehrwege-Routing für Pfadsuchealgorithmen

Im folgenden Abschnitt wird eine Erweiterung der Pfadsuche vorgestellt, die Mehrwege-Routing ermöglicht. Dazu werden zuerst die notwendigen Änderungen an den Tabellen beschrieben und anschließend die erreichbare Lastverteilung durch Simulationen ermittelt.

Die beiden Pfadsuche-Algorithmenvarianten, deren Funktionsweise bereits in Kapitel 5 erläutert wurde, sollen nun um ein Mehrwege-Routing erweitert werden. Beide Algorithmen verwenden die gleichen Tabellen, um die von den Nachbarn gesendeten Informationen zu sammeln und liefern nach der Berechnung ihre Resultate in der Form dreier Tabellen zurück:

- Die Routingtabelle D_i^k, die für jedes Ziel i die bestmögliche Distanz angibt.

– Die Nachfolgertabelle N_i^k, die beschreibt, über welchen Nachbarn das Ziel zu erreichen ist.

– Die Vorgängertabelle V_i^k, die den jeweils vorletzten Knoten im Pfad angibt.

Zur Implementierung des Mehrwege-Routing werden die Routingtabellen um einen weiteren Eintrag pro Ziel ergänzt, in dem jeweils die Kennung eines Nachbarn abgelegt wird, der eine Alternativ-Route zum Ziel anbietet.

Die zusätzliche Tabelle wird im Folgenden zweite Routingtabelle genannt:

– Die Alternativ-Nachfolgertabelle AN_i^k beschreibt, über welchen alternativen Nachbarn ein Ziel zu erreichen ist.

Jedes Mal, wenn eine neue Verbindung entsteht, eine alte verloren geht oder ein Paket empfangen wird, das von Verbindungsänderungen berichtet, wird die komplette erste Routingtabelle des Knotens (TERA) bzw. einzelne Zeilen dieser Tabelle (einfache Pfadsuche) neu berechnet. Sofort danach wird eine weitere Funktion aufgerufen, die nach alternativen Routen sucht und das Ergebnis in AN_i^k speichert. Dazu tauscht die neue Funktion keine Botschaften mit den Nachbarn aus. Die Funktion errechnet nur aus den bereits vorhandenen Tabellen $ND_{i\,j}^k$ und $NV_{i\,j}^k$ für jedes erreichbare Ziel eine Alternativroute: Ziele, die in der ersten Routentabelle schon unerreichbar sind, werden auch in der zweiten Tabelle als unerreichbar gekennzeichnet.

$$AN_i^k = \text{”}none\text{”}\ \forall\ i \mid N_i^k = \text{”}none\text{”}$$

Wenn Knoten A laut erster Routingtabelle erreichbar ist, werden die Tabellen aller Nachbarn der Reihe nach betrachtet und es wird überprüft, ob irgend ein Nachbar, außer dem in der ersten Routingtabelle bereits aufgeführten, eine Route zum Knoten A anbietet. Wird ein solcher Nachbar gefunden, so wird als nächstes die Distanz, die dieser Nachbar zum Knoten A laut seiner eigenen Routingtabelle hat, betrachtet. Nur wenn diese Distanz zuzüglich der einen Kante zum Nachbarn gleich der in der eigenen ersten Routingtabelle eingetragenen Distanz zum Nachbarn A ist, handelt es sich um einen Alternativeintrag, der in die zweite Routingtabelle übernommen wird. In der Tabelle $ND_{i\,j}^k$ ist die Distanz zum Nachbarn bereits eingerechnet und kann direkt verwendet werden.

$$AN_i^k = \begin{cases} j & \exists\ j \neq N_i^k\ \wedge\ ND_{i\,j}^k\ =\ D_i^k \\ \text{”}none\text{”} & sonst \end{cases}$$

Die Berechnungsfunktion durchsucht so für jede existierende erste Route die Tabellen der Nachbarinformation, um eine zweite Route zu finden. Wird eine solche Route gefunden, so wird der Nachbar, der sie anbietet, in der zweiten Routingtabelle als alternativer Nachfolger für A eingetragen. Wird kein Nachbar gefunden, der die Kriterien erfüllt, so wird in der zweiten Tabelle das Ziel als unerreichbar eingetragen.

Für die Alternativrouten sind in der vorliegenden Spezifikation nur gleiche Distanzen möglich. Diese sehr restriktive Beschränkung verhindert aber mögliche Schleifen, die durch ungünstig gewählte Alternativrouten der Nachbarn entstehen können. Die Distanzen müssen gleich sein, da es keine kürzeren Pfade als die bereits in der ersten Tabelle ermittelten geben kann und längere Pfade mit hoher Wahrscheinlichkeit zu Schleifen führen. Eine Alternativroute darf nicht länger als die bisher ermittelte kürzeste Route sein, denn sonst wird ein Nachbar, der ebenfalls alternative Routen berechnet und den kürzesten Pfad bevorzugt, ein so erhaltenes Paket mit hoher Wahrscheinlichkeit zurücksenden. Dies wäre ihm erlaubt, weil er nun ja ebenfalls längere Pfade für seine Alternativrouten nutzen darf.

Für die Berechnung der Alternativrouten werden lediglich die Routingtabellen der Nachbarn benötigt. Da diese zur Ermittlung der ersten Routingtabelle bereits in den Knoten abgespeichert wurden, müssen keine zusätzlichen Informationen zwischen den Knoten des Netzwerks ausgetauscht werden, um dieses Mehrwege-Routingverfahren zu verwenden. Die zweite Routingtabelle wird also mit äußerst geringem Rechenaufwand quasi gratis mitgeliefert.

Beim Mehrwege-Routing wird vor dem Senden eines Pakets geprüft, ob eine oder zwei Routen zum Ziel vorhanden sind. Ist nur eine Route verfügbar, dann wird das Paket an den in der Tabelle N_i^k angegebenen Nachbarn weitergeleitet. Bei zwei Routen zieht der Knoten zuerst eine gleichverteilte Zufallszahl X zwischen Null und Eins. Ist X kleiner als 0.5, so verwendet der Knoten die in der ersten Routingtabelle abgelegte Ausgangsleitung. Ist X größer oder gleich 0.5, so wird der Eintrag in AN_i^k zum Senden des Paketes verwendet. Damit wird nur in den Fällen, in denen eine Alternativroute vorhanden ist, zufällig eine der Routen ausgewählt. Ansonsten wird einfach wie beim Routing ohne Mehrwege-Funktionalität die erste Route benutzt. Auf diese Weise können keine bisher erreichbaren Ziele unerreichbar werden. Die Benutzung des Mehrwege-Routings hat also keinerlei negative Auswirkungen bezüglich der Erreichbarkeit von Knoten.

7.3.1 Simulation der Lastverteilung

Das Mehrwege-Routing erlaubt eine Verteilung der Pakete auf parallele Wege. Die Verteilung ist in der vorliegenden Spezifikation einfach durch einen Zufallsgenerator gesteuert. Dass sich damit dennoch eine sehr gleichmäßige Verteilung erzielen lässt, wird am folgenden Beispiel demonstriert, das zur besseren Darstellung der Lastverteilung ein statisches Netzwerk verwendet. Zusätzlich wird durch den Verzicht auf die Knotenmobilität die indirekte Lastverteilung durch Positionswechsel der Knoten verhindert.

In Netzwerken, in denen sich die Knoten schnell bewegen, entsteht schon allein durch diese Bewegung eine gewisse Verteilung der Last, da zentrale Knoten, die einen Hauptteil der Kommunikation tragen müssen, nicht lange an ihrem Ort verweilen und im nächsten Moment für die Aufrechterhaltung der Kommunikation völlig unwichtig sein können, während andere Knoten ihren zentralen Platz einnehmen. Eine Untersuchung der Lastverteilung bei Mehrwege-Routing in dynamischen Netzwerken wird im zweiten Teil dieses Unterkapitels durchgeführt, zunächst wird jedoch der statische Fall behandelt.

Lastverteilung im statischen Netzwerk

Kommunizieren in einem statischen Netzwerk zwei Knoten miteinander und stammt die dazu notwendige Route von einem Algorithmus, der kürzeste Pfade berechnet, dann wird nur eine Route berechnet und anschließend für die gesamte Dauer der Kommunikation genutzt. Dieser Fall ist für die Rechner, die auf der Route zwischen den zwei Kommunikationspartnern liegen, ziemlich unbefriedigend, da sie das volle Datenaufkommen weiterleiten und somit einen Großteil ihrer Ressourcen zur Verfügung stellen müssen, während die anderen Rechner des Netzwerks gar nicht an der Kommunikation teilnehmen.

Um diese Lastverteilung zu verdeutlichen, wurden Vergleichssimulationen mit und ohne Mehrwege-Routing auf einem statischen Netzwerk durchgeführt. Das verwendete Beispielnetz hat eine Gitterstruktur, wie sie in Abbildung 7.2 dargestellt ist. Es besteht aus 100 Knoten, die in einem Quadrat von 10 mal 10 Stationen angeordnet sind. Diese Topologie ist für einen Test der Lastverteilung besonders gut geeignet, da es viele, gleich lange Wege von einem Start- zu einem Zielknoten gibt.

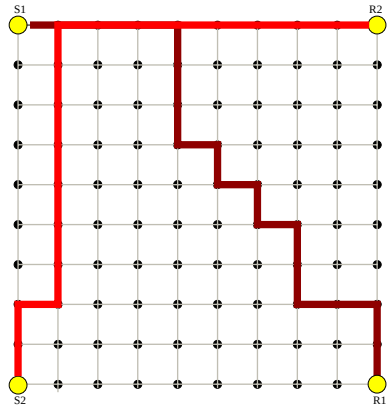


Abbildung 7.2: Netztopologie eines 10x10-Gitters

Obwohl der Simulator für dynamische Netzwerke entwickelt wurde, kann er auch statische Netze simulieren. Dafür muss lediglich eine Netzdatei erzeugt werden, in der zu Beginn sämtliche Verbindungen aufgebaut werden und an denen dann bis zum Laufzeitende keinerlei Änderungen mehr vorgenommen werden.

Zum Test der Lastverteilung werden zwei Datenströme im Netzwerk gesendet. Die Sender, bzw. Empfänger liegen dabei jeweils auf einer Ecke des Quadrats und senden die Daten an den Empfänger auf der gegenüberliegenden Seite (im Bild S1-R1, S2-R2). Bei dieser Anordnung müssen sich die Datenströme im Netzwerk kreuzen. An dieser Kreuzungsstelle ist der Engpass bei der Übertragung zu erwarten.

Im Verlauf der Simulation werden über jeden der beiden Datenströme 1000 Pakete gesendet. Dabei wird jede Station überwacht und es wird mitgezählt, wie viele Pakete über die Stationen übertragen werden. Die Anzahl der Pakete dient als Belastungsmaß und wird in einer Graphik veranschaulicht.

Auf diese Weise entstand Abbildung 7.3, in der die Lastverteilung für das Netz ohne Mehrwege-Routing dargestellt ist. Die Grundfläche entspricht dabei dem Netzwerk, in dem die Knoten im Gitter angeordnet sind und nach oben ist das Belastungsmaß des jeweiligen Knotens angetragen. Der verwendete Pfadsuche-Routingalgorithmus hat zwischen den Knotenpaaren die Pfade bestimmt, der Verlauf dieser Pfade ist in Abbildung 7.2 durch dicke Linien markiert.

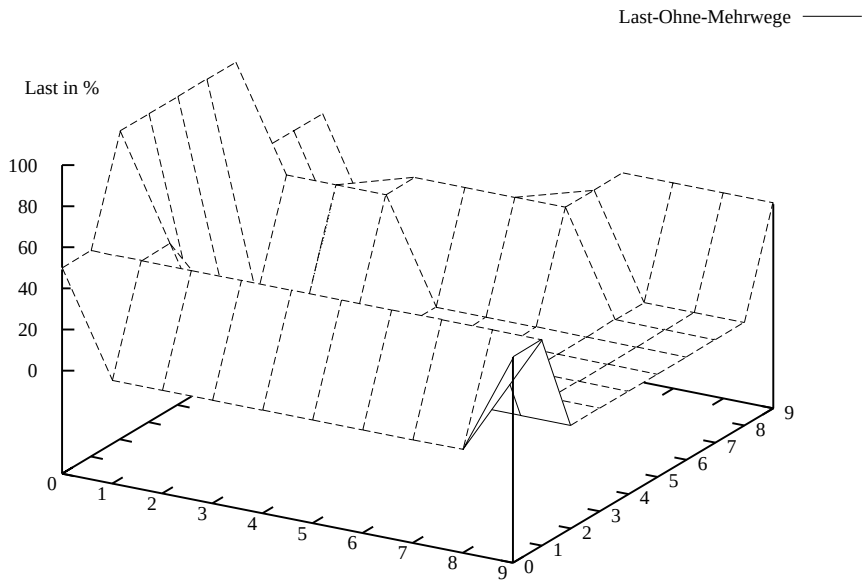


Abbildung 7.3: Lastverteilung in einem 10x10 Gitter ohne Mehrwege-Routing

Wie in Abbildung 7.3 erkennbar ist, findet keinerlei Lastverteilung statt. Alle Knoten auf einer Route von Eckpunkt zu Eckpunkt erhalten damit 50% der Gesamtlast des Netzwerks. Die beiden Routen überschneiden sich nicht nur in einem einzigen Knoten, sondern sie benutzen sogar 4 Kanten gemeinsam. Dadurch müssen die Knoten, die an diesen Kanten sitzen, das Datenaufkommen beider Verbindungen übertragen und erhalten damit die Spitzenbelastung von 100% der möglichen Last. Somit wird ein Großteil ihrer Übertragungs- und Rechenkapazitäten beansprucht, während die meisten Knoten des Netzwerks überhaupt keine Last tragen.

In Abbildung 7.4 ist dieselbe Simulation, diesmal jedoch mit Mehrwege-Routing, dargestellt. Dabei verteilt sich die Last gerecht auf die Knoten, die als Router agieren. Lediglich die Sende- und Empfangsknoten haben eine Last von 50%, da sie ja an der Hälfte der Übertragungen beteiligt sind, während alle anderen Knoten geringere Werte aufweisen. In statischen Netzwerken ist es deshalb sinnvoll, Mehrwege-Routing einzusetzen, da dadurch die Last gerechter verteilt wird und so einzelne Knoten nicht übermäßig beansprucht werden, während andere gar keine Arbeit verrichten müssen.

Der Vorteil der Lastverteilung wird beim erzielbaren Durchsatz besonders deutlich. Während bei der Variante ohne Mehrwege-Routing sogar 5 Knoten mit 100% ausgelastet waren, sind

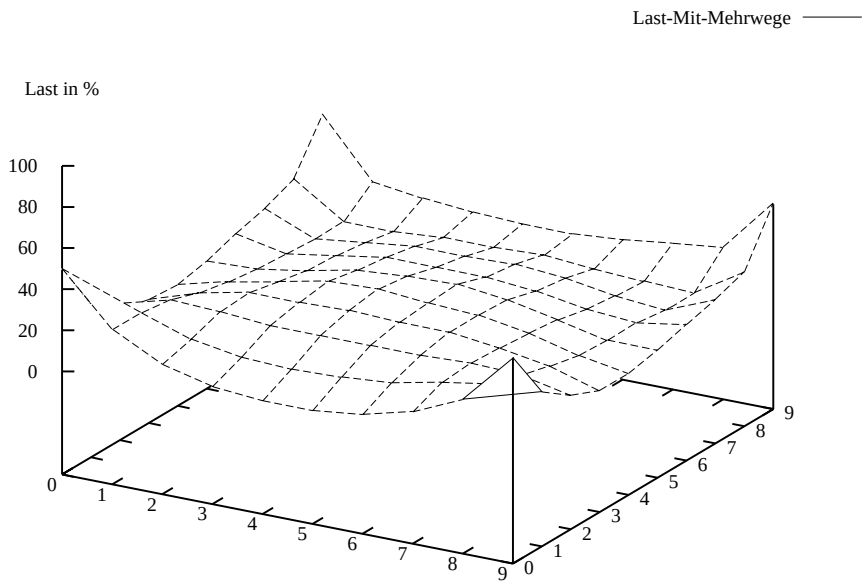


Abbildung 7.4: Lastverteilung in einem 10x10 Gitter mit Mehrwege-Routing

beim Mehrwege-Routing nur der Sender und der Empfänger mit jeweils 50% ausgelastet. Wird die Auslastung von 100% gleichzeitig als maximale Obergrenze für die Übertragungskapazität einer Station betrachtet, dann ist in diesem Beispiel mit Mehrwege-Routing eine Verdopplung des Durchsatzes für jeden Datenstrom möglich.

Lastverteilung in dynamischen Netzwerken

In bewegten, dynamischen Netzwerken mit vielen Topologieänderungen ist der Vorteil der Lastverteilung nicht so deutlich zu erkennen wie bei statischen Netzwerken. Allein schon durch die Bewegung der Knoten entsteht eine gewisse Lastverteilung. Außerdem verändert sich die Topologie des Netzwerks über die gesamte Laufzeit der Simulation, weshalb nicht nur ein statischer Ausschnitt des Netzwerks zu einem bestimmten Zeitpunkt betrachtet werden kann. Stattdessen muss das Verhalten des Netzwerks über die gesamte Laufzeit analysiert werden.

Die Simulation für dynamische Netzwerke untersucht die Lastverteilung im Netzwerk aus der Sicht von vier Knotenpaaren. Diese vier Knotenpaare legen die zu untersuchenden Kommunikationsbeziehungen fest, die entweder mit oder ohne Mehrwege-Routing arbeiten. Vor jeder Untersuchung wird eine Grundlast auf dem Netzwerk erzeugt, indem jeder Knoten an jeden anderen Knoten des Netzwerks jeweils ein Datenpaket versendet. Dabei wird für jeden Knoten die Anzahl, der über ihn laufenden Pakete, gezählt und als Maß für die Belastung gewertet.

Anschließend senden sich die zu Programmbeginn bestimmten Knotenpaare jeweils drei Da-

tenpakete zu, deren Route nachvollzogen wird. Dabei wird auf jedem der drei Wege der Knoten mit der höchsten Belastung ermittelt und dieser Wert in einer Liste abgespeichert. Anschließend werden sämtliche Belastungswerte der einzelnen Knoten wieder gelöscht, damit bei der nächsten Untersuchung das Netzwerk wieder mit einer neuen Grundlast belegt werden kann. Bei Programmende werden die in der Liste gespeicherten höchsten Belastungswerte analysiert.

Wie bereits erwähnt, muss in dynamischen Netzwerken die Lastverteilung über die gesamte Laufzeit und nicht nur zu einem einzelnen Zeitpunkt betrachtet werden. Deshalb wird das Netzwerk in jeder simulierten Sekunde erneut untersucht. Damit die Last auf den einzelnen Knoten des Netzwerks auch der augenblicklichen Situation angepasst ist, wird jedes Mal, bevor die Last untersucht wird, eine neue Grundlast erzeugt. Dazu kommuniziert jeder Knoten mit jedem Knoten, d.h. bei einem Netzwerk mit N Knoten werden $N \cdot (N-1)$ Pakete verschickt. Auf diese Weise entsteht eine möglichst gerechte Grundlasterzeugung durch alle Knoten.

Bei einer Datenübertragung über mehrere Teilstrecken bestimmt immer die Teilstrecke mit der geringsten Übertragungskapazität die Leistung der Gesamtübertragung. Daher wird bei der Bewertung einer Kommunikationsbeziehung die Route vom Sender zum Empfänger abgegangen und jeweils der Knoten mit der höchsten Last auf der Route abgespeichert. Dieser Knoten wird zur Bewertung herangezogen, da er die Gesamtübertragungskapazität der Route begrenzt.

Die Simulation untersucht immer vier Knotenpaare, damit sich Schwankungen und ungünstige Bewegungsmuster im Gesamtergebnis durch die Betrachtung mehrerer Knotenpaare wieder herausmitteln. Bei jeder Untersuchung der Last verschickt jedes dieser vier Knotenpaare jeweils 3 Pakete an seinen Partner, deren Routen nachvollzogen werden. Das Nachvollziehen der benutzten Routen muss mehrmals geschehen da nur so ein Unterschied zwischen der Nutzung einer Route und dem Mehrwege-Routing deutlich wird.

Die Simulationen verwenden folgende Parameter zur Netzwerksimulation:

- Ein Netzwerk mit 50 Knoten
- Sende- und Empfangsreichweite von 200 m
- Quadratisches Feld mit einer Fläche von 1000x1000 Metern
- 10 Simulationsläufe mit Bewegungsgeschwindigkeiten von 0,001 m/s bis 2,0 m/s
- Eine Stunde simulierte Netzwerkaktivität
- Adaptive Verbindungserkennung

Die Abbildung 7.5 zeigt die ermittelten Ergebnisse. Die Abbildung gibt den durchschnittlichen Belastungswert der am meisten belasteten Knoten auf den untersuchten Routen an. Durchschnittliche Belastung bedeutet dabei, dass jede simulierte Sekunde für alle vier Knotenpaare der Knoten mit dem Höchstwert auf ihrer Route ermittelt und abgespeichert wurde und bei Programmende aus diesen Werten der Durchschnitt errechnet wird.

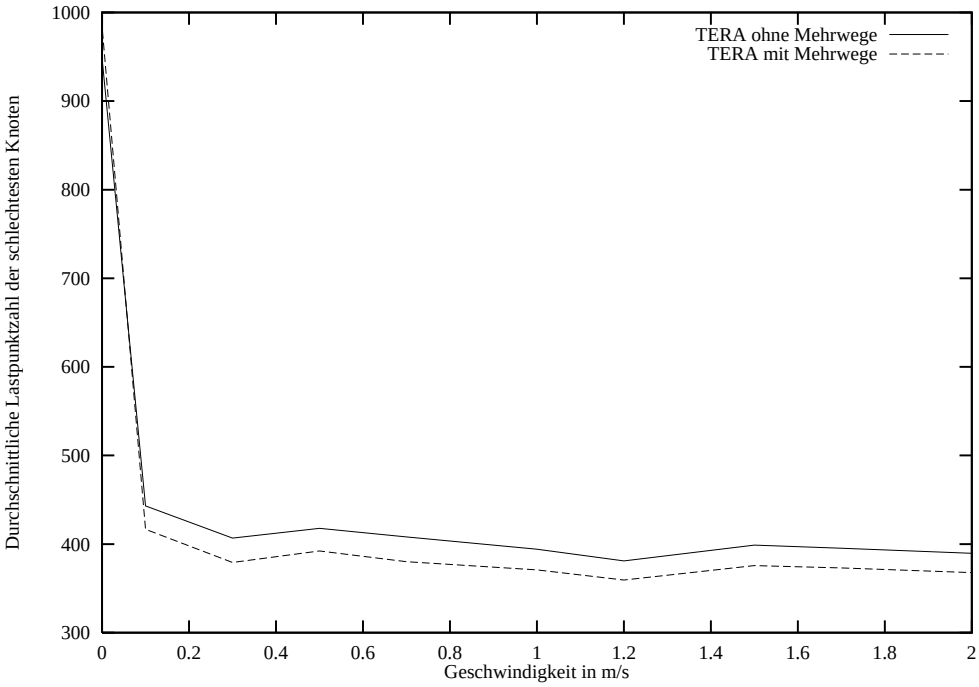


Abbildung 7.5: Durchschnittliche Belastung der begrenzenden Knoten

In Abbildung 7.5 sind die Ergebnisse der beiden Simulationsläufe dargestellt. Die durchschnittliche Belastung des schlechtesten Knoten mit Mehrwege-Routing ist durchgehend für alle Geschwindigkeiten geringer als ohne Mehrwege-Routing. Die Differenz beträgt dabei im Schnitt 24 Lastpunkte. Dies bedeutet, dass mit Mehrwege-Routing auf der schlechtesten Übertragungsstrecke einer Route, also dem „Nadelöhr“, ca. 6% Übertragungskapazitäten frei werden. Diese können anderen Verbindungen zur Verfügung gestellt werden, wodurch ein höherer Durchsatz erzielt wird. Die Last wird also durch Mehrwege-Routing auch in dynamischen Netzwerken, in denen viele Topologieänderungen stattfinden, besser verteilt.

Der besonders hohe Wert bei einer Knotengeschwindigkeit von 0,001 m/s ist durch das in diesem Fall quasi statische Netzwerk zu erklären. Dadurch liegen einige Knoten lange an zentralen Stellen des Netzwerks und dienen für viele Pfade als Router. Diese Knoten erhalten so extrem viele Pakete und sind dann auch das Nadelöhr im Netzwerk.

Um die gleichmäßige Verteilung der Last näher zu untersuchen, wird im nächsten Schritt die Standardabweichung der Knotenbelastung ermittelt. Dazu wird die oben beschriebene Simulation wiederholt, nur wird diesmal für verschiedene Bewegungsgeschwindigkeiten die Standardabweichung der Last der Knoten berechnet, die jeweils die höchste Lastpunktzahl auf einer untersuchten Route haben. Die so ermittelte Standardabweichung ist in Abbildung 7.6 zu sehen.

Wie die Ergebnisse in Abbildung 7.6 zeigen, ist die Standardabweichung der Last auf den untersuchten Routen mit Mehrwege-Routing fast durchgängig geringer als ohne Mehrwege-Routing. Dies bedeutet, dass mit Mehrwege-Routing die höchste Belastung eines Knotens weniger vom

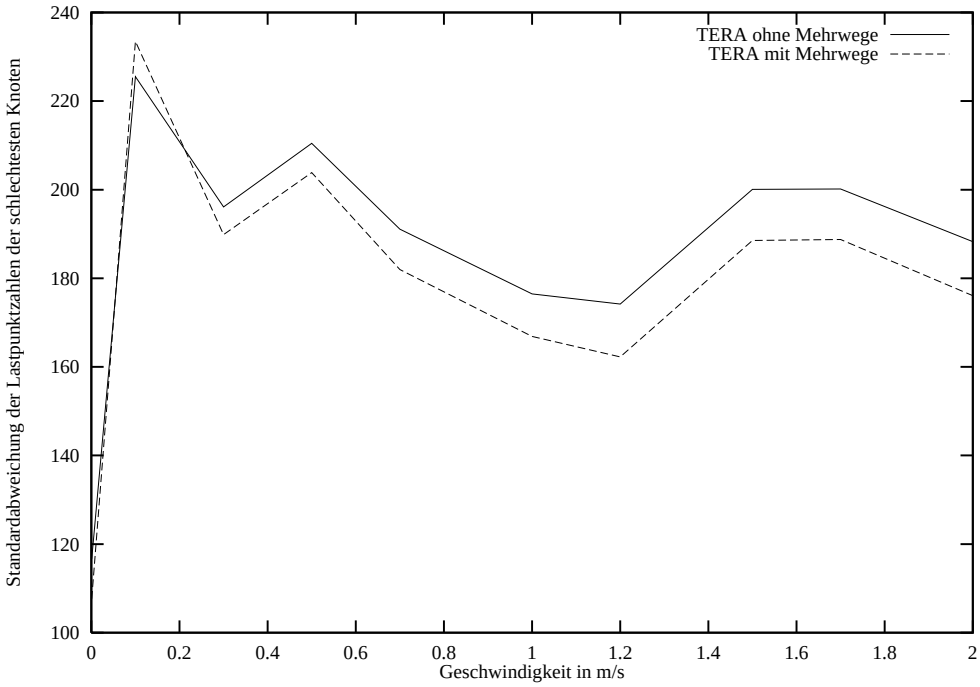


Abbildung 7.6: Standardabweichung der Belastung

Mittelwert der Belastungen abweicht. Die Last ist bei einer reduzierten Standardabweichung gleichmäßiger im Netzwerk verteilt, und es gibt weniger Knoten, die eine besonders hohe oder eine besonders niedrige Last haben.

Die Simulationen zeigen, dass durch Mehrwege-Routing sowohl in statischen als auch in dynamischen Netzwerken eine gleichmäßigere Lastverteilung erreicht werden kann. Dadurch wird das Datenaufkommen gerechter auf alle Knoten verteilt, so dass nicht mehr einige wenige Knoten den Hauptteil der Kommunikation übernehmen müssen, sondern alle Knoten gleichmäßig beteiligt werden. Auf diese Weise kann vermieden werden, dass einzelne, überlastete Knoten die Übertragungskapazität einer Verbindung herabsetzen.

7.4 Steigerung der Erreichbarkeit

Durch eine geringe Modifikation des oben beschriebenen Mehrwege-Routings ist es möglich, die Anzahl der Knoten, die sich in einem Netzwerk mit mobilen Knoten untereinander erreichen können, zu erhöhen. Die dazu notwendigen Änderungen am Mehrwege-Routing werden im folgenden vorgestellt, und anschließend wird durch eine Reihe von Simulationen die so erzielte Steigerung der Erreichbarkeit überprüft.

7.4.1 Funktionsprinzip

In dynamischen Netzwerken erhält ein Knoten von seinen Nachbarn gelegentlich inkonsistente oder sogar widersprüchliche Informationen. Wenn der Knoten aufgrund dieser Informationen seine Routingtabelle bzw. seinen minimal spannenden Baum neu aufbaut, kann er nicht entscheiden, welcher Nachbar ihm korrekte bzw. vollständige Informationen anbietet. Er muss sich also zufällig für einen Nachbarn entscheiden und dessen Routingtabelle in seinen Baum integrieren. Versucht der Knoten anschließend, die Routingtabelle des anderen Nachbarn ebenfalls zu integrieren, dann entstehen Widersprüche und die zusätzlichen Informationen werden nicht weiter genutzt. Die Widersprüche sind in der Regel nur temporär, da sie durch unterschiedliche Paketlaufzeiten, Paketverluste o.ä. verursacht werden.

Das folgende Beispiel erläutet dieses Problem anhand eines Netzwerks mit sechs Knoten, das in Abbildung 7.7 dargestellt ist. Das Bild zeigt ein bestehendes Netzwerk aus vier Knoten (A,B,C,D), an das sich ein kleines Netzwerk aus zwei Knoten (E,F) gerade anschließen will. Im Beispiel wird TERA zur Berechnung der Routen verwendet.

Die Verbindung zwischen den Knoten B und D ist dabei teilweise gestört. Dies wird durch die gestrichelten Linien symbolisiert. Dadurch haben Botschaften über diese Kante wegen der nötigen Wiederholungen etc. eine längere Laufzeit als über die anderen Kanten.

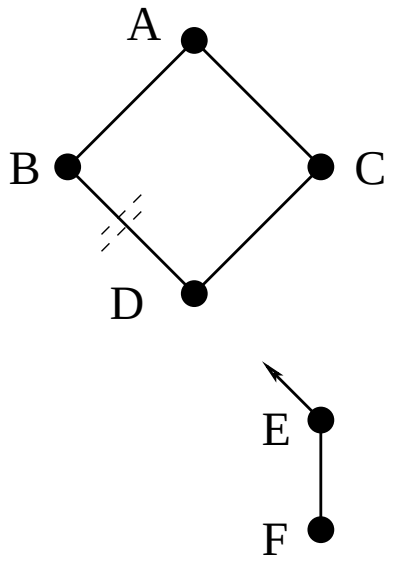


Abbildung 7.7: Netztopologie eines Beispielnetzes

Das Beispiel veranschaulicht die Vorgänge im Netzwerk (A,B,C,D) während die beiden Knoten angeschlossen werden. Dabei kommt zuerst eine Verbindung von Knoten D und E zustande. Die beiden Knoten erkennen die neue Verbindung und tauschen dann ihre Bäume miteinander aus. Anschließend berechnen beide Knoten ihre Bäume neu. Der Knoten D berechnet so einen erweiterten Baum, der zusätzlich E und F enthält. Diesen Baum schickt er an B und C. Diese Botschaft wird aufgrund der Störungen auf der Kante zwischen D und B verzögert, so

dass vorerst nur C die Botschaft erhält und damit seinen neuen Baum errechnet. Abbildung 7.8 veranschaulicht die in A, B und C zu diesem Zeitpunkt gespeicherten Bäume.

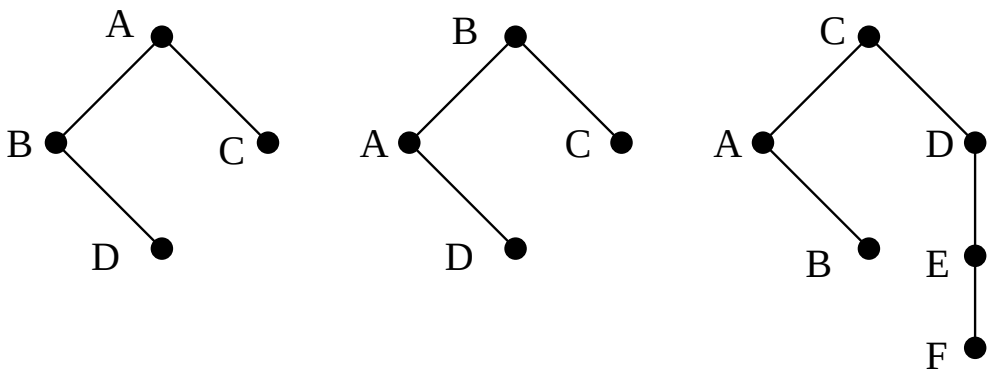


Abbildung 7.8: Minimal spannende Bäume der Knoten A, B und C

Der Knoten C sendet jetzt seinen erweiterten Baum an A und A startet aus diesem Grund eine Neuberechnung seines Baumes. Dadurch treten die ersten Widersprüche für Knoten A auf, denn der Baum von Knoten B enthält keine Stationen unterhalb von D, der Baum von C enthält aber die Stationen E und F.

Wenn Knoten A seinen Baum neu berechnet, dann untersucht er zuerst den von B angebotenen Baum und trägt alle über ihn erreichbaren Ziele in seinen neu entstehenden Baum ein. Daher wird der Knoten D als über den Nachbarn B erreichbares Ziel eingetragen. Anschließend wird der Baum von C untersucht und zuerst der Knoten C in den eigenen Baum übernommen. Der Knoten D kann nicht mehr übernommen werden, da er bereits über B eingetragen wurde.

Alle Knoten, die im Baum von C unterhalb des Knotens D liegen, können jetzt auch nicht mehr übernommen werden. Knoten A kann die ihm angebotenen Bäume seiner Nachbarn nicht beliebig mischen, denn ein Nachbar, der ein Ziel nicht anbietet, wird auch nie ein Paket an dieses Ziel weiterleiten. So können die Knoten E und F nicht in den Baum von A eingefügt werden.

Im Beispiel wird der Knoten A seinen Baum solange nicht erweitern können, bis B die bisher verzögerte Botschaft von D erhält. Der Zustand, in dem ein Knoten inkonsistente Informationen von seinen Nachbarn besitzt, dauert in der Regel nur eine kurze Zeitspanne, da dann auch B die neuen Verbindungsinformationen des Knoten D erhält, diese verarbeitet und an A weiterreicht, der somit wieder übereinstimmende Informationen besitzt und E und F in seinen Baum integrieren kann.

Um zu vermeiden, dass durch die kurzzeitigen Inkonsistenzen erreichbare Knoten, wie hier die Knoten E und F, als nicht erreichbar gekennzeichnet werden, wurde das Mehrwege-Routing aus dem vorhergehenden Kapitel um eine weitere Funktionalität erweitert: Jedes Mal, wenn die erste Routingtabelle komplett (TERA) oder teilweise (einfache Pfadsuche) neu berechnet wird, wird im Anschluss die zweite Routingtabelle vollständig neu berechnet. Im Unterschied zur vorangehenden Spezifikation wird diesmal aber nicht aufgegeben, wenn die erste Routingtabelle

keinen Pfad zum Ziel anbieten kann.

Im oben beschriebenen Beispiel bedeutet dies, dass Knoten A auch für die Ziele E und F die Routingtabellen seiner Nachbarn nochmals nach einer alternativen Route durchsucht. Dabei wird er beim Knoten C fündig. Bevor A diese Route nutzen kann, muss er sie auf Schleifengefahr hin überprüfen. Dies geschieht bei den Pfadsuche-Algorithmen durch Nachvollziehen des Pfades anhand der Vorgängerinformation. Die dazu notwendige Vorgehensweise ist bereits in 5.2.4 ausführlich beschrieben. Zur Prüfung nutzt ein Knoten lediglich die Routingtabelle des entsprechenden Nachbars, die ja komplett gespeichert ist, und nicht seine eigene Routingtabelle. Eine alternative Route darf nur dann genutzt werden, wenn durch die Prüfung sichergestellt ist, dass die Route keine Schleife enthält, ansonsten würden die alternativen Routen die Schleifensperre der Pfadsuche unterlaufen.

Wenn ein Knoten mit dem erweiterten Mehrwege-Routing ein Paket versendet, muss er jedes Mal beide Routingtabellen überprüfen. Ist in der ersten Routingtabelle eine Route eingetragen und in der zweiten Tabelle eine alternative Route vorhanden, dann wird, wie bei der vorangehenden Spezifikation, durch eine Zufallsentscheidung eine der beiden Routen ausgewählt und benutzt. Enthält lediglich die erste Routingtabelle einen gültigen Nachfolger, so bedeutet dies, dass keine Alternativroute mit der gleichen Distanz zum Ziel ermittelt werden konnte. In diesem Fall wird die erste Routingtabelle benutzt. Ist das Ziel in der ersten Routingtabelle als unerreichbar gekennzeichnet, die zweite Routingtabelle enthält jedoch einen gültigen Nachfolger, so bedeutet dies, dass der oben beschriebene Fall eingetreten ist und durch inkonsistente Informationen das Ziel nicht in den minimal spannenden Baum aufgenommen werden konnte. In diesem Fall wird die zweite Routingtabelle benutzt. Der Knoten glaubt also quasi dem Nachbarn, der behauptet, er könne das Ziel erreichen, auch wenn er dies in seinem eigenen Baum nicht nachvollziehen kann, und gibt das Datenpaket einfach weiter. Da er ja nachgeprüft hat, ob er auf dem Weg vom Nachbarn bis zum Ziel liegt, kann er davon ausgehen, das Paket nicht zurückzuerhalten.

Das gesamte Routingverfahren basiert auf den ersten Routingtabellen der Knoten, die auch an die jeweiligen Nachbarn weitergegeben werden. Für diese Routen wird durch die Pfadsuche z.B. die Schleifenfreiheit sichergestellt und immer der kürzeste Pfad ermittelt. Damit das Routingverfahren funktioniert, dürfen die in der zweiten Routingtabelle gespeicherten Informationen nicht an die Nachbarn weitergegeben werden. Diese Tabelle wird nur intern verwendet, deswegen dürfen bei ihr auch z.B. die Bäume zweier Nachbarn gemischt werden, ohne die Funktionsfähigkeit des Routingverfahrens zu gefährden. Der Hauptvorteil des erweiterten Mehrwege-Routing liegt in den geringen Kosten beim Einsatz des Verfahrens. Da die alternativen Routen nicht versendet werden und die Berechnung nur mit ohnehin gespeicherten Informationen auskommt, ist nur ein leicht erhöhter Berechnungsaufwand notwendig, um die Erreichbarkeit zu steigern. Im nächsten Abschnitt wird nun untersucht, welcher Gewinn mit dem Verfahren zu erzielen ist.

7.4.2 Simulationen

Das im Abschnitt 7.4.1 beschriebene erweiterte Mehrwege-Routing soll im folgenden zur leichteren Unterscheidung vom einfachen Mehrwege-Routing aus Abschnitt 7.3 als „Mehrwege+“ bezeichnet werden. Um die Leistungsfähigkeit dieses Mehrwege+ Routingverfahrens zu testen, wurden erneut Simulationen durchgeführt. Die Simulationen verwenden die gleichen Parameter, wie sie bereits bei der Untersuchung des ersten Mehrwege-Routing in Kapitel 7.3.1 eingesetzt wurden.

Zur Ermittlung der Ergebnisse werden jede simulierte Sekunde sämtliche Routen des Netzwerks überprüft. Dazu werden von jedem Knoten die Routen zu sämtlichen anderen Knoten des Netzwerks nachvollzogen. Die Ergebnisse zur Erreichbarkeit sind in Abbildung 7.9 zu sehen. In der Abbildung ist angegeben, welcher Prozentsatz der Knoten bei verschiedenen Geschwindigkeiten nicht mehr erreichbar sind. Während im fast statischen Fall bei einer Geschwindigkeit von 0,001 m/s sich nahezu alle Knoten gegenseitig erreichen können, sinkt die Anzahl der erreichbaren Knoten mit steigender Geschwindigkeit. Dabei treten immer häufiger Fehler auf, da immer schnellere Topologieänderungen verarbeitet werden müssen.

Wie in der Abbildung zu sehen ist, steigt durch den Einsatz des Mehrwege+ Routing die Erreichbarkeit durchgehend um ca. 0,3 - 0,4%. Bei einer Nicht-Erreichbarkeit von insgesamt ca. 3,5% bei 0,3 m/s bis hin zu ca. 5% bei 2 m/s ist dies schon eine bemerkenswert hohe Steigerung, da in der Summe der nicht erreichbaren Knoten hauptsächlich die Fälle enthalten sind, in denen wirklich keine Route zwischen zwei Knoten gefunden werden kann, d.h. es liegt nicht am Routingverfahren, dass keine Route gefunden werden kann, sondern es existiert vorübergehend gar keine Route, da sich die Knoten z.B. in verschiedenen Partitionen des Netzwerks aufhalten.

Mehrwege+ Routing kann bei Routingalgorithmen wie TERA die Anzahl der funktionierenden Routen im Netzwerk erhöhen. Da zur Berechnung der zweiten Routingtabelle lediglich die Nachbar-Routingtabellen nötig sind, die sowieso schon in den Knoten gespeichert sind, kann auf diese Weise ohne jeglichen Kommunikationsaufwand und mit nur geringem Rechenaufwand ein deutlicher Vorteil erzielt werden.

7.5 Analyse der Ausfallzeiten

Den Abschluss der Simulationen bilden Versuche zu den Ausfallzeiten einer Route. Die Simulationen verwenden dabei die schon aus den letzten Untersuchungen bekannten Parameter, die in 7.3.1 aufgelistet sind. Als Routingalgorithmus wird TERA mit und ohne die Mehrwege-Option eingesetzt.

Zur Bestimmung der Ausfallzeiten wird vor dem Start der Simulation ein Knotenpaar zufällig ausgewählt und die Route zwischen diesen beiden Knoten während der gesamten Simulationsdauer analysiert. Alle 0.1 Sekunden (Simulations-Zeit) der simulierten Netzwerkaktivität wird geprüft, ob die Route noch funktionsfähig ist. Sobald eine Unterbrechung festgestellt wurde,

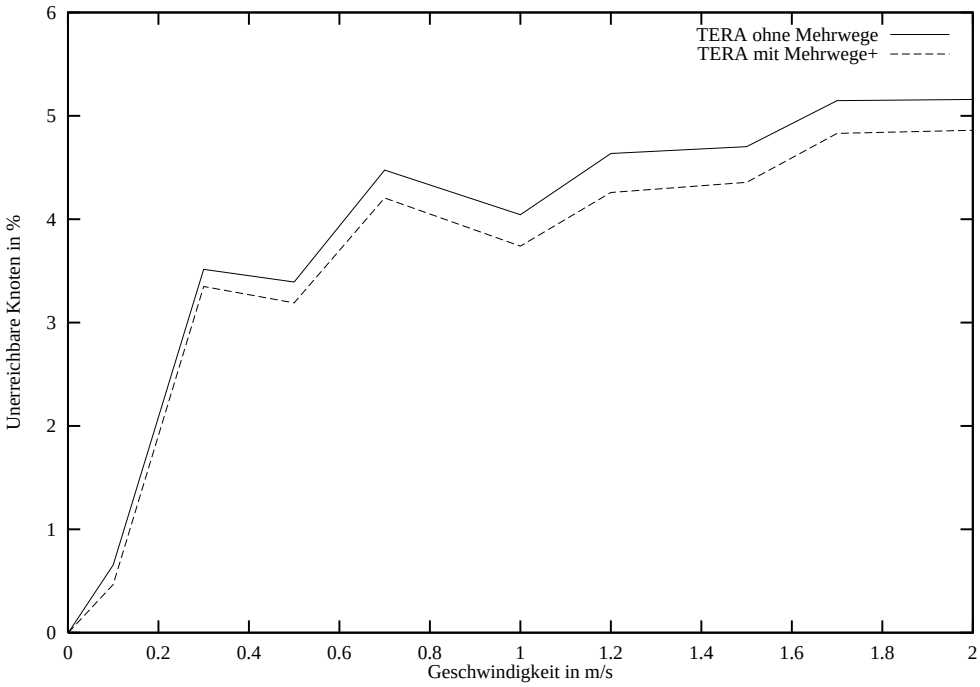


Abbildung 7.9: Nicht erreichbare Knoten mit TERA und Mehrwege+

wird die Zeit gespeichert und auf das Ende der Unterbrechung gewartet. Aus der Differenz der beiden Zeiten wird die Ausfallzeit errechnet und für die spätere Auswertung in einer Liste gespeichert.

Abbildung 7.10 zeigt die durchschnittliche Länge der Ausfallzeiten für die Bewegungsgeschwindigkeiten von 0,001 bis 2 m/s.

Bei der Geschwindigkeit 0,001 m/s beträgt die durchschnittliche Ausfallzeit unabhängig vom Routingalgorithmus Null. Da hier ein quasi statisches Netzwerk vorliegt, treten auch keine Unterbrechungen auf. Bei allen anderen Geschwindigkeiten unterscheidet sich die durchschnittliche Ausfallzeit des Mehrwege+ Routing deutlich vom einem Routing ohne diese Option.

Routing ohne Mehrwege-Funktionalität zeigt besonders lange Ausfallzeiten bei der niedrigen Geschwindigkeit von 0,1 m/s. Diese Ausfallzeit kann durch das Funktionsprinzip der adaptiven Verbindungserkennung erklärt werden. Wenn bei geringen Geschwindigkeiten wenig Veränderungen in der Umgebung eines Knotens auftreten, so versendet dieser auch nur in großen Zeitabständen Erkennungspakete und bemerkt folgerichtig Verbindungsverluste auch nur nach großen Zeitspannen. Geht beim Routing ohne Mehrwege-Funktionalität eine Verbindung verloren, so können während der gesamten Zeitspanne, bis der Verlust der Verbindung bemerkt wird, keine Datenpakete mehr zum Empfänger gelangen, da alle Pakete dieselbe Route verwenden. Die Ausfallzeit dauert also so lange, bis durch das Aussenden des nächsten Erkennungspaketes der Verbindungsverlust bemerkt wird und eine neue, funktionierende Route berechnet wird. Wenn, wie bei der Geschwindigkeit 0,1 m/s, nur sehr wenige Veränderungen in der Umgebung

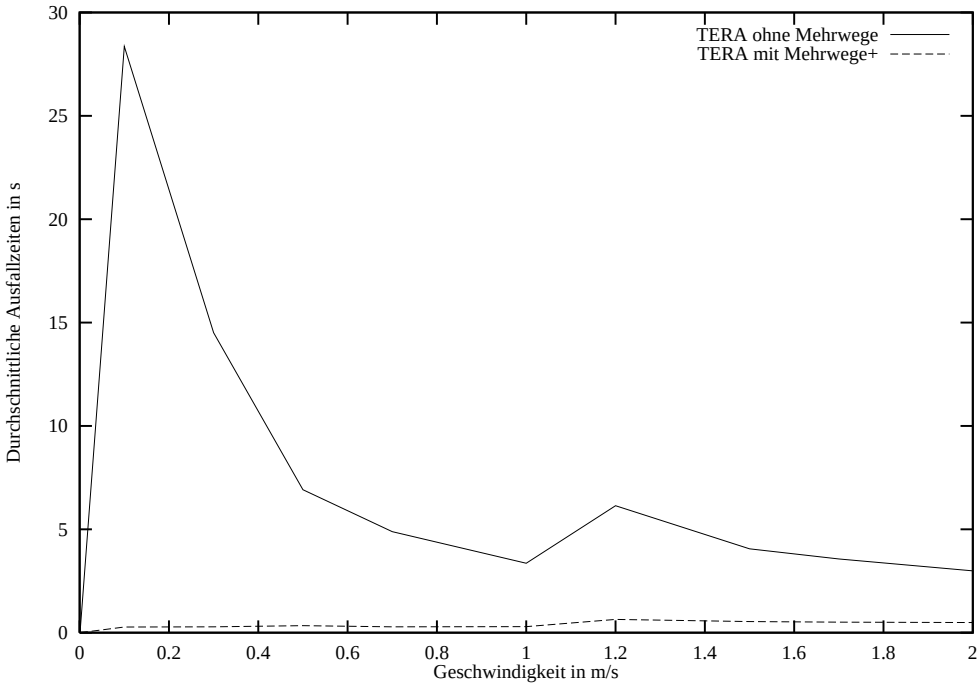


Abbildung 7.10: Durchschnittliche Länge der Ausfallzeiten

eines Knotens stattfinden, so versendet dieser durch die adaptive Verbindungserkennung auch nur in großen Zeitabständen Erkennungspakete. Die maximale Zeitspanne, die zwischen dem Senden zweier Erkennungspakete liegen kann, beträgt dabei 60 Sekunden, d.h. im Schnitt wird der Verlust einer Verbindung bei niedrigen Knotengeschwindigkeiten nach 30 Sekunden erkannt. Der Wert der durchschnittlichen Ausfallzeit in Abbildung 7.10 für die Geschwindigkeit 0,1 m/s nähert sich deshalb der 30-Sekunden-Marke auch sehr stark an.

Je höher die Durchschnittsgeschwindigkeit der Knoten ist, um so mehr Veränderungen finden in der Umgebung der einzelnen Knoten statt und nach um so kürzeren Zeitspannen werden durch die adaptive Verbindungserkennung erneut Erkennungspakete verschickt. Dadurch werden wiederum Verbindungsverluste schneller erkannt, weshalb die durchschnittliche Dauer einer Ausfallzeit für Routing ohne Mehrwege-Funktionalität mit steigender Geschwindigkeit sinkt.

Beim Routing mit Mehrwege+ werden die Datenpakete über mehrere Routen gesendet, dadurch reduzieren sich die Ausfallzeiten drastisch. Geht auf einer Route die Verbindung verloren, was bei geringen Knotengeschwindigkeiten erst nach fast 30 Sekunden bemerkt wird, so können Datenpakete auf alternativen Routen noch immer ihr Ziel erreichen. So kommt beim nächsten Versuch, über eine andere Route, wieder ein Datenpaket beim Empfänger an, wodurch die durchschnittliche Ausfallzeit auf wenige Zehntelsekunden sinkt (vgl. Abb. 7.10).

Dafür geht allerdings mit Mehrwege+ Routing im Falle eines Verbindungsverlustes, bei zwei möglichen Routen, jedes zweite Datenpaket verloren. Dies erhöht die Anzahl der Ausfälle beträchtlich, da ein Ausfall mit einer Dauer von 30 Sekunden beim Routing ohne Mehrwege-

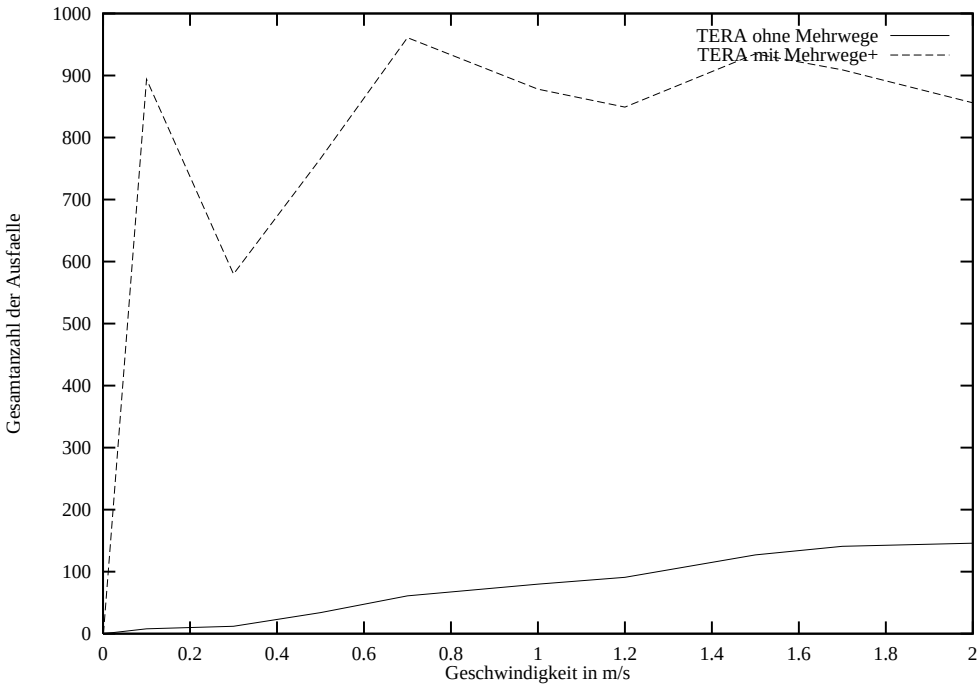


Abbildung 7.11: Anzahl der Ausfälle

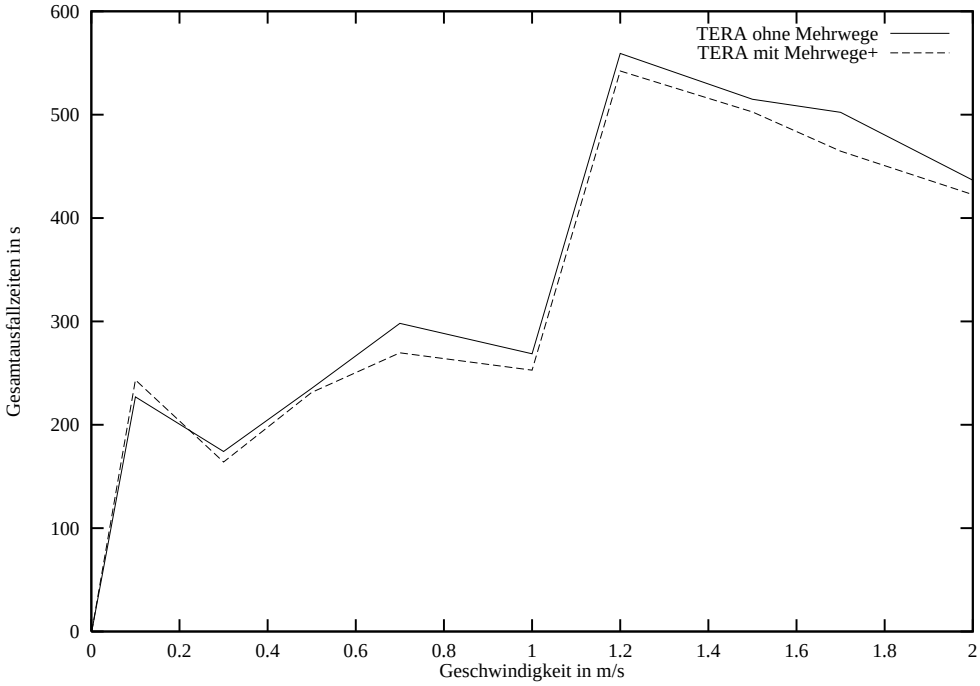


Abbildung 7.12: Summe der Ausfallzeiten

Funktionalität durch Mehrwege+ Routing in circa 150 Ausfälle mit einer durchschnittlichen Dauer von 0,2 Sekunden umgewandelt wird.

Diese Vermutung belegt auch die Grafik in Abbildung 7.11, in der die Gesamtzahl der Ausfälle der oben untersuchten Verbindung dargestellt ist.

In der Simulation ist die Wahrscheinlichkeit für einen Verbindungsverlust immer gleich groß. Es ist daher egal, ob eine Menge von Datenpakete über einen oder mehrere Wege an das Ziel gesendet werden. Auf allen Verbindungen ist die Wahrscheinlichkeit für den Verlust gleich groß. Deshalb kann durch Mehrwege-Routing die Gesamtausfallzeit der Verbindung auch nicht verringert werden, sondern es können lediglich wenige lange Ausfallzeiten in mehrere kurze umgewandelt werden. Dies bestätigt Abbildung 7.12, die die Gesamtausfallzeit, d.h. die Summe der einzelnen Ausfallzeiten der Verbindung, darstellt.

In der Abbildung ist die Summe der Ausfallzeiten einer Routingsimulation mit und ohne Mehrwege-Funktionalität fast identisch. Der geringe Vorteil des Routing mit Mehrwege+ entsteht durch die im letzten Abschnitt beschriebene Verbesserung der Erreichbarkeit, wodurch in Einzelfällen Empfänger noch erreicht werden, bei denen das Routing ohne Mehrwege-Funktionalität versagt. Generell ist durch Mehrwege-Routing jedoch kein Vorteil hinsichtlich der Gesamtausfallzeit einer Verbindung zu erzielen.

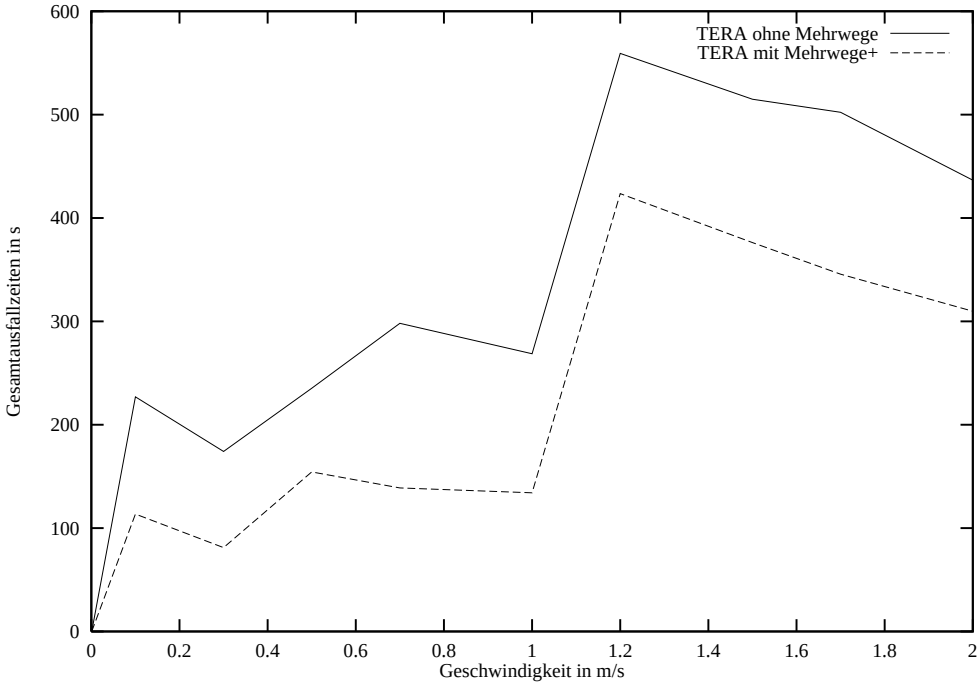


Abbildung 7.13: Summe der Ausfallzeiten bei Mehrfachsendungen

Es wurde jedoch gezeigt, dass Mehrwege-Routing das Ausfallverhalten einer Verbindung dahingehend verändert, um mehr, dafür aber kürzere Ausfälle zu erhalten. Diese Variante ist gegenüber derjenigen mit wenigen langen Ausfallzeiten zu bevorzugen, da sich kurze Fehler ein-

facher korrigieren lassen.

Schon ein einfaches Fehlerkorrekturverfahren kann den Verlust einzelner Datenpakete durch Wiederholungen kompensieren. Dabei fordert der Empfänger eine Wiederholung der nicht eingetroffenen Datenpakete an. Der Sender verschickt darauf hin so oft Kopien der Datenpakete, bis der Empfänger diese vollständig erhalten hat. Dieses Verfahren ist, in ähnlicher Form, im TCP-Standard implementiert. Aus diesem Grund kann auch TCP kurze Verbindungsausfälle hinnehmen, auch wenn dadurch die Übertragungsgeschwindigkeit deutlich absinkt. Für die Korrekturverfahren sind langanhaltende Ausfälle viel ungünstiger, da das Fehlerkorrekturverfahren dann keine Rückmeldungen mehr erhält und eventuell die Wiederholungen der Datenpakete abbricht. Daher ist die Verwendung eines Fehlerkorrekturverfahrens zusammen mit dem Einsatz von Mehrwege-Routing besonders sinnvoll, da die Fehlerkorrektur die kurzen Ausfälle leicht kompensieren kann.

Die letzte Simulation dieser Reihe untersucht die Gesamtausfallzeiten bei der gleichzeitigen Aussendung von jeweils drei Kopien eines Datenpakets. Das Ergebnis ist in 7.13 abgebildet. Die Graphik zeigt, wie bei der letzten Simulation, die Gesamtausfallzeit der Verbindung in Abhängigkeit von den Bewegungsgeschwindigkeiten der Konten. Für ein Routing ohne Mehrwege-Funktionalität ergibt sich durch die mehrfache Aussendung kein erkennbarer Vorteil. Beim Mehrwege-Routing wird die Gesamtausfallzeit der Verbindung aber deutlich vermindert, wie Abbildung 7.13 zeigt. Da jetzt mehrere, verschiedene Routen für ein und denselben Datensatz verwendet werden, sinkt die Verlustwahrscheinlichkeit für diese Nachricht. Wenn über die gesamte Laufzeit immer gleichzeitig mehrere Pakete verschickt werden, sinkt damit auch die Gesamtausfallzeit.

Durch gleichzeitiges Versenden eines Datenpaketes über verschiedene Routen kann in dieser Simulation die Gesamtausfallzeit der Route stellenweise um über 150 Sekunden vermindert werden, was einer Absenkung von über 30% entspricht. Dieser Vorteil wird allerdings durch eine mehrfache Belastung den Netzwerks durch die gesendeten Kopien erkauft. Daher ist diese Vorgehensweise nur für besonders kritische Daten empfehlenswert.

Diese Methode kann aber noch deutlich verbessert werden, wenn nicht einfach Duplikate von Paketen auf den Weg gebracht werden, sondern die Pakete mit Methoden aus der Fehlerschutzkodierung so erweitert werden, dass der Verlust einiger Pakete durch redundante Informationen ausgeglichen werden kann. Die mathematischen Voraussetzungen dafür sind vorhanden [ACLGM93] und können dem Algorithmus hinzugefügt werden.

Die Untersuchung der Ausfallzeiten durch Simulationen hat letztlich gezeigt, dass durch Mehrwege-Routing wenige und lange Ausfallzeiten in mehrere, dafür aber kürzere Ausfälle umzuwandeln sind. Viele Anwendungen, darunter auch die gebräuchlichsten Fehlerkorrekturverfahren, kommen mit den kurzen Ausfällen besser zurecht, auch wenn diese häufiger auftreten. Der Entwurf neuer Fehlerkorrekturverfahren, die direkt mit dem Mehrwege-Routing zusammenarbeiten, ist ein aussichtsreiches Forschungsgebiet für zukünftige Arbeiten.

7.6 Zusammenfassung

In diesem Kapitel wurde die Lastverteilung durch Mehrwegerouting vorgestellt. Dabei wird der Vorteil eines Paketfunksystems genutzt, bei dem nicht alle Pakete den gleichen Weg durch das Netzwerk nehmen müssen. Wenn mehrere Wege zur Verfügung stehen, dann werden die Pakete mit einer Zufallsentscheidung auf die Wege verteilt. Die vorgestellte Spezifikation setzt auf die in Kapitel 5 beschriebenen Pfadsuche-Algorithmen auf und zeigt, wie diese Algorithmen erweitert werden müssen, um Mehrwege-Routing nutzen zu können.

Anschließend wurde die Lastverteilung, die sich mit diesem Verfahren erreichen lässt, durch Simulationen untersucht. Die ersten Simulationen betrachten ein statisches Netzwerk, in dem zwei Datenströme übertragen werden. Es wurde veranschaulicht, dass im Fall ohne Mehrwegerouting insbesondere am Kreuzungspunkt der beiden Datenströme erhebliche Belastungen auftreten, bei aktiviertem Mehrwegerouting zeigte sich eine optimale Verteilung der Last.

Danach wurde die Lastverteilung in dynamischen Netzwerken untersucht. Es wurde dargelegt, dass in solchen Netzwerken nicht einzelne Knoten untersucht werden können, sondern dass über eine lange Simulationszeit die auftretenden Maxima und Standardabweichungen erfasst werden müssen. Die Untersuchungen ergaben auch in dynamischen Netzwerken für die Belastung geringere Maxima und eine kleinere Standardabweichung, wenn Mehrwegerouting eingesetzt wird.

Anschließend wurde das Problem inkonsistenter Routingtabellen behandelt. Dabei wurde erläutert, wie die von den Nachbarn gemeldeten Routingtabellen durch Laufzeitunterschiede der Botschaften inkonsistent werden können. Es wurde ein Vorschlag zur Modifikation des Mehrwege-Routings gemacht, der die Inkonsistenzen hinnimmt und trotzdem die bestmögliche Erreichbarkeit erzielt. Die Funktionsfähigkeit der Modifikation wurde durch Simulationen nachgewiesen.

Im letzten Teil des Kapitels wurden Ausfallzeiten von Verbindungen in dynamischen Ad-hoc Netzwerken untersucht. Die Analyse zeigt, dass in erster Linie die Nachbarschaftserkennung die Dauer dieser Ausfälle bestimmt, dass aber auch das verwendete Routingverfahren Einfluss auf die Ausfälle hat. Daher wurde die Art der Ausfälle mit und ohne Mehrwegerouting untersucht. Dabei wurden bei Verbindungen ohne Mehrwegerouting weniger, aber dafür langanhaltende Ausfälle gemessen. Dagegen sind mit Mehrwegerouting häufigere aber kurzzeitige Ausfälle zu finden. Es wurde dargestellt, dass ein Fehlerkorrekturverfahren wie beispielsweise TCP durch automatisches Wiederholen verlorener Pakete die häufigen und kurzen Ausfälle leichter korrigieren kann.

Kapitel 8

Internetanbindung kooperativer Paketfunknetze

8.1 Einleitung

In den letzten Kapiteln wurde das Routing innerhalb von Ad-hoc Netzwerken beschrieben und mit Hilfe vieler Simulationen analysiert. Viele Anwender benötigen aber nicht nur eine Vernetzung von Rechnern in einem abgeschlossenen Ad-hoc Netzwerk, sondern die Verbindungen sollen weiter bis in das Internet reichen.

Die in diesem Kapitel angebotenen Lösungen nutzen noch das Internet Protokoll Version 4 (IPv4). Mittlerweile existiert ein verbessertes Protokoll unter der Bezeichnung IPv6 [SM95, DH98], das wesentlich mehr Möglichkeiten eröffnet. Die vorgestellten Konzepte lassen sich jedoch ohne Probleme auch auf IPv6 übertragen. Da die meisten Netzwerke in Deutschland noch IPv4 nutzen, wurden die Lösungen für IPv4 realisiert. Einige Maßnahmen, beispielsweise zur Einsparung von Internetadressen, sind bei der Verwendung von IPv6 nicht mehr erforderlich. Zu Beginn des Kapitels wird deswegen die Weiterleitung von Paketen über IPv4 kurz erläutert und die Konfiguration des dabei eingesetzten Netzwerkstacks erklärt.

Da das Routing im Internet keine portablen Stationen unterstützt, wurden zusätzliche Protokolle wie Mobile-IP standardisiert, die diese Funktionalität nachträglich implementieren. Das Funktionsprinzip dieses Protokolls wird ausführlich dargestellt, da insbesondere die Umleitung von Paketen über sogenannte Tunnel eine notwendige Basisfunktion ist, die auch in den später angegebenen Anbindungskonzepten noch genutzt wird. Außerdem werden spezielle Erweiterungen für den Mobile-IP Standard erläutert, die eine Anbindung ganzer Netzwerke statt einzelner Stationen ermöglichen.

Anschließend werden verbesserte Konzepte zur Anbindung von Ad-hoc Netzwerken an das Internet vorgestellt. Die Konzepte basieren nicht auf dem Mobile-IP Protokoll, obwohl sie einige Funktionalitäten dieses Protokolls übernehmen. Statt dessen werden moderne Verfahren

aus der Netzwerktechnik zur transparenten Anbindung von Subnetzwerken eingesetzt. Die Beschreibung erläutert in mehreren Schritten, wie die Kompatibilität eines Ad-hoc Netzwerks mit IPv4 hergestellt wird und geht anschließend auf die Möglichkeiten zur Verbindung von Ad-hoc Netzen und dem Internet ein.

Dazu wird zuerst das Maskieren von Netzwerken erläutert, mit dem ein ganzes Netzwerk über eine einzige Netzwerkadresse angebunden werden kann. Diese Anbindungslösung wird anhand eines Beispiels ausführlich erläutert. Als zweite Anbindungsvariante werden die Virtuellen Privaten Netzwerke (VPNs) erläutert, die Rechner über sichere Tunnel miteinander verbinden. Auch dabei wird mit Hilfe eines Beispiels eine Lösung für Ad-hoc Netze erläutert. Abschließend werden die wichtigsten Eigenschaften der einzelnen Varianten noch einmal zusammengefasst.

8.2 Routing im Internet

Das Internet ist ein „Netz von Netzen“, das vorrangig die Interoperabilität zwischen Netzwerken unterschiedlicher Art ermöglichen muss. Dieses Netz ist fest und hierarchisch organisiert, eine Anpassung an veränderte Topologien geschieht schon wegen der enormen Größe des Netzwerks sehr langsam. Ad-hoc Netzwerke sind dagegen nicht für große Knotenzahlen ausgelegt. Sie können dafür schnelle Topologieänderungen akzeptieren und ohne feste Infrastruktur auskommen.

Im Internet wird jedem Rechner eine feste IP-Nummer zugewiesen. Diese Nummern sind hierarchisch organisiert, sie bestehen aus zwei Teilen, der Netzwerknummer und der Rechnernummer. Über die Netzwerknummer sind Rechner, wie bei der Vorwahl im Telefonnetz, einer Gruppe zuzuordnen. Die Rechnernummer gibt an, welcher Rechner innerhalb einer Gruppe angesprochen wird.

Die Berechnung der Routen ist ein relativ komplexer Prozess und in mehreren Ebenen organisiert. Schon eine Beschreibung des Routingprotokolls der obersten Stufe [RL95] übersteigt den Rahmen dieser Arbeit. Daher sei zu diesem Thema auf das Buch [Ste95] verwiesen, das eine Beschreibung aller Protokolle enthält und auch deren Entwicklungsgeschichte beschreibt. Die dort beschriebenen Routingprotokolle sind für das weitere Verständis der Arbeit nicht essentiell erforderlich, da diese Protokolle keine mobilen Stationen unterstützen und daher für den vorliegenden Anwendungsfall als quasi statisch angenommen werden müssen.

Die Unterteilung in Netzwerknummern ermöglicht erst das Routing für die große Anzahl von Rechnern im Internet. Ohne eine Unterteilung sind theoretisch 2^{32} Adressen in IPv4 verfügbar und ein Routing ohne Hierarchien muss eine Routingtabelle mit dieser Menge von Einträgen verwalten. Durch die Unterteilung in Netzwerknummern sinkt die Anzahl bei einer Subnetzgröße von 2^8 Stationen auf maximal 2^{24} Einträge. In der Praxis sind es deutlich weniger, da viele Netzwerke mehr als 2^8 Stationen umfassen und einige Nummern nicht belegt sind. Die verbleibende Menge an Routinginformation ist immer noch beachtlich, aber sie ist mit derzeit gebräuchlichen Arbeitspeichergrößen schon verwaltbar.

Die hierarchische Organisation der Rechner in Subnetzwerke und das Routing über Netzwerknummern erzwingt eine Ortsgebundenheit der Rechner. Das Routing orientiert sich vorrangig an der Netzwerknummer und kann daher einzelne Rechner, die an andere Orte verlegt werden, nicht berücksichtigen. Rechner, die verlegt werden, müssen analog wie beim Telefonsystem, eine neue Nummer erhalten, deren Netzwerknummer/Vorwahl der neuen Position entspricht. Der Bedarf nach einer Lösung für das Problem mobiler Rechner hat zur Entwicklung des Mobile-IP-Protokolls geführt, das anschließend beschrieben wird. Vorher müssen jedoch einige Begriffe erläutert werden, die erklären, wie in einer Station mit IPv4 die Weiterleitung von Paketen, entsprechend der vorab berechneten Pfade, konfiguriert wird.

8.2.1 Der IPv4 Stack

Im IP-Protokoll wird das Routing durch Routingtabellen gesteuert, die in jeder Station vorhanden sind. Jeder Rechner, der das IP-Protokoll unterstützt, hat einen IP-Stack und besitzt eine Routingtabelle, die bestimmt, über welche Ausgangsleitung ein Paket gesendet werden soll. Die meisten Stationen besitzen nur eine Ausgangsleitung und deswegen enthält deren Routingtabelle nur einen Eintrag, der als Defaultroute bezeichnet wird. Für Stationen, die mehrere Leitungen nutzen und Pakete auch weitervermitteln müssen, ist die Tabelle komplizierter, hier gibt es drei Arten von Tabelleneinträgen:

- Die **Hostrouten** sind Einträge, die für die IP-Nummer einer bestimmten Station eine Ausgangsleitung festlegen. Diese Einträge haben die höchste Priorität.
- Die **Netzwerkrouten** bestimmen die Ausgangsleitung für eine Gruppe von Stationen, die in einem Netzwerk liegen. Der IP-Stack untersucht dazu nur den vorderen Teil einer IP-Nummer, um die Netzwerkadresse zu lesen und daraus die Ausgangsleitung zu bestimmen.
- Die **Defaultroute** hat die niedrigste Priorität und wird nur dann verwendet, wenn keine andere Route zu finden war.

Das in den vorigen Kapiteln vorgestellte Ad-hoc Routing ist als flaches Routing konzipiert. Es berechnet für jedes einzelne Ziel im Ad-hoc Netz die richtige Ausgangsleitung. Es bietet sich daher an, die Ergebnisse der Routenberechnung als Hostrouten in den IP Stack einzutragen. Eine Station, die nur im Ad-hoc Netzwerk kommuniziert, hat keine Defaultroute, sondern nur Hostrouten zu den möglichen Zielen. In dieser Konfiguration kann eine Applikation schon beim Verbindungsaufbau vom IP-Stack darüber informiert werden, wenn eine Route zu einem gewünschten Ziel nicht existiert.

Eine Station, die gleichzeitig über einen Internetzugang und im Ad-hoc Netzwerk kommuniziert, besitzt dann Hostrouten und eine Defaultroute. Die Defaultroute verweist auf den nächsten Router im Internet. Damit gehen alle Pakete an den Internetrouter, die der Ad-hoc Routingalgorithmus nicht zuordnen kann.

8.2.2 Mobilitätsunterstützung im Internet

Da das Routing im Internet, schon allein wegen der Größe des Netzwerks, die Routen nur äußerst langsam ändern kann, ist es für die Mobilitätsunterstützung nicht zu gebrauchen. Es wird daher als quasi statisch angesehen, und es werden neue Wege gesucht, um innerhalb des statischen Internetrouting trotzdem mobile Stationen zu erlauben.

Aus der Annahme eines statischen Routings, ergeben sich sofort zwei Konsequenzen für mobile Stationen:

1. Eine bewegte Station muß, um im Internet kommunizieren zu können, immer die IP-Nummer annehmen, die am jeweiligen Ort der Station gerade gültig ist. Dabei kann die Station auch über einen lokalen Vermittler kommunizieren und so dessen Nummer mitbenutzen.

2. Wenn eine mobile Station vom Internet aus angesprochen werden soll, dann muss es einen Auskunftsdienst oder Vermittler an einem festen Punkt im Internet geben, dem bekannt ist, unter welcher Nummer die mobile Station gerade zu erreichen ist.

Um die Mobilitätsunterstützung für alle anderen Internetstationen vollkommen transparent zu machen, sind weitere Protokolle nötig, die nun vorgestellt werden. Zu diesem Zweck wurde der Mobile-IP-Standard geschaffen, der ein zusätzliches Routingprotokoll für portable Stationen definiert und im folgenden Abschnitt ausführlich erläutert wird. Vorab muss jedoch das Tunneln von Paketen beschrieben werden.

Tunneln von Paketen

Es existieren mehrere Gründe, Pakete nicht über das normale Routing ausliefern zu lassen. Es können beispielsweise Sicherheitsbedenken gegen eine Übermittlung von Paketen durch Netzwerke mit unbekannten Stationen vorliegen. Das normale Routing schickt jedoch Pakete durch diese Netzwerke, sobald sie den kürzesten Weg anbieten. Ein anderes Beispiel ist die Versendung von Paketen, die ein Sonderformat aufweisen, und sich deshalb nicht mit normalem Routing ausliefern lassen.

Die wichtigste Anwendung besteht aber in der Umleitung von Paketen zu Zielen, die sich vorübergehend an einem anderen Ort aufhalten. Um so ein Paket trotzdem in der gewünschten Art und Weise ausliefern zu können, wird zwischen zwei Stationen ein sogenannter IP-Tunnel aufgebaut, durch den dann die eigentliche Auslieferung erfolgt [FL00]. Dazu werden die zu übertragenden Pakete von der einen Station in ein größeres Paket komplett eingepackt (gekapselt) und dann an die Station am anderen Ende des Tunnels gesendet, wo sie wieder ausgepackt (entkapselt) werden.

Die auszuliefernden Pakete werden beim Tunneln wie ganz normale Daten behandelt und innerhalb größerer Pakete verschickt. Durch das Verpacken entsteht immer eine erhöhte Netz-

belastung, da jedes Paket nun zwei Header enthält. Für dieses Problem sind auch Verfahren verfügbar, die eine Kompression der ungenutzten Header anbieten [Ran96, Pal97].

Um die Umleitung von Pakete für andere Stationen möglichst transparent zu gestalten, ist die Kooperation mehrerer Stationen nötig. Der nachfolgend beschriebene Mobile-IP Standard beschreibt die dazu notwendige Vorgehensweise.

Mobile-IP

Funktionsweise Mobile-IP ist eine Erweiterung des IP-Standards [Gro96], die es einzelnen portablen Stationen, den sogenannten Mobile Hosts (MH), ermöglichen soll unabhängig vom Standort unter einer IP Adresse erreichbar zu sein. Um dies zu erreichen, werden zwei weitere Stationen benötigt, die Home Agent (HA) und Foreign Agent (FA) genannt werden.

Der HA ist an einem festen Punkt im Internet angesiedelt, besitzt eine eigene IP-Nummer und erfüllt die Aufgabe eines Vermittlers. Er nimmt Pakete, die an den MH adressiert sind entgegen und leitet sie an den FA weiter. Der HA muss dazu Pakete, die an die IP-Nummer des MHs adressiert sind, abfangen und durch einen Tunnel an den FA senden.

Ein MH muss, um eine Internetverbindung zu erlangen, sich mit einem FA verbinden. Der FA stellt nun dem MH seine IP-Adresse indirekt zur Verfügung, diese Adresse wird im Standard als „Care-of-Adresse“ bezeichnet. Der FA nimmt aus dem Tunnel die Pakete entgegen und gibt sie an den MH weiter.

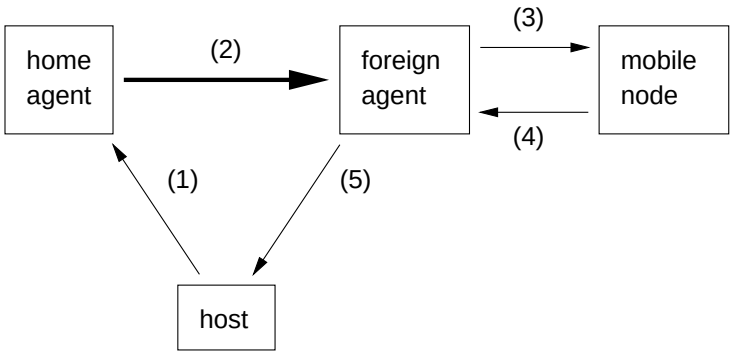
In der Basiskonfiguration ist Mobile-IP nur für die Umleitung von Paketen zum MH zuständig. Pakete die von MH gesendet werden, werden nicht umgeleitet, da das Routing ab dem FA diese normalerweise korrekt ausliefert.

Der ganze Vorgang ist in Abbildung 8.1 nochmals dargestellt. Die Abbildung stammt aus [Dem99] und zeigt die Kommunikation zwischen Home Agent, Foreign Agent, Mobile Host und einem beliebigen Internetrechner, der hier Host genannt wird.

In Mobile-IP hat jeder MH seine eigene IP-Adresse. Solange sich der MH in seinen Heimatnetzwerk aufhält, wird der HA nicht aktiv und die Pakete werden dem MH direkt zugeleitet. Verlässt der MH sein Heimatnetzwerk, dann sucht er sich einen FA, dieser teilt dem HA die neue Care-of-Adresse mit und aktiviert so die Umleitung der Pakete durch den Tunnel.

Im Standard ist auch eine Variante beschrieben, die ohne FA auskommt. Ist in einem Netzwerk kein FA vorhanden, dann kann der MH auch selbst die Rolle des FA übernehmen. Der MH muss dazu allerdings eine lokale IP-Adresse annehmen und zu seinem HA einen Tunnel aufbauen. Anschließend muss er die über den Tunnel kommenden Pakete selbst entkapseln und die so gewonnenen Pakete mit seiner Heimat IP-Nummer nochmals durch seinen IP-Stack leiten.

Protokolleigenschaften Mobile-IP erfüllt bereits viele Anforderungen, die für den Betrieb portabler Stationen nötig sind:



(1) Der externe Rechner schickt ein IP-Datagram an die feste IP-Adresse des Mobile Host.

(2) Der Home Agent fängt das IP-Datagram ab und schickt es durch einen Tunnel an die Care-of-Addresse des Mobile Host.

(3) Der Foreign Agent nimmt das IP-Datagram am Ende des Tunnels entgegen und leitet es an den Mobile Host weiter.

(4) Der Mobile Host schickt eine Antwort an den Foreign Agent.

(5) Der Foreign Agent schickt die Antwort weiter an den externen Rechner (Host).

Abbildung 8.1: Kommunikation über Mobile-IP

- Der MH kann sein Heimatnetzwerk verlassen und bleibt trotzdem unter seiner IP-Nummer erreichbar.
- Die Position darf sich auch während der Kommunikation des MH verändern. Ein Wechsel des FA geschieht einfach durch ein Umschalten des Tunnels auf einen anderen FA.
- Die Umleitung der Pakete ist für alle Internetrechner außer dem MH, FA und HA transparent.

Es gibt einige Gesichtspunkte, die den praktischen Einsatz des vorgestellten Protokolls bzw. des Mobile-IP Protokolls im Allgemeinen erschweren. Ein Problem entsteht durch die vom MH versendeten Pakete, da er Pakete mit einer „falschen“ Absender-Adresse verschickt. Da viele Angriffe auf fremde Systeme durch das Verwenden falscher Absender-Adressen ermöglicht werden, sind fast alle Firewall-Rechner so konfiguriert, dass sie Pakete mit offensichtlich falschen Absender-Adressen einfach ausfiltern, d.h. verwerfen.

Diese Maßnahme dient dem Schutz von Rechnern, die sich außerhalb der Firewall befinden, vor Benutzern, die hinter der Firewall agieren. Auf diese Weise können z.B. Universitäten die Rechner in der restlichen Welt vor ihren allzu experimentierfreudigen Studenten schützen. Bedauerlicherweise macht das die Verwendung von mobilen Rechnern innerhalb des Campus unmöglich.

Um dieses Problem zu vermeiden, lässt sich Mobile-IP so erweitern, dass der MH sowohl die gesendeten als auch die empfangenen Pakete immer über den HA leitet. Dieses Verfahren ver-

braucht mehr Netzwerkkapazität, da die vom MH gesendeten Pakete nun nicht mehr den direkten Weg nehmen.

Durch das Tunneln der Pakete in beiden Richtungen werden viele Probleme vermieden, die beim sogenannten Dreiecksrouting auftreten, wie es in Mobile IP verwendet wird. Dreiecksrouting kann zu Problemen bei der Übertragung führen, wenn die Laufzeiten der gesendeten und empfangenen Pakete sehr unterschiedlich sind. Das ist bei Mobile-IP insbesondere dann der Fall, wenn zwei Stationen direkt nebeneinander positioniert sind, aber der HA von den Stationen weit entfernt gelegen ist. Das im Internet eingesetzte Verfahren zur Flusskontrolle (ein Teil des TCP Standards) ist auf symmetrische Laufzeiten hin optimiert und zeigt in solchen besonderen Situationen dann deutliche Leistungseinbußen.

Die Sicherheitsfunktionen dieses Protokolls werden im Standard ebenfalls festgelegt, allerdings ist nur der Tunnelaufbau durch ein kryptographisches Verfahren geschützt. Der HA prüft damit jede Anforderung nach einem neuen oder veränderten Tunnel. So wird sichergestellt, dass nur der richtige MH einen Tunnel aufbauen oder verändern kann.

Viele andere sicherheitskritische Punkte sind allerdings noch offen, so ist beispielsweise die gesamte Kommunikation nicht gegen Abhören geschützt. Ein anderes Beispiel ist der nach dem Aufbau ungesicherte Tunnel. Ein Angreifer kann nach dem Tunnelaufbau den Tunnel übernehmen und an die Stelle des MH treten. Dies ist besonders für die spätere Verwendung in Ad-hoc Netzwerken kritisch, da hier vermittelnde Stationen leicht eine Übernahme durchführen können.

Der vorgestellte Mobile-IP Standard zeigt eine Lösung auf, um portable Rechner im Internet zu betreiben. Die Lösung muss allerdings für das vorliegende Problem, einer Anbindung ganzer Netzwerke aus mobilen Stationen, noch um einige Punkte erweitert werden.

Ein Lösungsvorschlag für Ad-hoc Netzwerke, die mit dem reaktiven DSR Routingprotokoll arbeiten, ist in [BMJ99] beschrieben. Eine Erweiterung, um ein Ad-hoc Netz mit dem proaktiven DSDV Routing über Mobile-IP anzubinden, wurde von Lei vorgeschlagen [LP97]. Diese Erweiterung wird nun kurz vorgestellt.

Dabei wird der FA als Übergangsstation zwischen dem Internetrouting und dem Ad-hoc Netzwerk genutzt. Der FA muss daher beide Routingprotokolle unterstützen und die Paketformate entsprechend konvertieren. Bevor die Stationen in einem Ad-hoc Netz eine Internetverbindung etablieren können, müssen sie einen FA in ihren Netzwerk suchen und ihn dazu veranlassen, einen Tunnel zum HA aufzubauen. Deswegen wurde eine Methode definiert, wie eine Station im Ad-hoc Netzwerk einen FA finden kann. Dazu bietet sich eine Erweiterung der bereits vorhandenen Routingtabelle an. In der vorgeschlagenen Erweiterung erhält die Tabelle einen neuen Eintrag, der den FA besonders kennzeichnet. Auf diese Weise empfangen die Stationen über die normalen Routingbotschaften auch die Information über den FA und können sich direkt bei ihm anmelden.

Nach der Anmeldung stehen einer Station alle Funktionen eines MH zur Verfügung. Der MH setzt seine Defaultroute auf den FA und erreicht damit, dass von ihm gesendete Pakete über den FA in das Internet umgesetzt werden. Pakete an den MH werden entsprechend über den HA an den FA getunnelt, und dort in das Ad-hoc Netz umgesetzt.

Die Anbindung über Mobile-IP ist eine Möglichkeit, um eine Internetanbindung herzustellen. Die hier vorgestellte Lösung bemüht sich, den bereits existierenden Standard weitgehend beizubehalten. Deshalb ist der Aufbau einer entsprechenden Infrastruktur für jede Station (der notwendige HA) und für die einzelnen Anschlußpunkte über FAs erforderlich. Für die meisten Anwender ergibt sich dabei das Problem, keine eigene Internetadresse zu besitzen und auch über keinen HA zu verfügen. Außerdem sind die erwähnten Sicherheitsprobleme so gravierend, dass hier eine Verbesserung erzielt werden muss. Aus diesem Grund werden im nächsten Abschnitt Konzepte zur Internetanbindung vorgeschlagen, die diese Mängel beheben.

8.3 Anbindungskonzepte

Da – wie gerade erläutert – beim Einsatz von Mobile-IP die Anbindung von Ad-hoc Netzwerken nicht ohne Probleme möglich ist, wird hier ein neuer Ansatz vorgestellt, der Ad-hoc Netzwerke als eigenständige Subnetze anbindet.

In diesem Abschnitt werden nun zwei Verfahren aus der Netzwerktechnik vorgestellt, die üblicherweise zur Abschirmung von Netzwerken gegen Angriffe aus dem Internet genutzt werden. Dabei handelt es sich um ein Verfahren zur Netzwerk-Adress-Translation (NAT), das zum Maskieren von IP-Adressen benutzt wird, um Rechner eines ganzen Netzwerkes hinter einer einzigen IP-Nummer zu verstecken. Zusätzlich werden die Virtuellen Privaten Netzwerke (VPNs) vorgestellt, die durch verschlüsseltes Tunneln eine sichere Kommunikation ermöglichen.

Diese Protokolle werden im Folgenden dazu verwendet, um Ad-hoc Netzwerke um einen Internetzugang zu erweitern. Dazu wird in einem ersten Schritt ein IPv4 konformes Ad-hoc Netzwerk vorgestellt. Das Routing in diesem Netzwerk erfolgt über IP-Nummern und ein zusätzlicher Dienst ermöglicht die Zuordnung von Rechnernamen zu den IP-Nummern. Ein solches Ad-hoc Netzwerk erscheint für die Applikationen dann wie ein kleines, privates IPv4 Netzwerk.

Im nächsten Schritt wird dem Netzwerk ein Internetanschluss über einen maskierenden Router hinzugefügt. Damit sind alle Stationen des Ad-hoc Netzwerks in der Lage, Verbindungen in das Internet aufzubauen. Diese Art der Internetanbindung erlaubt aber keine Verbindungen aus dem Internet an die Stationen. Außerdem besteht kein Schutz gegen Abhören oder Manipulieren der ins Internet gesendeten Daten.

In der zweiten Konfiguration wird eine Lösung für die Sicherheitsprobleme erläutert. Dafür wird der Router nicht mehr zur Maskierung, sondern als eine Art Home Agent genutzt. Die Verbindungen aus dem Ad-hoc Netzwerk an diese Stationen werden über verschlüsselte Tunnel realisiert. Damit wird die Sicherheit für die Internetkommunikation der Ad-hoc Stationen gewährleistet.

8.3.1 IPv4 konformes Ad-hoc Netzwerk

Um in einem Ad-hoc Netzwerk ein dem IPv4 Standard entsprechendes Routing zu implementieren, muss jede Station einen kompletten TCP/IP-Netzwerkstack erhalten. Die Spezifikation verlangt die Zuordnung einer eindeutigen IP-Nummer zu jeder Station. Für nicht mit dem Internet verbundene Netzwerke sind dafür die Nummernbereiche der privaten Netzwerke reserviert. Im hier vorgestellten Beispiel werden IP-Nummern im Bereich 10.X.X.X eingesetzt, da dieser Nummernbereich mit ca. 16 Millionen Adressen der größte verfügbare private Nummernbereich ist.

Die Stationen müssen ihre IP-Nummern zugeteilt bekommen, um die Eindeutigkeit zu gewährleisten. Idealerweise wird dazu eine zentrale Stelle geschaffen, die Nummern sequenziell vergibt. Eine Nummernzuordnung durch *zufällig* ausgewählte Adressen ist auch möglich, dabei entsteht jedoch das Risiko von Doppelbelegungen, die den Routingvorgang erheblich stören.

Sind diese Voraussetzungen für die Kommunikation über IPv4 erfüllt, dann kann die Ad-hoc Routenberechnung und die Nachbarerkennung als Systemprozess auf jeder Station gestartet werden. Die Prozesse senden in regelmäßigen Abständen ihre Kennungen zur Nachbardetektion per Broadcast aus. Wird ein neuer Nachbar entdeckt, dann tauschen die Nachbarn, gemäß der Spezifikation in Kapitel 5.3.3, ihre aktuellen Routingtabellen aus. Aus den gespeicherten Tabellen errechnet dann jede Station die aktuellen Routen zu allen Zielen. Diese Routen werden als Hostrouten in die Routingtabelle des IPv4-Stacks eingetragen. Eine Defaultroute wird nicht eingetragen. Dadurch leitet der IP-Stack nur Pakete an bekannte Ziele weiter. Ein Paket für ein unbekanntes Ziel löst dagegen eine Fehlermeldung an den Absender aus. Das weitere Vorgehen der einzelnen Stationen folgt der Algorithmusbeschreibung von Kapitel 5.3.3.

Die vorgestellte Konfiguration ermöglicht es den Stationen über IPv4 Pakete an alle erreichbaren Ziele zu versenden. Jeder Netzwerkstack leitet die Pakete automatisch entlang der eingetragenen Hostrouten weiter bis zum Ziel. Fehlerhafte Routen werden erkannt und durch entsprechende Botschaften angezeigt.

In dieser Konfiguration sind die Stationen lediglich über ihre IP-Nummer, aber nicht über Namen erreichbar. Praktisch alle Applikationen sprechen die Stationen aber über ihre Namen an. Im Internet wird durch den zentralen Domain Name Service (DNS) ein Name in eine IP-Nummer gewandelt und anschließend die IP-Nummer für den Verbindungsaufbau genutzt. Da ein Ad-hoc Netzwerk nicht auf solche zentralen Netz-Dienste zurückgreifen kann, wird statt dessen eine lokale Lösung verwendet, die nur auf der Kooperation von Stationen basiert.

Wenn keine Infrastruktur zur Verfügung steht, muss jede Station ihren eigenen Namen bekannt machen. Eine sehr einfache, aber ressourcenfressende Variante ist das regelmäßige Fluten der Namensinformation und der zugehörigen IP Nummer. Die Namensinformation hat dabei den großen Vorteil, nicht zu veralten und braucht deswegen nur selten geflutet zu werden.

Eine wesentlich elegantere Möglichkeit ist die Erweiterung der Routingtabelle um sogenannte Servicetags. Diese Tags sind zusätzliche Datenfelder, die einem Routingtabelleneintrag direkt zugeordnet sind, und diese werden bei der Übertragung der Routingtabellen an die Nachbarn

ebenfalls mitübertragen. Für die Übertragung existiert eine Vielzahl von Kompressionsmöglichkeiten, auf die allerdings im Rahmen dieser Konzeptvorstellung nicht näher eingegangen wird.

Durch die Servicetags werden eine ganze Reihe von Zusatzdiensten ermöglicht. Die Namensverwaltung ist mit einem zusätzlichen alphanumerischen Zeichenfeld leicht zu implementieren. Da in einem Routingtabelleneintrag die IP-Nummer bereits enthalten ist, wird mit dem Zusatzfeld nur noch der Name angefügt. Damit kann jede Station für alle erreichbaren Ziele auch nach den Namen in den Tabellen suchen.

8.3.2 Anbindung durch Netzwerk-Adress-Translation

Nach der Etablierung einer IPv4-Kommunikation zwischen den Ad-hoc Stationen wird nun eine erweiterte Konzeption für die Anbindung eines derartigen Netzwerks an das Internet vorgestellt.

Da die IP-Nummern der Ad-hoc Stationen aus dem Bereich der privaten Netzwerke stammen, werden Pakete mit diesen Adressen im Internet von den dortigen Routern nicht weitergeleitet. Daher müssen die Adressen von jedem Paket beim Übergang vom Ad-hoc Netzwerk in das Internet und bei der Rückkehr in das Ad-hoc Netzwerk umgewandelt werden. Dies übernimmt die Netzwerk-Adress-Translation (NAT).

Funktionsprinzip von NAT

Durch die ständig wachsende Zahl von Rechnern im Internet entsteht eine Knappheit von frei verfügbaren IPv4 Adressen. Deswegen ist die Beantragung von IP-Adressen für große Netzwerke schwierig und auch teuer geworden. Durch Maskieren (engl. Masquerading) wird ein ganzes Netzwerk über nur eine IP-Adresse mit dem Internet verbunden. Dazu müssen allerdings die Adressen aller Pakete durch NAT umgerechnet werden. Das so verbundene Netzwerk ist nicht direkt aus dem Internet erreichbar, doch kann jede Station eines solchen Netzwerks Verbindungen in das Internet aufbauen.

Zur Realisierung eines solchen Netzwerks wird am Übergangspunkt zum Internet ein Router mit NAT-Fähigkeit installiert. Dieser Router rechnet sämtliche Netzwerkadressen in den vermittelten Paketen so um, dass für die Stationen im Internet der Eindruck entsteht, direkt mit dem Router zu kommunizieren. Deswegen benötigt nur der Router eine im Internet gültige Netzwerkadresse.

Die Umrechnung von Netzwerkadressen ist nur für Protokolle möglich, die eine zusätzliche Kennzeichnung der Verbindung in jedem Paket mitsenden. Diese Informationen werden regulär dazu verwendet, um die Pakete von mehreren, gleichzeitigen Kommunikationsvorgängen zwischen zwei Stationen unterscheiden zu können. Die beiden gebräuchlichsten Protokolle im Internet – also TCP und UDP – verwenden Portnummern für diese Unterscheidung und sind somit auch unter NAT nutzbar.

Der NAT-Router betrachtet immer ein Adresspaar, das sich aus der IP-Adresse und der Portnummer zusammensetzt und setzt die Pakete mit Hilfe einer Tabelle um:

$$< Quellen-Adresse, Quellen-Portnummer, ZielIP-Adresse, Ziel-Portnummer >$$

$$\Updownarrow$$

$$< RouterIP-Adresse, Router-Portnummer, ZielIP-Adresse, Ziel-Portnummer >$$

Für jedes aus dem maskierten Netzwerk in das Internet gesendete Paket sucht der Router in seiner Tabelle den passenden Eintrag mit Quellen IP-Adresse, Quellen-Portnummer, Ziel IP-Adresse und Ziel-Portnummer und ersetzt die Quellen-Adresse und Quellen-Portnummer im Paket durch seine Internet IP-Adresse und eine Router-Portnummer aus der Tabelle. Findet er keinen passenden Eintrag, dann wird einfach ein neuer Eintrag erzeugt, wobei eine noch nicht belegte Portnummer gefunden werden muss.

Für Antwort-Pakete, die aus dem Internet beim NAT-Router ankommen, verläuft die Ersetzung der Paketeinträge entsprechend umgekehrt. Hier muss allerdings ein passender Tabelleneintrag vorhanden sein. Es ist nicht möglich, bei Bedarf einen Tabelleneintrag zu erzeugen.

Durch die Adressumrechnung werden die Adressräume des internen Netzwerks und des Internets voneinander getrennt. [1] Vom internen Netzwerk kann weiterhin voll auf das Internet zugegriffen werden. Vom Internet aus ist hingegen kein Aufbau einer Verbindung mehr möglich, da nur die IP-Adresse des Routers angesprochen werden kann. Diese Eigenschaft ist in den meisten Fällen sogar gewünscht, damit die Stationen im internen Netzwerk vor Angriffen aus dem Internet geschützt sind. Wenn ein Verbindungsaufbau vom Internet zu bestimmten Stationen im internen Netzwerk aber unbedingt erforderlich ist, so kann dies durch feste Tabelleneinträge im NAT-Router explizit ermöglicht werden.

Abbildung 8.2 zeigt ein Beispiel für das Vorgehen beim Einsatz von NAT. Die Stationen im internen Netzwerk erhalten die für private Netzwerke vorgesehenen IP-Adressen ($10.X.X.X$). Diese privaten Adressen werden von Internet-Routern deswegen auch nicht weitervermittelt. Der NAT-Router besitzt im internen Netzwerk einen Anschluss mit einer Adresse aus diesem Bereich (hier 10.1.1.1) und einen Internetzugang mit einer gültigen IP-Adresse, wie z.B. 141.99.131.72. Sendet eine Station, die im maskierten Netz die Nummer 10.1.1.3 benutzt, ein Paket an eine Station im Internet mit der Nummer 131.99.10.4, dann wird zuerst ein Paket erzeugt, das folgende Einträge enthält:

AbsenderIP	Absenderport	ZielIP	Zielport
10.1.1.3	55123	131.99.10.4	80

[1] Dadurch wird die Anzahl der im Internet verwaltbaren Rechner zwar größer, aber der insgesamt ansprechbare Adressraum bleibt unverändert, da immer nur 32 Bit-Adressen benutzt werden. Am Router wird definiert, welche Adressen zum internen Netzwerk oder zum Internet gehören. Einige IP-Adressbereiche sind speziell für diese Mehrfachnutzung vorgesehen.

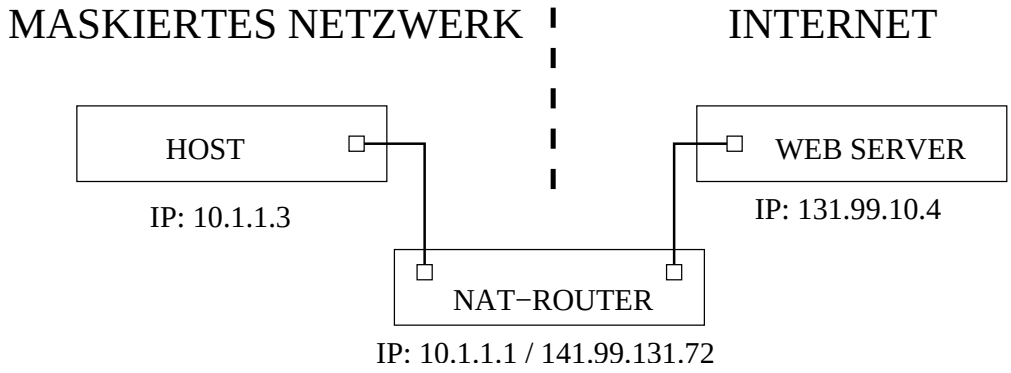


Abbildung 8.2: Netzwerk Adress Translation

Dieses Paket durchläuft den NAT-Router, der die Absenderadresse durch seine eigene Adresse ersetzt und eine freie Portnummer einsetzt, bevor er das Paket in das Internet ausliefert:

AbsenderIP	Absenderport	ZielIP	Zielport
141.99.131.72	6321	131.99.10.4	80

Der angesprochene Rechner antwortet mit einem Paket, das folgende Adressen enthalten muss:

AbsenderIP	Absenderport	ZielIP	Zielport
131.99.10.4	80	141.99.131.72	6321

Wenn das Paket am NAT Router eintrifft, verändert dieser die Zieladresse und den Zielport bevor es in das interne Netzwerk weitergesendet wird:

AbsenderIP	Absenderport	ZielIP	Zielport
131.99.10.4	80	10.1.1.3	55123

Das Paket erreicht in dieser Form den internen Rechner, der überhaupt nicht bemerkt hat, dass seine Pakete umadressiert wurden. NAT ist daher für die Stationen ein transparentes Protokoll. Das ist für den Einsatz in Ad-hoc Netzwerken von großem Vorteil, da mit NAT der Übergang ins Internet selbstkonfigurierend ist. Dieser Vorteil ist durch keine andere Anbindungsvariante zu erreichen.

Anbindung über NAT

Nachdem das Funktionsprinzip von NAT erläutert wurde, wird dieses Verfahren nun dazu genutzt, ein Ad-hoc Netzwerk als Subnetz an das Internet zu koppeln. Der Vorgang wird anhand des folgenden Beispiels erläutert.

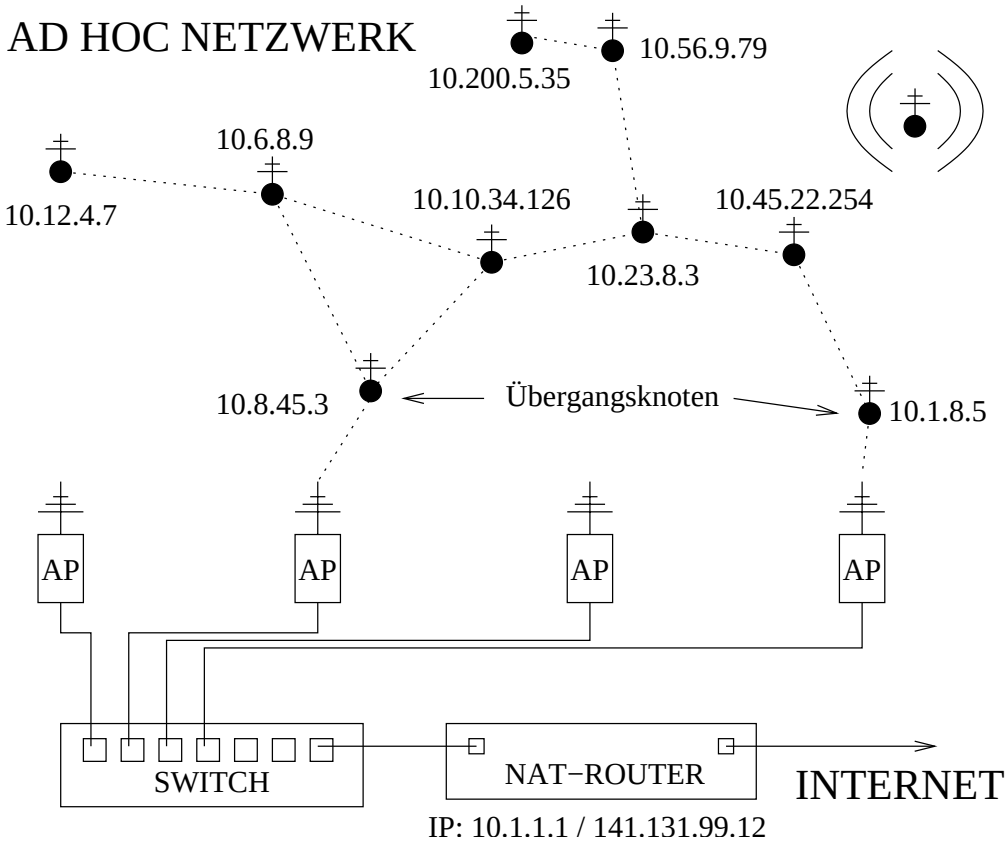


Abbildung 8.3: Konzept zur Anbindung von Ad-hoc Netzwerken

Die Abbildung 8.3 veranschaulicht die Konfiguration einer solchen Anbindung. Im oberen Teil des Bildes ist ein kleines Ad-hoc Netzwerk dargestellt. Im unteren Teil ist ein NAT-Router dargestellt, der in diesem Beispiel vier Antennen als Anschlusspunkte (AP) verwaltet, die über einen Switch mit dem Router zusammengeschaltet sind. Der NAT-Router verhält sich auf der dem Ad-hoc Netzwerk zugewandten Seite wie eine normale Station, und er besitzt auch eine entsprechende IP-Nummer (hier 10.1.1.1). Er besitzt zusätzlich einen Anschluss an das Internet und verwendet dafür eine im Internet gültige IP-Adresse.

Erreichen den NAT-Router Pakete aus dem Ad-hoc Netz, die an ein Ziel im Internet adressiert sind, dann rechnet der NAT-Router die Adressen im Paket mit der oben beschriebenen Methode um und leitet sie weiter. Ebenso werden aus dem Internet ankommende Pakete an das Ad-hoc Netzwerk weitergeleitet, wenn der NAT-Router diese einer Verbindung zuordnen kann. Aus diesem Grund müssen Verbindungen immer aus dem Ad-hoc Netz heraus aufgebaut werden.

Die Stationen im Ad-hoc Netz müssen zum Aufbau einer Internetverbindung zuerst einen NAT-Router in ihrer Umgebung lokalisieren. Dazu kann wieder ein Servicetag benutzt werden. Ein NAT-Router hängt an seinen Routingtabelleneintrag ein kleines Datenfeld an, das ihn als Internetrouter kennzeichnet. Die Stationen im Ad-hoc Netzwerk sind damit in der Lage, aus ihren Tabellen den nächstgelegenen Router herauszusuchen. Die Station setzt daraufhin ihr Defaultroute auf den Nachbarn, der dem Internetrouter am nächsten ist. Damit werden die Pakete mit Internetadressen an den NAT-Router geleitet.

Bewertung der Anbindung über NAT

Das hier vorgestellte Konzept auf der Basis von NAT bietet einen selbstkonfigurierenden Internetzugang, der außer dem NAT Router keine weitere Infrastruktur, wie beispielsweise einen Home Agent, benötigt. Diese Lösung erlaubt den Ad-hoc Stationen eine Verbindung in das Internet aufzubauen, allerdings sind die Stationen vom Internet aus nicht erreichbar. Diese Einschränkung trifft aber auf alle Lösungen zu, die ohne festen Vermittler arbeiten, auf diese Konsequenzen wurde bereits in Abschnitt 8.2.2 hingewiesen.

Ein noch ungelöstes Problem ist die mangelhafte Sicherheit in diesem Netzwerk. Alle im Ad-hoc Netzwerk gesendeten Pakete sind unverschlüsselt und werden über eine Kette von Stationen weitergereicht. Dies eröffnet viele Möglichkeiten zum Abhören und Manipulieren der weitergeleiteten Daten. Da die Ad-hoc Adressen durch NAT maskiert sind, kann ein Internetrechner nicht erkennen, mit welcher Ad-hoc Station er kommuniziert, er erkennt nur die IP-Adresse des NAT-Routers. Der NAT-Router selbst hat kaum eine Kontrollmöglichkeit über die umgesetzten Pakete. Es lassen sich zwar Einschränkungen hinsichtlich der erreichbaren Adressen etc. konfigurieren, aber eine effektive Zugangskontrolle fehlt. Diese ist notwendig, um den Zugang zum Internet auf vertrauenswürdige Stationen des Ad-hoc Netzwerks zu beschränken.

8.3.3 Anbindung durch virtuelle private Netzwerke

Im zweiten Konzept werden die Ad-hoc Stationen über gesicherte Tunnel mit dem Internet verbunden. Dazu wird der NAT-Router durch einen VPN-Router ersetzt. Dadurch lassen sich Verbindungen in das Internet nur noch über gesicherte Tunnels aufbauen.

Funktionsprinzip von VPN

Private Netzwerke definieren eine abgesicherte Zone innerhalb eines Netzwerks oder ein eigenständiges Netzwerk, zu der Außenstehende keinen Zugang erlangen können. Diese Technik wird auch von anderen Anbietern, wie z.B. der Telekom, unter dem Begriff „geschlossene Benutzergruppe“ angeboten. Eine umfangreiche Beschreibung dieser Technologie ist in [SWE98] zu finden.

Die VPN erweitern das Konzept der privaten Netzwerke, indem sie mehrere, voneinander unabhängige, private Netzwerke über eine sichere Verbindung zusammenschalten und so ein grosses privates Netzwerk bilden. Diese Anwendung ist besonders für Firmen mit mehreren Außenstellen interessant, da sich damit auch die Stationen der Außenstellen bequem zu einem großen abgesicherten Netzwerk integrieren lassen.

Die Verbindung der einzelnen privaten Netzwerke erfolgt über spezielle Tunnel, die jeweils zwischen zwei VPN-Routern aufgebaut werden. Ein VPN Router muss nicht unbedingt ein eigenständiges Gerät sein. Für die Anwendungen in Ad-hoc Netzwerken ist es auch möglich, dass eine einzelne Station für sich selbst die Routerfunktion übernimmt. Dabei verbindet sich die Station über einen Tunnel mit einem VPN-Router und erhält damit eine künstliche zweite Netzwerkverbindung über den VPN-Router. Die Station kann dann sowohl das ursprüngliche Netzwerk nutzen als auch gleichzeitig über das VPN kommunizieren.

Das Konzept der Tunnel wurde bereits in Zusammenhang mit Mobile-IP angesprochen, für VPN Anwendungen müssen die Tunnel allerdings eine Sicherheit gegen Abhören und Manipulationen bieten. Beim Tunneln mit VPN wird ein Paket, das von einem privaten Teilnetz in ein anderes privates Teilnetz übertragen werden muss, zuerst verschlüsselt, dann in ein Tunnelpaket eingepackt (gekapselt) und zum VPN-Router am anderen Ende des Tunnels geschickt, der das Paket wieder auspackt, entschlüsselt und ausliefert.

Da die Tunnelpakete bei vielen Anwendungen durch das Internet geleitet werden, ist die Absicherung der Paketinhalte am einfachsten durch eine sogenannte Shared-Key-Verschlüsselung zu erreichen [Sch94]. Die beiden VPN-Router gehören in der Regel zur gleichen Organisation, damit kann vorab ein Schlüssel zwischen den Routern vereinbart werden, mit dem anschließend die gesamte Kommunikation durch den Tunnel geschützt wird. Durch die Verschlüsselung sind die Tunnel gleichzeitig auch gegen Manipulationen an den Paketen und vor der Einspeisung fremder Pakete geschützt.

Für die Anbindung von Ad-hoc Netzwerken sind VPNs besonders interessant, da innerhalb des VPN jeder Station eine dauerhafte IP-Adresse zugewiesen wird, die wie beim Mobile-IP

unabhängig vom Standort der Station ist. Der zweite besondere Vorteil ist die vollständige Verschlüsselung der Kommunikation. Kommuniziert die Station über ihre dauerhafte IP-Adresse sind alle Pakete bis zum Tunnellende verschlüsselt. Insbesondere sind damit alle Übertragungen über die drahtlosen Verbindungen des Ad-hoc Netzwerkes vor Abhören geschützt.

Anbindung über VPN

Um ein Ad-hoc Netz über VPN mit dem Internet zu koppeln, ist im einfachsten Fall nur ein VPN Router am Übergang vom Ad-hoc Netzwerk zum Internet erforderlich. In dem hier beschriebenen Konzept wird allerdings das VPN Prinzip in einer ungewöhnlichen Weise angewendet, da das Ad-hoc Netzwerk als die gefährliche Zone betrachtet wird und folglich das Internet wie das private Netzwerk behandelt wird. Demnach führen die geschützten Tunnel durch das Ad-hoc Netzwerk bis zum VPN Router, der die Pakete dann in das Internet umsetzt.

Damit haben die Ad-hoc Stationen nun nur noch die Möglichkeit, ihre Internerverbindungen über einen gesicherten Tunnel zum VPN-Router aufzubauen. Der VPN-Router erlaubt aber nur den berechtigten Stationen den Tunnelaufbau. Dazu müssen vorab geheime Schlüssel zwischen dem VPN-Router und jeder berechtigten Station ausgetauscht werden. Beim Verbindungsaufbau kontrolliert der VPN-Router, ob die Station den Schlüssel kennt und akzeptiert dann den Tunnelaufbau.

Ist der Tunnel etabliert, dann erhält die Station vom VPN-Router ihre eigene IP-Adresse zugewiesen, die auch im Internet gültig ist. Damit ist die Station aus dem Internet erreichbar und kann auch gleichzeitig mit dem Ad-hoc Netzwerk über ihre erste IP-Nummer kommunzieren. Auch in diesem Beispiel werden Ziele im Ad-hoc Netzwerk über Hostrouten erreicht, während für die Internetpakete die Defaultroute zuständig ist und dazu über den Tunnel den VPN-Router nutzt.

Im folgenden Beispiel soll eine Ad-hoc Station eine gesicherte Verbindung ins Internet erhalten. Diese Konfiguration ist in Abbildung 8.4 dargestellt. Die Station mit der Nummer 10.10.34.126 hat einen Tunnel zum VPN-Router aufgebaut. Der Tunnel ist in der Abbildung durch einen dicken Pfeil symbolisiert. Sobald ein Tunnel besteht, erfüllt der VPN-Router die gleiche Funktion wie ein Home Agent und leitet alle Pakete für die IP-Adresse 131.99.12.5 an die Station 10.10.34.126 weiter. Die Station erhält damit eine zusätzliche, voll gültige IP-Adresse zur Kommunikation mit dem Internet. Sie ist in dieser Konfiguration auch aus dem Internet erreichbar.

In diesem Beispiel ist die gesamte Kommunikation der Station 10.10.34.126 mit dem Internet bis zum VPN-Router durch den verschlüsselten Tunnel geschützt. Da die vermittelnden Stationen im Ad-hoc Netz nur die verschlüsselten Tunnelpakete erhalten ist ein Abhören oder eine Manipulation der Internetkommunikation ausgeschlossen. Die vermittelnden Stationen sind nicht einmal in der Lage, herauszufinden, mit welchen Stationen im Internet kommuniziert wird. Der sichere Tunnel endet allerdings am VPN-Router. Danach laufen die Pakete wieder unverschlüsselt und ungeschützt durch das Internet.

Nur Stationen, die eine Zugangskennung zum VPN-Router besitzen, erhalten einen Tunnel und

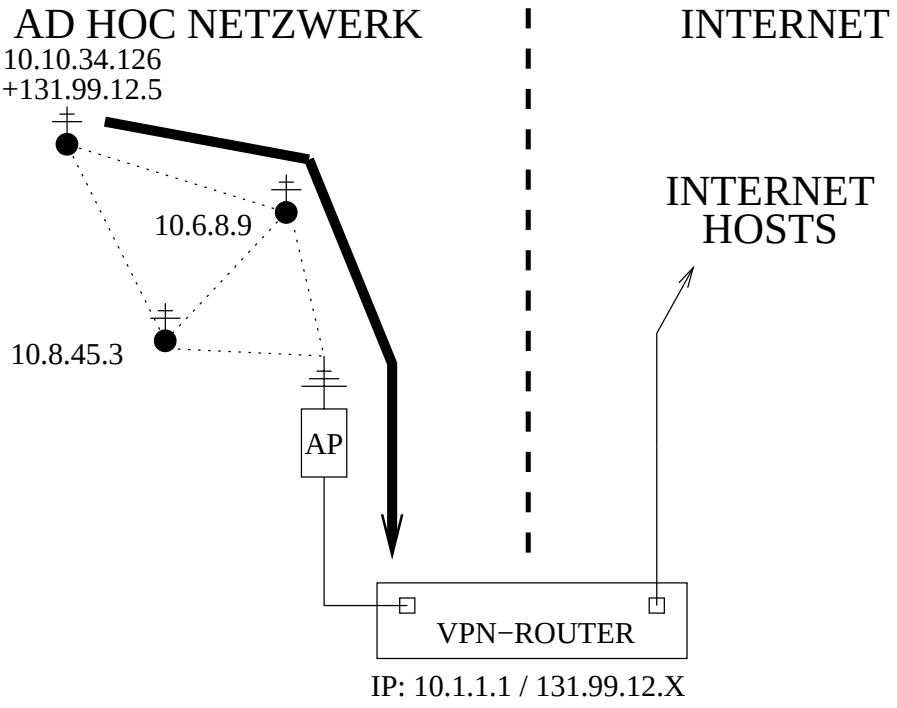


Abbildung 8.4: Absicherung der Anbindung

damit eine im Internet gültige Adresse. Die Sicherheit für den Internetzugang ist damit gesichert, denn eine Aufzeichnung der zugewiesenen Internetadressen erlauben dem Administrator des VPN-Routers eine Identifizierung der einzelnen Nutzer einer Internetverbindung. Diese Daten werden aber überwiegend zur Abrechnung des Internetzugangs verwendet. Denn mit großer Wahrscheinlichkeit endet beim Internetzugang die kostenlose Netznutzung. Entsprechende Vorschläge zur Abrechnung und Bezahlung von Zugängen für drahtlose Netzwerke wurden bereits vom Autor und einem Studenten, im Rahmen einer Semesterarbeit, entwickelt und veröffentlicht [Dem99, PJ97], werden aber hier nicht weiter behandelt.

8.4 Zusammenfassung

Dieses Kapitel zeigte, wie sich Ad-hoc Netzwerke mit dem Internet verbinden lassen. Das standardmäßig verfügbare Routing im Internet ist nicht in der Lage, mobile Stationen oder Ad-hoc Netzwerke zu unterstützen. Dafür existieren eine Reihe von Protokollen, die Pakete an beliebige Orte umleiten oder IP-Adressen umrechnen können. Zu Beginn des Kapitels wurde deshalb das Funktionsprinzip eines Tunnels und Mobile-IP erläutert, mit dem auch portable Rechner das Internet nutzen können. Es wurden allerdings auch einige Schwächen des Protokolls beschrieben, insbesondere die vorab nötigen Konfigurationsarbeiten und die Sicherheitsprobleme erschweren den Einsatz von Ad-hoc Netzwerken in Kombination mit Mobile-IP.

Als Alternative zu Mobile-IP wurde die Netzwerk-Adress-Translation vorgestellt, die durch

Adressumrechnungen eine Netzwerkanbindung über einen speziellen Router anbietet. Durch NAT können die Stationen eines Ad-hoc Netzwerks ohne langwierige Anmeldung und Konfiguration mit dem Internet kommunizieren. Allerdings erlaubt NAT keine Verbindungen aus dem Internet an die Stationen und bietet wegen der fehlenden Konfigurationsmöglichkeiten auch keine Sicherheitsfunktionen an.

Zum Abschluss des Kapitels wurde eine besonders sichere Anbindung über sogenannte VPN-Router vorgestellt. Diese Router nutzen verschlüsselte Tunnel, um beliebige Stationen zu einem virtuellen Netzwerk zu verbinden. Die Sicherheit erfordert dann zwar eine Anmeldung jeder einzelnen Station, um eine Identifikation zu ermöglichen und die nötigen Schlüssel auszutauschen, aber die dadurch erreichte Verschlüsselung bietet eine in Ad-hoc Netzwerken dringend notwendige Sicherheit gegen Abhören. Zusätzlich verhindert die Verschlüsselung die Übernahme von Verbindungen durch andere Stationen und verbirgt sogar die Identität der Internet-Kommunikationspartner der Ad-hoc Stationen.

Anhand der vorgestellten Lösungen läßt sich bereits ein Problem erkennen, das in zukünftigen selbstkonfigurierenden Netzwerken häufig auftreten wird. Ein vollständig selbstkonfigurierendes Netzwerk bemüht sich, möglichst viele Kommunikationsmöglichkeiten zu eröffnen. Dies ist allerdings in vielen Fällen aus Sicherheitsgründen nicht erwünscht. Hier sind für die Zukunft noch Konzepte zu entwickeln, die möglichst selbstständig zwischen eigenen und fremden Stationen unterscheiden können. Auf der Basis dieser Informationen lassen sich dann neue Routingalgorithmen entwerfen, die auch die dringend benötigten Sicherheitsfunktionen anbieten können.

Kapitel 9

Zusammenfassung und Ausblick

In dieser Arbeit wurden Routingalgorithmen für kooperative, mobile Paketfunknetzwerke vorgestellt, die auch als Ad-hoc Netzwerke bezeichnet werden. Dieser Netzwerktypus ist bislang wenig verbreitet. Die dafür nötige Funkhardware hat sich erst in den letzten Jahren, besonders durch den enormen Erfolg der Mobilfunknetze, zu kleinen und günstigen Geräten weiterentwickelt. Inzwischen sind leistungsfähige Paketfunksysteme mit nur einem Chip zu realisieren. Dies erlaubt bereits die serienmäßige Integration solcher Geräte in Laptops und Handhelds. Die Hardware-Voraussetzungen zum Aufbau von Ad-hoc Netzwerken sind damit vorhanden. Bisher werden diese Funksysteme aber meistens nur für die direkte Übertragung verwendet, das Funksystem wird so nur als Kabelersatz genutzt. Dabei ist über das Funksystem, nur durch Einsatz intelligenter Software, auch ein Ad-hoc Netzwerk mit Internetfähigkeit zu realisieren. Diese Arbeit untersuchte nun speziell die Routingprobleme, die bei der Implementierung und dem Betrieb von Ad-hoc Netzwerken entstehen und stellte neue Algorithmen sowie einige Verbesserungen für bestehende Verfahren vor.

Am Anfang der Arbeit stand eine Einführung in die Funkkommunikation und eine Erläuterung der geschichtlichen Entwicklung dieser Technologie, von den ersten Funksystemen, die noch mit analoger Technik arbeiteten, bis hin zur Entwicklung von Paketfunksystemen. Die Technik der Paketfunksysteme wurde anschließend ausführlich dargestellt. Um die Zusammenhänge zu verdeutlichen, wurde dazu das OSI-Modell beschrieben und die dort verwendete Schichtenstruktur am Beispiel der Paketfunknetze erläutert. Anschließend wurden die benötigten Begriffe und Definitionen aus der Graphentheorie für die später folgenden Algorithmusdefinitionen festgelegt.

Danach wurde die Funktionsweise von Routingalgorithmen ausführlich beschrieben. Es begann mit der Vorstellung der beiden gängigen Algorithmen zur Berechnung kürzester Pfade, die als Distanz-Vektor und als Link-State Algorithmen bekannt sind. Dabei wurde auch gezeigt, warum diese Standardalgorithmen nicht ohne Modifikationen in Ad-hoc Netzwerken eingesetzt werden können.

Anschließend wurden die speziellen, für Ad-hoc Netze verwendbaren, Routingalgorithmen beschrieben und klassifiziert. Die Klassifizierung unterteilt die Algorithmen nach ihrem Funkti-

onsprinzip in hierarchische und nicht hierarchische Verfahren, die weiter – in Bezug auf die verwendeten Suchverfahren für die Routen – untergliedert wird. In der Beschreibung der Routingalgorithmen wurden die Vor- und Nachteile der vorhandenen Algorithmen oder ganzer Algorithmengruppen erläutert. Dies führte schließlich zu einer Darstellung der in Routingalgorithmen enthaltenen Informationsmenge. Mit Hilfe dieser Darstellung wurde verdeutlicht, dass die verfügbare Information über die Netztopologie die Möglichkeiten eines Routingalgorithmus bestimmt. Diese Information muss aber gesammelt und im Netzwerk verbreitet werden, wodurch eine andauernde Belastung des Netzwerks entsteht. Für einen bestimmten Einsatzzweck ist daher jedes Mal abzuwägen, ob der gewünschte Funktionsumfang die dadurch entstehenden Belastungen rechtfertigt.

Anschließend wurde die Analyse von Routingalgorithmen besprochen. Dabei wurden die Vor- und Nachteile von theoretischen Methoden mit den praktischen Methoden verglichen, zu denen auch die Simulation gehört. Die Analyse per Simulation zeigte dabei die meisten Vorteile, sowohl hinsichtlich der Analysemöglichkeiten als auch bei der Betrachtung des nötigen Aufwandes. Aus diesem Grund wurden die Algorithmenvergleiche dieser Arbeit durch Simulationen erzeugt. Für die Simulation von Ad-hoc Netzwerken war ein geeignetes Netzwerkmodell erforderlich, das definiert und durch theoretische Betrachtungen auf seine Tauglichkeit untersucht wurde. Danach wurde die Architektur des Simulators vorgestellt, und es wurde erläutert, warum eine zweistufige Simulation für die durchgeführten Vergleiche die beste Variante darstellt. Die erste Simulationsstufe berechnet dabei die physikalischen Bewegungen der Stationen und legt so die entstehende Netztopologie fest. Die zweite Stufe nutzt diese Topologie und berechnet nur noch die Kommunikationsvorgänge im simulierten Netzwerk. Durch diese Simulationsart wurden mehrfache Berechnungen der Topologie eingespart, und die Algorithmenvergleiche fanden immer auf der Basis identischer Netztopologien statt.

Nach der Vorstellung der Testwerkzeuge wurde detailliert das Routing durch Pfadsuche beschrieben. Routing durch Pfadsuche ist in der oben erwähnten Einteilung, nach der zur Verfügung stehenden Informationsmenge, zwischen Distanz-Vektor und Link-State eingeordnet. Dies machte die Pfadsuche zum idealen Ausgangspunkt für die Entwicklung eines möglichst universellen Ad-hoc Routingalgorithmus.

Die Nachteile der bisher verfügbaren Pfadsuche zeigten sich bei der Spezifikation des Algorithmus. Die zunächst vorgestellte einfache Pfadsuche erfordert eine umfangreiche formale Spezifikation, und die Vorgänge im Netzwerk sind schwer nachzuvollziehen. Aus diesem Grund wurde eine verbesserte Pfadsuche vorgestellt, sowie deren Funktionsprinzip erläutert. Diese Pfadsuche wurde „Tree Exchange Routing Algorithm“ (TERA) genannt.

Anschließend wurde eine Vergleichssimulation durchgeführt, um die Unterschiede der beiden Implementierungen zu untersuchen. Dabei waren in einem statischen Netzwerk kaum Unterschiede zwischen den beiden Algorithmen hinsichtlich der Erreichbarkeit festzustellen. Merkbare Unterschiede traten erst bei ansteigender Dynamik im Netzwerk auf, dabei zeigte TERA dann eine deutliche höhere Erreichbarkeit.

Bereits in der Arbeit wurden vielfach Hinweise gegeben, wie eine weiterführende Untersuchung die Leitungsfähigkeit der Algorithmen ausbauen könnte. Ein erstes Ziel für eine verbesserte

Bandbreitennutzung ist sicherlich eine optionale Komprimierung der Tabellen vor der Übertragung. In Abschnitt 5.3.2 wurden dazu Ideen vorgestellt, um die Tabellen effektiver übertragen zu können. Diese Weiterentwicklung wird besonders wichtig, wenn die Routingtabellen zusätzlich Netzwerkdienste, wie die Namensauflösung, unterstützen müssen.

In der Arbeit wurde dargelegt, wie durch die Zusammenführung der Aufgaben mehrerer OSI-Schichten die Effektivität eines Algorithmus zu steigern ist. Ein Beispiel dafür ist das adaptive Verfahren zur Nachbarschaftserkennung. Dabei passt sich der Algorithmus automatisch an die Bewegungsgeschwindigkeiten der jeweils eigenen Station an. Durch Simulationen wurde gezeigt, dass die automatische Anpassung in jeder Station ein Netzwerk ohne Veränderungen erkennen kann und dann seine Versuche, neue Nachbarn zu finden, drastisch reduziert. Dies spart Bandbreite und Energie ein.

Die adaptive Nachbarschaftserkennung ist nur ein Beispiel dafür, wie ein Algorithmus zusätzliche Informationen über die Umgebung gewinnen und nutzen kann. Die adaptive Nachbarschaftserkennung ermöglicht eine Abschätzung der eigenen Knotenmobilität in Relation zu den Nachbarn. Im vorgestellten Verfahren wurde die Information nur dazu genutzt, um die Nachbarerkennung selbst zu steuern. Die Information kann aber noch wesentlich effizienter verwendet werden, wenn sie z.B. dem Routingalgorithmus zur Verfügung gestellt wird und sie dort Einfluss auf die Routenwahl hat. Da jede Station mit dem Aussenden der Kennungen auch ihre eigene Mobilitätsschätzung an die Nachbarn sendet, kann ein Routingalgorithmus eventuell sogar voraussehen, ob die Verbindung zu einem Nachbarn kurzfristig oder dauerhaft sein wird. Die Untersuchung der besten Routenwahl auf der Basis von Mobilitätsinformation ist ein neues Forschungsgebiet, das gegenwärtig erst erschlossen wird. Weiterführende Arbeiten sind notwendig, um beispielsweise mit der Mobilitätsinformation eine Überlastkontrolle für das Routing zu entwickeln, die extrem schnell bewegte Stationen gegebenenfalls bei hoher Netzbelastung aus der Routenberechnung herausnimmt.

Eine weitere Verbesserung, welche die Bandbreiteneffizienz steigert, wird durch sogenanntes lokales Routing möglich. Dazu wurde der TERA umkonfiguriert, so dass Pfade nicht bei jeder kleinen Topologieänderung nachoptimiert werden. Eine anschließende Simulationsreihe untersuchte dann die dadurch erreichbare Einsparung. Die Untersuchung brachte das überraschende Ergebnis, dass bis zu 50 Prozent der Botschaften nur für die Neuberechnung der kürzesten Pfade bei kleinen Topologieveränderungen verwendet werden, obwohl die Routen noch einsatzfähig sind. Mit dem hier entwickelten lokalen Routing wird ein Algorithmus präsentiert, der solche Optimierungen unterlässt und nur bei vollständig defekten Routen tätig wird. Mit dem lokalen Routing wurde aber nur eine Möglichkeit vorgestellt, wie bandbreiteneffizientes Routing gestaltet werden kann. Im vorliegenden Fall wurden die Kriterien für die kürzesten Pfade gelockert, um eine Einsparung der Botschaften zu erzielen. Eine neue Erweiterung kann beispielsweise durch den (kurzzeitigen) Verzicht auf die Schleifenfreiheit von Routen realisiert werden. Durch diesen Verzicht wird die Löschung einer defekten Route eine Zeit lang hinausgezögert, um einem lokalen Routing eine Möglichkeit zu geben, eine neue Route zu berechnen und dann nur ein Update zu senden.

Das siebte Kapitel konzentrierte sich auf Lastverteilung und Erreichbarkeitssteigerung durch Mehrwegerouting. Dazu wurde eine Erweiterung für Pfadsuchealgorithmen vorgestellt, die bei

vorhandenen parallelen Pfaden die ausgesendeten Pakete gleichmäßig auf die Pfade verteilt. Durch eine Reihe von Simulationen wurde die so erreichbare Lastverteilung für statische und dynamische Netzwerke ermittelt.

Die durchgeführten Simulationen untersuchten zunächst hauptsächlich die Realisierbarkeit von Mehrwegerouting. In weiteren Untersuchungen sollte dann geklärt werden, ob statt der gleichmässigen Verteilung eventuell ein besserer Verteilungsmechanismus existiert, und ob sich mit unterschiedlicher Verteilung eine bessere Lastverteilung erreichen lässt. Der hier verfügbare Algorithmus nutzt nur parallele Routen, die sich aus den von der Pfadsuche gesammelten Informationen berechnen lassen. Es werden keine zusätzlichen Botschaften zur Verbreitung zusätzlicher Informationen eingesetzt. Damit wird garantiert, dass die Effizienz des Routings hinsichtlich der benötigten Bandbreite nicht sinkt. Das Auffinden von weiteren Routen erfordert aber zusätzliche Botschaften. Eine zukünftige Untersuchung mit einer Reihe von weiteren Simulationen kann klären, unter welchen Umständen es sich lohnt, die Kosten der zusätzlichen Botschaften in Kauf zu nehmen.

Anschließend wurde ein Verfahren präsentiert, das mit den erweiterten Tabellen des Mehrwegerouting eine Lösung für Probleme anbietet, die durch gelegentlich auftretende Inkonsistenzen in den Routingtabellen entstehen. Mit der Lösung wurden die Pfadsuchealgorithmen so verändert, dass alle von den Nachbarn angebotenen Ziele – auch die Stationen mit inkonsistenter oder unvollständiger Pfadinformation – immer erreichbar sind. Die mit dieser Methode erzielbare Verbesserung der Erreichbarkeit wurde durch Simulationen bestätigt. Da die meisten Inkonsistenzen in den Routingtabellen durch Laufzeitunterschiede verursacht werden, ist die vorgeschlagene Verbessung besonders für Netzwerke mit einer hohen Dynamik geeignet.

Bei der Steigerung der Erreichbarkeit durch nachträgliche Auswertung auch inkonsistenter Tabellen sind durch zusätzliche Maßnahmen noch weitere Verbesserungen zu erwarten. Beispielsweise nutzt der implementierte Routingalgorithmus aus Geschwindigkeitsgründen immer die erste Routeninformation, die er findet. Eine Vorabbewertung der von den Nachbarn angebotenen Routen erlaubt aber z.B. die Tabelle mit den wenigsten Inkonsistenzen herauszusuchen.

Die letzten Simulationen dieser Arbeit untersuchten die Ausfallzeiten in dynamischen Netzwerken. Die Häufigkeit und die Länge der Ausfälle wurden ermittelt und statistisch ausgewertet. Anschließend wurde untersucht, ob durch die Nutzung paralleler Routen die Ausfallzeit reduzierbar ist. Die Simulationsergebnisse zeigten, dass bei der mehrfachen Versendung von Paketen über parallele Wege die Ausfallzeiten sinken. Die besseren Ausfallzeiten werden aber so durch eine wesentlich stärkere Netzbelastung erkauft. Es wurde jedoch dargelegt, dass schon die standardmäßig vorhandene Fehlerkorrektur, die verlorene Pakete durch wiederholtes Versenden ersetzt, von parallelen Routen profitiert.

Zuletzt wurden Verfahren zur Anbindung von Ad-hoc Netzwerken an das Internet vorgestellt. Dazu war eine kurze Einführung in Netzwerkprotokolle erforderlich, die mobile Stationen im Internet ermöglichen. Dabei wurde insbesondere das Tunneln und das darauf basierende Mobile-IP vorgestellt. Anschließend wurden noch Protokolle für private Netzwerke erläutert, die den Aufbau von abgeschirmten Subnetzen erlauben oder durch Adressenmanipulation ganze Netzwerke hinter einem Router verstecken.

Dabei wurden verschiedene Anbindungsvarianten vorgestellt. Eine Variante erlaubte die selbstkonfigurierende Anbindung von Ad-hoc Netzwerken, konnte aber nur ein unbefriedigendes Sicherheitsniveau bieten. Deswegen wurde zusätzlich eine alternative Variante beschrieben, die ein besonders hohen Sicherheitsniveau erreicht. Eine Voraussetzung für die so erzielte Sicherheit war allerdings die Anmeldung der berechtigten Nutzer eines Internetzugangs. Dies muss durch einen Administrator erfolgen. Damit konnte in dieser Variante ein vollständig selbstkonfigurierender Internetzugang nicht mehr realisiert werden.

Drahtlose Netzwerke werden in naher Zukunft wesentlich häufiger anzutreffen sein, als dies bisher der Fall gewesen ist. Die in den letzten Jahren entstandenen Standards und die Verfügbarkeit als hochintegrierte Schaltungen erlauben inzwischen den Einsatz drahtloser Netzwerke in fast jedem Bereich. Gelingt es noch, die Netzwerke so zu gestalten, dass sie keinen Administrationsaufwand benötigen, dann steht ihnen ein weites Feld für Anwendungen in Haushalten und in Firmen offen.

Die Einführung von IPv6 wird neue Wege für die Kommunikation portabler Stationen eröffnen. Die praktisch unbegrenzt verfügbare Anzahl von Internetadressen und die in IPv6 erstmals eingeführten Sicherheitsfunktionen auf Paketebene erlauben dann noch vielseitigere Lösungen. Derzeit verhindert hauptsächlich die mangelnde Verbreitung von IPv6 den Einsatz für portable Stationen.

Ein weiteres Gebiet für zukünftige Forschung ist die Integration neuer Dienste und Verbindungstypen in Ad-hoc Netzwerke. Ein Aspekt dabei – die Einführung von Diensten mit Leistungsgarantien (engl. Quality of Service) – ist die Voraussetzung für die Nutzbarkeit von Ad-hoc Netzwerken im Multimedia-Bereich. Die Implementierung dieser zusätzlichen Dienstleistungen gestaltet sich sehr schwierig, da ein Netzwerk mit mobilen Stationen und ohne Infrastruktur keine dauerhaften Leistungsgarantien geben kann. Es kann immerhin eine Reservierung der gerade verfügbaren Kapazitäten angeboten werden.

Zur Erforschung dieser neuen Netzwerke existiert bereits eine Kooperation mit der Universität Münster, mit der zusammen die Digital Inter Relay Communication (DIRC) untersucht wird. Das System ist mit Ad-hoc Netzwerken verwandt, allerdings ist es kein Paketfunknetz. Die Hauptaufgabe von DIRC ist die Bereitstellung von Verbindungen für die Telephonie, die andere Qualitätsanforderungen an das Netzwerk stellt. Deswegen werden in DIRC die Übertragungskapazitäten zwischen den Routern fest reserviert. Eine genaue Beschreibung dieses Netztypus findet sich in [Bor02, Sie97]. Im Rahmen der Kooperation wurden Routingalgorithmen und Simulationsmodelle an der Universität Siegen entwickelt [BJLF01a, BJLF01b] und inzwischen ist von der Arbeitsgruppe auch ein Patent für diesen Netzwerktyp angemeldet worden. Bisher wurden die neuen Verfahren nur mittels Simulationen untersucht, für die weitere Arbeit muss nun ein Prototyp entwickelt werden, der eine Bewertung der Verfahren in einer praxisnahen Umgebung ermöglicht.

Weitere Entwicklungen im Netzwerkbereich werden auch in Zusammenhang mit dem sogenannten *Pervasive Computing* erwartet, mit dem eine allgegenwärtige Infrastruktur für Informationsdienstleistungen aller Art aufgebaut werden soll. Die Ad-hoc Netzwerke spielen in der Planung der zukünftigen Kommunikationsstruktur für dieses Konzept eine entscheidende Rol-

le. Auch hier muss noch viel Forschungsarbeit geleistet werden, um mehrstufige, skalierbare Ad-hoc Netzwerke zu entwickeln, die beispielsweise zuerst ein kleines persönliches Netzwerk aus mitgeführten Geräten wie Handy, PDA und Laptop aufbauen und sich dann zu größeren Netzwerken für ein Gebäude, eine Firma oder eine ganze Ortschaft zusammenschließen.

Die Sicherheit in drahtlosen Netzwerken wird auch in der Zukunft noch ein Forschungsgebiet mit großem Bedarf an neuen Lösungen sein. Schon an den vorgestellten Anbindungskonzepten war der Widerspruch zwischen dem Sicherheitsbedürfnis und dem vollautomatischen Kommunikationsaufbau zu erkennen. Die selbstkonfigurierenden Netzwerke stellen in erster Linie ein Kommunikationsmedium dar, das sich darum bemüht, möglichst alle erreichbaren Geräte miteinander zu verbinden. Dies ist auf den ersten Blick wünschenswert. Allerdings wird dabei leicht vergessen, wie schnell ein Ad-hoc Netzwerk über die geplante Größe hinauswachsen kann. Ein gutes Beispiel dafür ist ein Mietshaus, in dem jeder Drucker und jeder Terminplaner von allen Bewohnern ansprechbar wird. Die Automatik versagt zwangsläufig an den Stellen, an denen das Sicherheitsbedürfnis in den Vordergrund tritt und nach privaten Netzwerken verlangt. Hier sind neue Konzepte nötig, die es jedermann erlauben, seine Privatsphäre festzulegen und zu bewahren.

Literaturverzeichnis

[ABSK95] E. Amir, H. Balakrishnan, S. Seshan, and R. Katz. Efficient TCP over networks with wireless links. In *Proceedings of 5th Workshop on Hot Topics in Operating Systems*, 1995.

[ACLGM93] E. Ayanoglu, I. Chih-Lin, R. Gitlin, and J. Mazo. Diversity coding for self-healing and fault tolerant communication networks. In *IEEE Trans. on Communication*, volume COM-41, pages 1677–1688, Nov. 1993.

[BB95] Ajay V. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. In *15th International Conference on Distributed Computing Systems (ICDCS)*, pages 136–143, May 1995.

[BB97] Ajay Bakre and B.R. Badrinath. Implementation and performance evaluation of indirect TCP. In *IEEE Trans. on Computers*, volume 46-3, March 1997.

[Ben96] Thorsten Benkner. Kapazitätssteigernde Maßnahmen für digitale Mobilfunksysteme der dritten Generation. Doktorarbeit an der Universität Siegen, Shaker Verlag, ISBN 3-8265-1682-6, 1996.

[Ber92] Dimitri Bertsekas. *Data Networks*. Prentice-Hall International, 1992.

[BJLF01a] Markus Borschbach, Ralph Jansen, Wolfram-M. Lippe, and Bernd Freisleben. Specific self-organized system architecture optimized and controlled via a load profile driven ad hoc network routing scheme. In *World Congress on Communication 2001*, pages 542–549. World Society of Engineering and Science (WSES)/IEEE, 2001.

[BJLF01b] Markus Borschbach, Ralph Jansen, Wolfram-M. Lippe, and Bernd Freisleben. Subjective hierarchical neighborhood load profile driven routing. In *World Congress on Communication 2001*, pages 96–103. World Society of Engineering and Science (WSES)/IEEE, 2001.

[BMJ+98a] Josh Broch, David Maltz, David Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of of the Fourth Annual ACM/IEEE Conference on Mobile Computing and Networking (Mobicom98)*, October 1998.

[BMJ+98b] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 85–97, 1998.

[BMJ99] J. Broch, D. Maltz, and D. Johnson. Supporting hierarchy and heterogeneous interfaces in multi-hop wireless ad hoc networks. In *Proceedings of the Workshop on Mobile Computing held in conjunction with the International Symposium on Parallel Architectures, Algorithms, and Networks*, Perth, Australia, 1999.

[Bor02] Markus Borschbach. Verfahren der reservierenden Wegewahl in selbstorganisierenden Funknetzen mit spezifischer Funksystemarchitektur. Doktorarbeit an der Universität Münster, 2002.

[CG98] Tsu-Wei Chen and Mario Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks. *Proceedings of IEEE ICC'98*, 1998.

[Col87] Charles Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[CQJM01] Thomas Clausen, Amir Qayyum, Philippe Jacquet, and Paul Muhlethaler. Optimized link state routing protocol. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-06.txt, September 2001. IETF Manet Working Group.

[Dem99] Sascha Demetrio. Ein Abrechnungssystem für Mobile IP. Semesterarbeit, September 1999. TU-Darmstadt, Fachbereich Informatik, Fachgebiet Betriebssysteme.

[Dep97] IEEE Standards Department. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard 802.11*. IEEE, 1997.

[DH98] S. Deering and R. Hinden. Internet protocol version 6 (IPv6) specification. Internet Request for Comments (RFC) 2460, December 1998.

[FJ97] Bernd Freisleben and Ralph Jansen. Analysis of routing protocols for ad hoc networks of mobile computers. In *Proceedings of the 15th IASTED International Conference on Applied Informatics*, pages 133–136. Iasted-Acta Press, 1997.

[FL00] D. Farinacci and T. Li. Generic routing encapsulation. Internet Request for Comments (RFC) 2784, March 2000.

[GHM01] Mario Gerla, Xiaoyan Hong, and Li Ma. Landmark routing protocol for large scale ad hoc networks. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-lanmar-02.txt, May 2001. IETF Manet Working Group.

[GHP00] M. Gerla, X. Hong, and G. Pei. Landmark routing for large ad hoc wireless networks. In *Proceedings of IEEE Gobecom 2000*, San Francisco, CA, November 2000.

[GJ79] Michael Garey and David Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[GLAS01] J.J. Garcia-Luna-Aceves and Marcelo Spohn. Bandwidth efficient link state routing in wireless networks. In Charles E. Perkins, editor, *Ad Hoc Networking*, pages 323–350. Addison-Wesley, 2001.

[GP00] Daniel Grossman and William Portnoy. Ad hoc routing for mobile packet networks: a literature review. CSE 561 – Computer Networks – Spring 2000, http://www.cs.washington.edu/homes/grossman/projects/ 561projects/adhoc/AdHoc.ps, 2000.

[GPH00] Mario Gerla, Guangyu Pei, and Xiaoyan Hong. Fisheye state routing protocol for ad hoc networks. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-fsr-02.txt, November 2000. IETF Manet Working Group.

[Gro96] IBM Network Working Group. IP mobility support. Internet Request for Comments (RFC) 2002, October 1996.

[Hel01] Gilbert Held. *Data over Wireless Networks*. McGraw Hill, 2001. ISBN 007-212621-3.

[HP98] Zygmunt Haas and Marc Pearlman. The Zone Routing Protocol for Highly Reconfigurable Ad-Hoc Networks. *Proceedings of ACM SIGCOMM 98*, August 1998.

[HPS01a] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar. The interzone routing protocol for ad hoc networks. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-ierp-01.txt, June 2001. IETF Manet Working Group.

[HPS01b] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar. The intrazone routing protocol for ad hoc networks. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-iarp-01.txt, June 2001. IETF Manet Working Group.

[Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, 1991.

[Jan98] Ralph Jansen. Simulating adaptive routing in highly dynamic wireless networks. In *Proceedings of the 1998 European Simulation Multiconference*, pages 863–869, Manchester, United Kingdom, June 1998. Society for Computer Simulation (SCS).

[JF98] Ralph Jansen and Bernd Freisleben. Adaptive connection discovery in mobile wireless networks. In *Proceedings of the 1998 International Conference on Interactive Applications of Mobile Computing (IMC98)*, pages 118–124, Rostock, Germany, November 1998.

[JF01] Ralph Jansen and Bernd Freisleben. Bandwidth efficient distance vector routing for ad hoc networks. In *Proceedings of the Wireless and Optical Communications Conference (WOC)*, pages 117–122, Banff, Canada, June 2001. IASTED, Iasted-Acta Press.

[JLT98] Mingliang Jiang, Jinyang Li, and Yong Chiang Tay. Cluster based routing protocol. Manet Internet Draft http://www.ietf.org/internet-drafts/draft-ietf-manet-cbrp-spec-00.txt, August 1998. IETF Manet Working Group.

[JMHJ01] David B. Johnson, David A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt, March 2001. IETF Manet Working Group.

[Joh94] David B. Johnson. Routing in ad hoc networks of mobile hosts. In *Workshop on Mobile Computing Systems and Applications*, pages 158–163, Santa Cruz, California, USA, December 1994.

[Kar90] P. Karn. MACA: A New Channel Access Protocol for Packet Radio. In *Proceedings of the 9th ARRL Computer Networking Conference*, pages 134–140, 1990.

[KPD85] P. Karn, H. Price, and R. Diersing. Packet radio in the amateur service. In *Journal on Selected Areas in Communications*, volume SAC-3, pages 431–439. IEEE, Mai 1985.

[Küh00] Jörg Kühn. Entwurf und Simulation von Mehrwege-Routing für kooperative Mobilfunknetze. Diplomarbeit an der Universität Siegen, September 2000. Fachbereich Elektrotechnik und Informatik, Fachgebiet Parallele Systeme.

[LKBP96] R. LaMaire, A. Krishna, P. Bhagwat, and J. Panian. Wireless LANs and mobile networking: standards and future directions. In *IEEE Personal Communications*, pages 86–94, August 1996.

[LP97] Hui Lei and Charlie Perkins. Ad hoc networking with mobile IP. In *Second European Personal Mobile Communication Conference*, Bonn, Germany, 1997.

[Meh94] Asha Mehrotra. *Cellular Radio: Analog and Digital Systems*. Artech House, 685 Canton Street, Norwood, MA 02062, 1994. ISBN 0-89006-731-7,.

[MGLA96] Shree Murthy and J.J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. In *Mobile Networks and Nomadic Applications (NOMAD), Special issue on Routing in Mobile Communication Networks*, volume 1/4, 1996.

[Pal97] G. Pall. Microsoft point-to-point compression (MPPC) protocol. Internet Request for Comments (RFC) 2118, March 1997.

[PB94] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing for mobile computers. In *Proceedings of the Symposium on Communication Architectures and Protocols*, pages 234–244. ACM SIGCOMM, 1994.

[PC97] Vincent Park and Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of INFOCOM '97, Kobe, Japan*, pages 1405–1413. IEEE, 1997.

[PC01] V. Park and S. Corson. Temporally-ordered routing algorithm (tora). Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-04.txt, July 2001. IETF Manet Working Group.

[PD00] Larry Peterson and Bruce Davie. *Computer Networks*. Morgan Kaufmann, second edition, 2000.

[Per97] Charles Perkins. Ad-hoc on-demand distance vector routing. In *MILCOM '97 panel on ad hoc networks*, Monterey, California, November 1997. IEEE.

[Per02] Charles Perkins. Ad hoc on demand distance vector (aodv) routing. Manet Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-10.txt, January 2002. IETF Manet Working Group.

[PJ97] Henning Pagnia and Ralph Jansen. Towards multiple-payment schemes for digital money. In Rafael Hirschfeld, editor, *Financial Cryptography: First International Conference, FC '97*, volume 1318, pages 203–215, Anguilla, British West Indies, 24–28 1997. Springer-Verlag.

[Ran96] D. Rand. The PPP compression control protocol. Internet Request for Comments (RFC) 1962, June 1996.

[RL95] Y. Rekhter and T .Li. A border gateway protocol 4 (BGP-4). Internet Request for Comments (RCF) 1771, March 1995.

[RT99] Elizabeth Royer and C-K Toh. A review of current routing protocols for ad-hoc mobile wireless networks. In *IEEE Personal Communications Magazine*, pages 46–55, Apr. 1999.

[SAMDZ92] A. U. Shankar, C. Alaettinoglu, I. Matta, and K. Dussa-Zieger. Performance comparison of routing protocols using maRS: Distance-vector versus link-state. In *Proc. 1992 ACM SIGMETRICS and PERFORMANCE '92 Int'l. Conf. on Measurement and Modeling of Computer Systems*, page 181, Newport, Rhode Island, USA, 1-5 1992.

[Sch94] Bruce Schneier. Applied cryptography. John Wiley & Sons, 1994.

[Sie97] Richard Sietmann. Drahtloses Festnetz nach dem Internet-Modell. *Funkschau*, Nr. 25, pages 109-112, 1997.

[SLH94] A. Santamaria and F.J. Lopez-Hernandez, editors. *Wireless LAN Systems*. Artech House, ISBN 0-89006-609-4 1994.

[SM95] S.Bradner and A. Mankin, editors. *IPng: Internet Protocol Next Generation*. Addison-Wesley Reading, MA, 1995.

[SR96] M. Steenstrup S. Ramanathan. A survey of routing techniques for mobile communications networks. In *Mobile Networks and Applications*, volume 1, pages 89–105, 1 1996.

[Ste95] Martha Steenstrup. *Routing in Communication Networks*. Prentice Hall, Englewood Cliffs, New Jersey, 1995. ISBN 0-13-010752-2.

[SWE98] Charlie Scott, Paul Wolfe, and Mike Erwin. *Virtual Private Networks, 2nd Edition*. O'Reilly, December 1998.

[Tan92] Andrew S. Tanenbaum. *Computer-Netzwerke*. Wolframs Verlag, 2nd edition, 1992. Deutsche Übersetzung.

[Tan96] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, Upper Saddle River, New Jersey 07458, 3rd edition, ISBN 0-13–349945-6, 1996.

[Toh97a] C.-K. Toh. Associativity based routing for ad hoc mobile networks. In *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, volume 4/2, pages 103–139, March 1997.

[Toh97b] C-K Toh. *Wireless ATM and AD-HOC Networks*. Kluwer Academic Publishers, 1997.

[Tsu88] P. F. Tsuchiya. The landmark hierarchy: A new hierarchy for routing in very large networks. *ACM Computer Communications Review*, 18(4):35–42, 1988.

[Tür00] Frank Türke. Entwicklung eines energie-effizienten Routingverfahrens für kooperative Mobilfunknetze. Diplomarbeit an der Universität Siegen, Dezember 2000. Fachbereich Elektrotechnik und Informatik, Fachgebiet Parallele Systeme.

[WSFW97] Jost Weinmiller, Morten Schlager, Andreas Festag, and Adam Wolisz. Performance study of access control in wireless LANs - IEEE 802.11 DFWMAC and ETSI RES 10 hiperlan. *Mobile Networks and Applications*, 2(1):55–67, 1997.

[WWW96] J. Weinmiller, H. Woesner, and A. Wolisz. Analyzing and improving the IEEE 802.11-MAC protocol for wireless LANs. In *Proceedings of the 4th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '96)*, pages 200–206, 1996.