Diplomarbeit

Stefan Uhlig
Matr.-Nr.: 2643805

Thema:
„Modellierung von Wireless LANs mittels Bedienungstheorie“

Betreuender Hochschullehrer: Prof. Dr. habil. A. Schill
Betreuer:
Dr. rer. nat. D. Gütter
Dr.-Ing. A. Luntovskyy

Eingereicht am: 17. August 2006
Inhaltsverzeichnis

1 Einführung .............................................................. 1
   1.1 Motivation .......................................................... 1
   1.2 Aufgabenstellung .................................................. 2
   1.3 CANDY ............................................................... 2
   1.4 Beispielszenario .................................................... 3
   1.5 Konventionen ....................................................... 4
   1.6 Aufbau dieser Arbeit .............................................. 5

2 Wireless LAN .......................................................... 7
   2.1 Formen von WLAN Netzen ...................................... 8
   2.2 Schichtenarchitektur .............................................. 8
   2.3 Die Bitübertragungsschicht .................................... 9
      2.3.1 Frequency Hopping Spread Spectrum ...................... 9
      2.3.2 Direct Sequence Spread Spectrum .......................... 10
      2.3.3 Orthogonal Frequency Division Multiplexing ............. 10
      2.3.4 802.11g PHY-Layer ........................................ 11
      2.3.5 Wichtige Parameter ........................................ 12
   2.4 Die MAC Teilschicht ............................................. 12
      2.4.1 Inter Frame Spaces - Prioritäten ......................... 13
      2.4.2 Physikalische Carrier-Sense-Funktion .................... 14
      2.4.3 Virtuelle Carrier-Sense-Funktion .......................... 14
      2.4.4 Empfangsbestätigungen .................................... 14
      2.4.5 Backoff-Algorithmus ....................................... 15
      2.4.6 RTS/CTS-Mechanismus ...................................... 15
      2.4.7 Beacon Frames .............................................. 16
      2.4.8 Verschlüsselung ............................................ 16
      2.4.9 Mehrere Datenraten (Multirate support) ................. 17

3 Bedienungstheorie ................................................ 19
   3.1 Einleitung ....................................................... 19
      3.1.1 Geschichte ................................................... 19
      3.1.2 Begriffe und Bezeichnungen ................................ 20
   3.2 Elementare Wartesysteme ....................................... 20
3.2.1 Die Kendall-Notation ................................. 21
3.2.2 Exponentialverteilung: Markov- und Poisson-Eigenschaft ........ 22
3.2.3 Leistungsgrößen ................................ 24
3.3 Bedienungsnetze ........................................ 24
3.3.1 Ein- und Mehrklassennetze ......................... 24
3.3.2 Offene und geschlossene Netze ..................... 25
3.3.3 Markovketten, Zustandsübergangsdiagramm .......... 25
3.3.4 Gleichgewicht .................................... 26
3.3.5 Globale Gleichgewichtsgleichungen ................. 27
3.3.6 Lokales Gleichgewicht ............................. 28
3.3.7 Jackson Theorem .................................. 28
3.3.8 Weitere Verfahren ................................ 29

4 Modellierung von WLAN-Netzen .......................... 31
4.1 Vergleich Mathematische Modellierung und Simulation ........ 32
4.2 Lastmodellierung ....................................... 32
4.3 Strukturelle Modellierung .............................. 35
4.3.1 Andere mögliche Konstellationen .................. 37
4.3.2 Typ der Bedienstation ............................... 38
4.3.3 Forderungen ...................................... 39
4.4 Parametrisierung des Modells ........................... 40
4.5 Zu berücksichtigende Aspekte ........................... 41
4.5.1 Unterschiedliche Brutto-Datenraten ................. 41
4.5.2 Verluste durch Wettbewerb (Contention) ........... 43
4.5.3 Protokoll-Overhead ................................ 46
4.5.4 Interaktionen TCP und WLAN ....................... 50
4.6 Bedienrate ............................................ 51
4.7 Auslastung ............................................. 52

5 Das Programm „Kapazitätsanalyse“ ....................... 53
5.1 Benutzung des Programmes ............................. 53
5.1.1 Systemvoraussetzungen ............................ 53
5.1.2 Programmaufruf .................................... 53
5.1.3 Funktionen der Programmoberfläche ................ 53
5.1.4 Karteikarte „Optionen“ ............................ 55
5.1.5 Karteikarte „Modellmanipulation“ .................. 57
5.1.6 Laden eines Projektes .............................. 58
5.1.7 Durchführen der Analyse ........................... 58
5.1.8 Ausgabe der Analyseergebnisse ..................... 59
5.2 Interne Funktionsweise ................................ 61
5.2.1 Repräsentation eines Bedienungsnetzes ............. 61
5.2.2 Aufbau des Bedienungsnetzes .................................................. 63
5.2.3 Parametrisierung der Knoten im Detail .................................... 65
5.2.4 Ermittlung der Protokoll-Overheads ....................................... 66
5.2.5 Durchführung der Analyse ....................................................... 68
5.2.6 Einschränkungen des Programmes .......................................... 68
5.3 Genutzte Technologien .............................................................. 69
5.3.1 Java Architecture for XML Binding (JAXB) .............................. 69
5.4 Integration in CANDY ................................................................. 71
5.4.1 Anforderungen an NDML ....................................................... 71
5.4.2 Umgehung der Probleme mit NDML 3.0 ................................. 73
5.4.3 Graphische Lastdarstellung .................................................... 74

6 Leistungsbetrachtungen zu WLAN in der Literatur 75
6.1 G. Bianchi:, „Performance Analysis of the IEEE 802.11 Distributed Coordination Function“ ............................................................. 75
6.2 Übersicht weiterer Papiere ............................................................ 79

7 Validierung mathematischer Modelle 85
7.1 NS-2 ......................................................................................... 85
7.1.1 Prinzip der Simulation mit NS-2 ............................................. 86
7.1.2 Unterstützung der WLAN-Modellierung ................................. 87
7.1.3 Funkausbreitungsmodelle ....................................................... 88
7.2 OMNeT++ ............................................................................... 89
7.2.1 Zusammenspiel der Bestandteile von OMNeT++ ....................... 89
7.2.2 Prinzip der Modellierung mit OMNeT++ ................................. 90
7.2.3 Modellierung von Rechnernetzen mit OMNeT++ ....................... 91
7.3 Vergleich .................................................................................. 91

8 Andere Modelle 93
8.1 „Analytische Verfahren für die Kapazitätsplanung von Rechnernetzen auf Basis der Bedienungstheorie“ ......................................................... 93
8.2 LAN-Lastmodell von Wang Xuewei .......................................... 93
8.2.1 Wertung ............................................................................... 95
8.3 Der StandortFinder-Algorithmus ................................................. 95
8.3.1 Verwendetes Ausbreitungsmodell ......................................... 95
8.3.2 Algorithmus .......................................................................... 96
8.3.3 Programmtechnische Realisierung ......................................... 96
8.3.4 Wertung ............................................................................... 98
8.4 Wireless Planer ........................................................................... 98
8.4.1 Lastaspekte ........................................................................... 98
8.4.2 Verschiedene Ausbreitungsmodelle ....................................... 99
8.4.3 Algorithmus ................................................. 99
8.4.4 Wertung .................................................. 99
8.5 Multifarben-Tintenfleck-Algorithmus .......................... 99
  8.5.1 Formel .................................................. 101
  8.5.2 Analysemodi .......................................... 101
  8.5.3 Wertung ............................................... 101
8.6 Fazit ...................................................... 102

9 Zusammenfassung und Ausblick ................................. 103

Glossar ......................................................... 105

Tabellenverzeichnis ............................................. 107

Literaturverzeichnis ............................................ 111

Eigenständigkeitserklärung und Danksagung .................... 114

Beiliegende Software .......................................... 115
Kapitel 1
Einführung

1.1 Motivation

WLANs werden heute routinemäßig eingesetzt. Die aktuellen WLAN-Standards 802.11a/h und 802.11g ermöglichen beide Datenraten bis zu 54 MBit/s. Proprietäre Lösungen erreichen das Doppelte und im Standard 802.11n, der die nächste WLAN-Generation beschreibt, sind Datenraten bis 320 MBit/s vorgesehen. Diese - schon erfolgte und zukünftige - Leistungssteigerung bewirkt, daß die WLAN-Technik für viele Anwendungen als den drahtgebundenen Technologien gleichwertiges Mittel bei der Planung von Rechnernetzen berücksichtigt werden kann, mit dem Bonus an Flexibilität, der sich durch die fehlende Notwendigkeit der Verkabelung ergibt.

Um einen optimalen Einsatz von WLANs zu gewährleisten, sind einige Besonderheiten gegenüber drahtgebundenen Technologien zu beachten. Diese betreffen unter anderem die Kanalwahl, die Datensicherheit und die höheren Overheads, die durch das vergleichsweise empfindliche Funkmedium erforderlich sind.

Wie die Rechnernetzprojektierung allgemein, erfolgt auch die Planung von WLAN-Infrastrukturnetzen auch heute noch meist intuitiv, basierend auf den Kenntnissen und Erfahrungen des Planers. Die Unterstützung durch Software-Werkzeuge beschränkt sich meist auf einzelne Teilaufgaben.

Das CANDY\(^1\)-Projekt hat das Ziel, ein integriertes Werkzeug zu entwickeln, welches den gesamten Rechnernetzplanungsprozeß und seine verschiedenen Teilaufgaben unterstützt. Teil von CANDY soll auch ein Werkzeug zur Kapazitätsanalyse von Rechnernetzen auf Basis der Bedienungstheorie werden, so daß Kapazitätsanalysen bereits in der Planungsphase eines Rechnernetzes durchgeführt werden können.


\(^1\)Computer Aided Network Design Utility
1.2 Aufgabenstellung


Im Rahmen der Arbeit ist die Modellierung von Wireless LANs mit Hilfe der Bedienungstheorie zu untersuchen. Es sind ein Vorschlag über die Realisierung innerhalb der Netzwerkplanungsumgebung CANDY und eine prototypische Lösung zu erarbeiten.

Außerdem ist das gefundene Modell mit vorhandenen Modellen für WLANs zu vergleichen.

1.3 CANDY

CANDY ist ein integriertes Rechnernetzplanungswerkzeug, welches am Lehrstuhl Rechnernetze an der Informatik-Fakultät der TU Dresden entwickelt wird. Ziel ist, den gesamten Rechnernetzplanungsprozeß zu begleiten und Routineaufgaben zu automatisieren. CANDY ist modular aufgebaut. Jedes Modul unterstützt einen bestimmten Teilaspekt der Modellierung, z.B.:

- grafische Topologie-Eingabe
- Topologie-Validierung
- Kabel-Trassierung
- Kostenanalyse
- Access Point-Platzierung
- Last-Simulation
- Kapazitätsanalyse

Die XML-basierte Fachsprache NDML\(^2\) integriert die einzelnen Module miteinander. Sie dient dem Datenaustausch sowie der Datenhaltung. Der Planungsprozeß insgesamt wird durch ein Modul zum Projektmanagement überwacht.


NDML 3.0 definiert drei Viewpoints. BASIC dient der allgemeinen Beschreibung eines Projektes. LOAD beschreibt Lastanforderungen und TOPOLOGY die Topologie des betrachteten Rechnernetzes. Weitere Viewpoints sollen als Erweiterungen realisiert werden.

Abbildung 1.1 gibt einen Überblick über die CANDY-Architektur.

\(^2\)Network Design Markup Language
1.4 Beispielszenario

Ein Beispielszenario soll den Einsatz eines Werkzeugs zur Kapazitätsanalyse und damit in Beziehung stehender CANDY-Module verdeutlichen.

Abbildung 1.2 zeigt einen Ausschnitt eines Gebäudegrundrisses mit einer schematischen Darstellung der installierten/geplanten Netzwerkinfrastruktur. Die verwendete Symbolik wird in Abschnitt 1.5 erklärt. Teil dieses Netzes sind auch WLAN-Clients, die über einen Access Point angebunden sind. In der Abbildung sind die Datenströme einiger Anwendungen eingetragen, de-
ren Einsatz geplant ist. Aufgabe der Kapazitätsanalyse ist es nun zu ermitteln, ob die vorgesehene Netzwerkinfrastriktur den Lastanforderungen gewachsen ist.


Sollte die nun durchgeführte Kapazitätsanalyse eine Überlastung einzelner Netzkomponenten aufzeigen, kann das Szenario u.a. im grafischen Topologieeditor modifiziert werden.

Wird etwa eine Überlastung des Access Points deutlich, kann versucht werden, diesen so umzusetzen, daß der mit 24 MBit/s angebundene Client eine höhere Datenrate nutzen kann.

Anschließend kann die veränderte Konstellation durch eine erneute Kapazitätsanalyse überprüft werden.

### 1.5 Konventionen

Die folgenden Notationen und Formatierungen werden im gesamten Text verwendet.

Neu eingeführte Begriffe erscheinen in *Kursivschrift*.


Quelltextfragmente werden abgesetzt auf grauem Hintergrund dargestellt.

Zur Darstellung der Schnittstellen von Java-Klassen wird eine an die Deklaration abstrakter Methoden angelehnte Notation verwendet. Es werden dabei lediglich die Signaturen der Methoden notiert, gefolgt von einem Semikolon, ohne den Methodenrumpf.

Klassen-, Methoden-, Variablennamen etc. im laufenden Text werden in *Schreibmaschinen­schrift* dargestellt.

Bei der Darstellung von Netzwerk-Topologien findet die folgende Symbolik Verwendung:

- **AP** Access Point
- **SW** Switch
- **C** Client-PC/Laptop
- **S** Server
- **- - - -** WLAN-Verbindung
- **- - -** Datenstrom (farbig)
1.6 Aufbau dieser Arbeit

In Kapitel 2 wird ein kurzer Überblick über die für diese Arbeit relevanten Aspekte der WLAN-Standards 802.11, 802.11b, 802.11a/h und 802.11g gegeben.

Anschließend stellt Kapitel 3 Grundlagen der Bedienungstheorie dar.

In Kapitel 4 wird ein Modell zur Kapazitätsanalyse für WLAN entwickelt. Dabei wird in diesem wie auch dem folgenden Kapitel das übergeordnete Ziel, ein Kapazitätsanalyse-Tool für allgemeine Rechnernetze, berücksichtigt.

In Kapitel 5 wird das Programm „Kapazitätsanalyse“ beschrieben, welches des Modell des vorangegangenen Kapitels umsetzt. Neben der Bedienung des Programms wird auf technische Details eingegangen. Außerdem ist die Integration mit CANDY Gegenstand dieses Kapitels.


Kapitel 7 geht auf Möglichkeiten ein, die Ergebnisse eines Lastmodells zu vergleichen bzw. zu überprüfen. Speziell werden die Simulationsprogramme NS/2 und OMNeT++ vorgestellt und auf ihre Eignung zur Simulation von WLAN-Funknetzen untersucht.

Anschließend geht Kapitel 8 kurz auf andere Modelle ein, die mit dem Gegenstand dieser Arbeit in Beziehung stehen. Es handelt sich zum einen um Funkausbreitungsmodelle, die die Eingabedaten für das Modell dieser Arbeit liefern. Zum anderen sind alternative Modelle zur Last- und Kapazitätsanalyse Gegenstand der Betrachtung.

Zum Abschluß faßt Kapitel 9 die Arbeit zusammen und gibt einen Ausblick.
Kapitel 2

Wireless LAN

Ziel dieses Kapitels ist es, einen Überblick über die relevanten WLAN-Standards 802.11, 802.11b, 802.11a/h und 802.11g zu geben. Die Darstellung beschränkt sich auf die Leistungsmodellierung relevante Aspekte der MAC-Schicht und der einzelnen PHY-Realisierungen.

Die Informationen dieses Kapitels basieren größtenteils auf Darstellungen in [4] bzw. den WLAN-Standards selbst ([7]).


1999 wurde die Standarderweiterung 802.11b verabschiedet. In ihr ist eine PHY-Realisierung definiert, die im 2,4GHz-Frequenzband nach dem DSSS-Verfahren arbeitet und die zusätzlichen Datenraten 5,5 und 11 MBit/s bietet.


2003 wurde die heute besonders in Europa weit verbreitete Standarderweiterung 802.11g verabschiedet. Darin wird ein PHY-Layer definiert, der ebenfalls die Datenraten von 6, 9, 12, 18, 24, 36, 48 und 54 MBit/s ermöglicht. Im Unterschied zu 802.11a wird hier das 2,4GHz-Band genutzt. Es kommt ebenfalls das OFDM-Verfahren zum Einsatz. Außerdem unterstützt 802.11g das DSSS/CCK-Verfahren mit den Datenraten 1, 2, 5,5 und 11 MBit/s und ist daher kompatibel.
IEEE 802.11e MAC-Erweiterung für Quality of Service und Performance-Steigerung (EDCF - Enhanced DCF).
IEEE 802.11f Definition des Inter Access Point Protocols (IAPP).
IEEE 802.11i MAC-Erweiterung zur Verbesserung der Datensicherheit und Authentifizierung.

Tabelle 2.1: Weitere 802.11-Standarderweiterungen

zu 802.11 und 802.11b. Vorhandene WLAN-Technik nach diesen Standards kann zusammen mit 802.11g-Technik genutzt werden.
Einige weitere Standarderweiterungen sind in Arbeit oder bereits verabschiedet. Sie werden in Tabelle 2.1 kurz beschrieben.

2.1 Formen von WLAN Netzen


Im Infrastrukturennetz erfolgt die Kommunikation ausschließlich über einen Access Point. Dieser besitzt neben einer WLAN-Schnittstelle in der Regel eine Ethernet-Schnittstelle und ermöglicht den WLAN-Clients den Zugang zum drahtgebundenen LAN bzw. ins Internet.


Ist im Folgenden von einem BSS die Rede, sind damit WLAN-Stationen gemeint, die im Infrastrukturnmodus einen Funkkanal gemeinsam nutzen: meistens ein Access Point und die angemeldeten Clients oder auch mehrere Access Points (Wireless Bridge).

2.2 Schichtenarchitektur

Die in den 802.11x Standards\(^5\) definierte Funktionalität entspricht den beiden unteren Schichten des ISO/OSI-Modells. Die MAC-Schicht, der untere Teil der Sicherungsschicht, hat die Aufgabe,
2.3 Die Bitübertragungsschicht

Eine fehlerfreie Datenübertragung zu gewährleisten.


Die auf den Schichten hantierten Dateneinheiten lassen sich in zwei Gruppen einteilen: Dateneinheiten, die von der darüber liegenden Schicht übergeben wurden - diese werden als Service Data Unit (SDU) bezeichnet - und Dateneinheiten, die die betrachtete Schicht durch Hinzufügen eigener Header erzeugt - sie werden als Protocol Data Unit bezeichnet. MAC Frames sind damit aus Sicht der MAC Schicht MAC Protocol Data Units (MPDU) und aus Sicht der PLCP Teilschicht PLCP Service Data Units (PSDU). Analog ist ein an die MAC Schicht heruntergereichtes Datensegment eine MSDU und das vom PHY-Layer oder genauer der PLCP Teilschicht erzeugte Frameformat eine PPDU.

2.3.1 Frequency Hopping Spread Spectrum


Die folgende Darstellung soll sich auf diese relevanten Aspekte beschränken, dementsprechend wird bei den einzelnen Frameformaten nur auf Größe und Übertragungsdauer von Header, Präambel und Datenteil eingegangen. Für eine detailliertere Darstellung der Funktionsweise der einzelnen PHY-Realisierungen sei auf die reichlich vorhandene Literatur zu WLANs und die Standards selbst ([7]) verwiesen.

Das von dem im 802.11-Grundstandard eingesetzten FHSS-Verfahren genutzte Frameformat besteht aus folgenden Elementen:

- PLCP Präambel: 96 Bits
- PLCP Header: 32 Bits
- PSDU

Präambel und Header werden mit 1 MBit/s übertragen, die PSDU mit einer der unterstützten Datenraten, 1 bzw. 2 MBit/s.
2.3.2 Direct Sequence Spread Spectrum

Das DSSS-Verfahren wird im Grundstandard sowie in 802.11b und 802.11g eingesetzt. Folgendes Frameformat ist in 802.11 spezifiziert:

- PLCP Präambel: 144 Bits
- PLCP Header: 48 Bits
- PSDU

Header und Präambel werden wieder mit 1 MBit/s übertragen, die PSDU mit einer der vom Standard unterstützten Datenraten. Die Übertragung von Präambel und Header zusammen dauert damit 192 $\mu$s. Der Grundstandard definiert für DSSS ebenfalls die Datenraten 1 MBit/s und 2 MBit/s. 802.11b unterstützt zusätzlich 5,5 und 11 MBit/s.

Die 802.11b Erweiterung definiert ein optionales kurzes Frameformat (Short Preamble):

- Kurze PLCP Präambel: 72 Bits
- Kurzer PLCP Header: 48 Bits
- PSDU

Die kurze Präambel wird nach wie vor mit 1 MBit/s gesendet, der Header jedoch mit 2 MBit/s. Die Übertragung von Header und Präambel dauert damit nur noch 96 $\mu$s. Die PSDU wird mit 2, 5,5 oder 11 MBit/s übertragen.

Der Access Point zeigt über das sog. Beacon-Frame (siehe Abschnitt 2.4.7) an, ob er das Short-Frame-Format unterstützt. Dieses kann dann genutzt werden, muß jedoch nicht, da es sich um ein optionales Feature handelt.

2.3.3 Orthogonal Frequency Division Multiplexing

Das OFDM Verfahren wird in 802.11a/h und 802.11g eingesetzt. Die Informationen hier gelten für 802.11a, wenn auch das Verfahren bei 802.11g grundsätzlich das gleiche ist. Auf 802.11g geht der nächste Abschnitt ein.

Bei OFDM werden verschiedene Modulationsverfahren sowie Fehlerkorrekturcodes\(^6\) verschiedener Raten eingesetzt, woraus die unterschiedlichen erzielbaren Datenraten resultieren. Tabelle 2.2 gibt eine Übersicht der relevanten Parameter. Die Datenrate berechnet sich nach der Formel

\[
\text{Datenrate} = \text{Symbolrate} \cdot N_{\text{BPSC}} \cdot \text{Unterkanäle} \cdot R
\]


Das eingesetzte Frame-Format sieht folgendermaßen aus:

\(^6\)FEC - Forward Error Correction
2.3 Die Bitübertragungsschicht

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Bits/Unterkanal ($N_{BPSC}$)</th>
<th>Bits/OFDM-Symbol ($N_{CBPS}$)</th>
<th>FEC Rate (R)</th>
<th>Datenbits/OFDM-Symbol ($N_{DBPS}$)</th>
<th>Datenrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>48</td>
<td>1</td>
<td>1/2</td>
<td>24</td>
<td>6 MBit/s</td>
</tr>
<tr>
<td>BPSK</td>
<td>48</td>
<td>1</td>
<td>3/4</td>
<td>36</td>
<td>9 MBit/s</td>
</tr>
<tr>
<td>QPSK</td>
<td>96</td>
<td>2</td>
<td>1/2</td>
<td>48</td>
<td>12 MBit/s</td>
</tr>
<tr>
<td>QPSK</td>
<td>96</td>
<td>2</td>
<td>3/4</td>
<td>72</td>
<td>18 MBit/s</td>
</tr>
<tr>
<td>16-QAM</td>
<td>192</td>
<td>4</td>
<td>1/2</td>
<td>96</td>
<td>24 MBit/s</td>
</tr>
<tr>
<td>16-QAM</td>
<td>192</td>
<td>4</td>
<td>3/4</td>
<td>144</td>
<td>36 MBit/s</td>
</tr>
<tr>
<td>64-QAM</td>
<td>288</td>
<td>6</td>
<td>2/3</td>
<td>192</td>
<td>48 MBit/s</td>
</tr>
<tr>
<td>64-QAM</td>
<td>288</td>
<td>6</td>
<td>3/4</td>
<td>216</td>
<td>54 MBit/s</td>
</tr>
</tbody>
</table>

Tabelle 2.2: OFDM-Datenraten

- PLCP Präambel: 12 Symbole
- SIGNAL: 1 Symbol
- Daten

Präambel und Signal-Feld werden mit BPSK, FEC-Rate=1/2 übertragen, der Datenteil mit einer der unterstützten Datenraten. Der Datenteil besteht aus dem 16 Bit großen Service-Feld, der PSDU, 6 Tail Bits sowie einer variablen Anzahl Pad-Bits. Diese resultieren daraus, daß nur ganze Symbole übertragen werden können und daher die Länge des Datenteils ohne Rest durch die Anzahl der Datenbits pro OFDM-Symbol teilbar sein muß. Ist dies nicht der Fall, wird mit der entsprechenden Anzahl Null-Bits aufgefüllt. Die Anzahl $N_{PAD}$ der Pad-Bits kann mit folgenden Formeln ermittelt werden:

\[
N_{BITS} = 16 + 8 \cdot \text{Length}(PSDU) + 6
\]
\[
N_{SYM} = \left\lfloor N_{BITS}/N_{DBPS} \right\rfloor
\]
\[
N_{DATA} = N_{SYM} \cdot N_{BITS}
\]
\[
N_{PAD} = N_{DATA} - N_{BITS}
\]

2.3.4 802.11g PHY-Layer

Der 802.11g Standard definiert zwei PHY-Typen: den Extended Rate PHY (ERP) und den non Extended Rate PHY (nonERP). Der ERP unterstützt die in 802.11g definierten höheren Datenraten. Der nonERP unterstützt die Datenraten von 1 bis 11 MBit/s nach 802.11b. 802.11g definiert die Datenraten 6, 9, 12, 18 und 24 MBit/s als obligatorisch und 36, 48 und 54 MBit/s als optional. Diese Datenraten werden wie bei 802.11a mit dem OFDM-Verfahren realisiert.

Desweiteren ist eine optionale PHY-Realisierung vorgesehen, die das PBCC\(^7\) Verfahren nutzt und

\(^7\text{Packet Binary Convolutional Code}\)
Datenraten von 22 und 33 MBit/s anbietet.

Für den ERP sind verschiedene Betriebsmodi definiert: Sind 802.11g-Geräte unter sich, kann das OFDM Verfahren in Reinform eingesetzt werden. Das entspricht dann vom Prinzip her 802.11a. Um mit 802.11b-Geräten zu kommunizieren, kann das DSSS-Verfahren genutzt werden. So kann z.B. ein 802.11g Client in ein 802.11b WLAN eingebunden werden. Für den umgekehrten Fall, also die Integration von 802.11b Clients in 802.11g Netze, sind mehrere Verfahren definiert, die eine reibungslose Zusammenarbeit gewährleisten.

Das erste Verfahren ist der *Protection Mechanismus*. Dabei werden Beacon Frames grundsätzlich mit dem DSSS-Verfahren übertragen. Sobald sich ein 802.11b Gerät beim Access Point anmeldet, werden alle Übertragungen durch das RTS/CTS-Verfahren (siehe Abschnitt 2.4.6) eingeleitet, wobei die RTS- und CTS-Frames ebenfalls per DSSS übertragen werden. Die eigentliche Datenübertragung erfolgt per OFDM oder PBCC. Um den entstehenden Overhead zu begrenzen existiert eine Variante des Protection Mechanismus, bei der eine Station per DSSS ein CTS an sich selbst schickt (*CTS-to-self*), um eine Datenübertragung einzuleiten. Das RTS wird eingespart.


Der Betrieb von 802.11b Geräten in einem 802.11g WLAN wird als *Mixed Mode* bezeichnet.

### 2.3.5 Wichtige Parameter

Tabelle 2.3 fasst einige wichtige Parameter der einzelnen PHY Realisierungen zusammen. Deren Bedeutung wird sich - soweit nicht schon geklärt - im nächsten Abschnitt zur MAC-Teilschicht erschließen.

### 2.4 Die MAC Teilschicht

Dieser Abschnitt widmet sich der durch die MAC-Teilschicht realisierten Funktionalität. Im 802.11 Standard sind zwei Betriebsmodi vorgesehen:

- Distributed Coordination Function (DCF)
- Point Coordination Function (PCF)

Bei Einsatz der DCF erfolgt die Steuerung des Zugriffs auf den Funkkanal dezentral. Die Stationen eines BSS konkurrieren um den Kanalzugriff. Dabei wird allen Stationen langfristig die
2.4 Die MAC Teilschicht

<table>
<thead>
<tr>
<th>Parameter</th>
<th>802.11</th>
<th>802.11/802.11b</th>
<th>802.11a</th>
<th>802.11g</th>
</tr>
</thead>
<tbody>
<tr>
<td>aSlotTime FHSS</td>
<td>50μs</td>
<td>20μs</td>
<td>9μs</td>
<td>9μs, 20μs (M)</td>
</tr>
<tr>
<td>sSIFS FHSS</td>
<td>28μs</td>
<td>10μs</td>
<td>16μs</td>
<td>10μs</td>
</tr>
<tr>
<td>aMPDUMaxLength</td>
<td></td>
<td>4095 Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>aCWmin</td>
<td>15</td>
<td>31</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>aCWmax</td>
<td></td>
<td>1023</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dauer der PLCP Präambel</td>
<td>96μs</td>
<td>144μs, 72μs (SF)</td>
<td>16μs</td>
<td>20μs, 72μs (M)</td>
</tr>
<tr>
<td>Dauer des PLCP Headers</td>
<td>32μs</td>
<td>48μs, 24μs (SF)</td>
<td>4μs</td>
<td>4μs, 24μs (M)</td>
</tr>
</tbody>
</table>

SF = Short Frame, M = Mixed Mode

Tabelle 2.3: Parameter der PHY Realisierungen

gleiche Kanalzugriffschance garantiert. Das genutzte Protokoll ist CSMA/CA\(^8\) und ist dem bei Ethernet genutzten CSMA/CD verwandt. Grundprinzip ist, daß eine Station mit Sendewunsch den Kanal abhört und erst sendet, wenn sie ihn als frei erkennt. Es werden Maßnahmen zur Kollisionsvermeidung getroffen.


Es ist auch ein gemischter Modus vorgesehen, in dem abwechselnd mit DCF und PCF gearbeitet wird. Die Phasen, in denen die DCF genutzt wird, heißen Contention Period (CP). Entsprechend sind die Phasen der PCF-Nutzung Contention Free Periods (CFP)

Im folgenden soll nur auf die Distributed Coordination Function näher eingegangen werden, da die PCF bis heute in keinem am Markt verfügbaren WLAN-Produkt umgesetzt wurde.

### 2.4.1 Inter Frame Spaces - Prioritäten

Zur Realisierung unterschiedlicher Prioritäten von Übertragungen wurden vier verschiedene Mindestabstände zwischen aufeinanderfolgenden Übertragungen festgelegt:

**SIFS** Short Interframe Space
Höchste Priorität, Mindestabstand vor ACK-Frames, CTS-Frames und Folgeframes fragmentierter Nachrichten

**PIFS** Point (Coordination Function) Interframe Space
Wird von Stationen im PCF-Modus verwendet; stellt sicher, daß diese vor anderen Stationen bevorzugt werden

\(^8\)Carrier Sense Multiple Access/Collision Avoidance
**DIFS** Distributed Interframe Space

**EIFS** Extended Interframe Space
Bricht eine Station ihre Übertragung ab, dürfen andere Stationen nach Ablauf einer EIFS auf das Medium zugreifen, ohne den Ablauf des NAV-Vektors (siehe Abschnitt 2.4.3) abzuwarten.

Die Dauer der Interframe Spaces hängt vom genutzten PHY-Typ ab. Dabei ist jeweils die Dauer eines SIFS und die Slot Time festgelegt, die anderen IFS ergeben sich nach folgenden Formeln:

- \( \text{PIFS} = aSIFSTime + sSlotTime \)
- \( \text{DIFS} = aSIFSTime + 2 \times aSlotTime \)
- \( \text{EIFS} = aSIFSTime + \text{DIFS} + (8 \times \text{ACK-Länge}) + \text{Präambel-Dauer} + \text{PLCP-Header-Dauer} \)

Für die ACK-Länge wird eine Datenrate von 1 MBit/s zugrunde gelegt. Die Längen der PLCP-Präambel und des PLCP-Headers sind vom verwendeten PHY abhängig.

### 2.4.2 Physikalische Carrier-Sense-Funktion
Eine Station mit Sendewunsch führt zunächst ein physikalisches Carrier-Sensing durch. Wird der Kanal für die Dauer einer DIFS als frei erkannt, sendet die Station. Wird das Medium als belegt erkannt, stellt die Station ihren Sendewunsch zurück. Sie überwacht den Kanal weiter; wird er nach Ende der laufenden Übertragung für eine DIFS als frei erkannt, wird das Backoff-Verfahren (siehe Abschnitt 2.4.5) eingeleitet, um das Risiko einer Kollision mit anderen Übertragungen zu verringen.

### 2.4.3 Virtuelle Carrier-Sense-Funktion

Jedes übertragene Frame enthält im Duration/ID-Feld die Information, wie lange die vollständige Übertragung der Daten dauern wird. Alle Stationen, die das Frame empfangen, können mit dieser Information ihren NAV-Wert aktualisieren. Dabei gibt das Duration/ID-Feld eines Datenframes die Dauer an, bis die Übertragung des zugehörigen Acknowledgement-Frames abgeschlossen ist.

### 2.4.4 Empfangsbestätigungen
Kollisionen lassen sich trotz aller Vermeidungsmaßnahmen nicht ausschließen. Daher bestätigt die empfangende Station den Erhalt eines Frames durch Senden eines Acknowledgement-Frames (kurz: ACK). Bleibt dieses ACK aus, betrachtet der Sender die Übertragung als gescheitert und
sendet die Daten - nach Ablauf einer Backoff-Phase - erneut. Broadcast- und Multicast-Frames werden nicht bestätigt.

Die Aussendung eines ACK erfolgt nach Ablauf eines SIFS. Dadurch wird eine höhere Priorität der ACK-Frames gegenüber normalen Datenframes gewährleistet.

### 2.4.5 Backoff-Algorithmus


Der Wert für den Backoff-Zähler wird zufällig aus der Menge \((0 \ldots CW - 1)\) gewählt. \(CW\) ist das sogenannte Contention Window und ergibt sich nach der Vorschrift \(CW = \min(2^m \cdot CW_{\text{min}}, CW_{\text{max}})\). Die Werte \(CW_{\text{min}}\) und \(CW_{\text{max}}\) sind durch den genutzten PHY-Layer vorgegeben. \(m\) ist die Backoff-Ebene\(^10\). Der Anfangswert für \(m\) ist 0. Nach Auftreten von Kollisionen wird \(m\) jeweils um 1 erhöht. Obere Schranke für \(CW\) ist jedoch \(CW_{\text{max}}\). Konnte ein Frame erfolgreich verschickt werden, wird \(m\) auf 0 zurückgesetzt. Dabei ist auch der Fall eingeschlossen, daß ein RTS-Frame durch ein CTS bestätigt wurde.

Bei Aussendung mehrerer Frames hintereinander in Folge ist auch ein Backoff durchzuführen. Dies dient der Vermeidung des sog. *Channel-Capture-Problemes*. Dieses tritt auf, wenn eine Station ein Frame nach dem anderen sendet und die übrigen Stationen keinen Zugriff auf den Kanal erhalten.

Wird der Kanal bei Auftreten eines Sendewunsches sofort als frei erkannt, kann direkt nach Ablauf einer DIFS, also ohne Backoff, gesendet werden.

### 2.4.6 RTS/CTS-Mechanismus


---

\(^9\) engl.: time slots

\(^10\) engl.: backoff stage
Um die Performance-Einbußen durch derartige Kollisionen und die damit verbundenen Backoff-Phasen und Retransmissionen zu verringern, sieht der Standard den RTS/CTS-Mechanismus vor. Dabei geht jeder Datenübertragung der Austausch zweier kurzer Kontrollframes voraus. Die sendewillige Station sendet ein RTS-Frame (Request-To-Send). Dieses wird von der Empfängerstation durch ein CTS (Clear-To-Send) quittiert. In beiden Frames ist die Dauer der geplanten Datenübertragung inklusive ACK vermerkt. So können auch Stationen, die nur das RTS oder das CTS empfangen konnten, ihre NAV-Werte anpassen und eigene Übertragungen entsprechend zurückstellen.


### 2.4.7 Beacon Frames


Aufgabe der Beacon-Frames ist es zum einen, potentiellen Clients die Dienste des Access Points anzubieten und Ihnen wichtige Parameter des BSS mitzuteilen. Zum anderen werden die Stationen des BSS über die Beacon Frames synchronisiert.

### 2.4.8 Verschlüsselung

Da Vertraulichkeit der übertragenen Daten in einem drahtlosen Netz wesentlich problematischer ist als in einem drahtgebundenen Netz, sah bereits der WLAN-Grundstandard ein Verfahren zur
2.4 Die MAC Teilschicht

Datenverschlüsselung vor - den sog. WEP\textsuperscript{11}-Algorithmus. Dieser erweitert den Datenteil eines MAC-Frames um insgesamt 8 Byte:

- 4 Byte IV-Feld (3 Byte Initialisierungsvektor + 6 Pad Bits + 2 Key ID Bits)
- 4 Byte ICV (Integrity Check Value)

Da der WEP-Algorithmus (WEP40) den an ihn gestellten Anforderungen nicht gerecht wurde, sind einige Erweiterungen/Alternativlösungen entstanden. WEP128 ist nicht im WLAN-Standard spezifiziert. Es handelt sich um eine durch die Hersteller von WLAN-Hardware definierte Erweiterung, bei der die Schlüssellänge von 40 auf 108 Bit verlängert wurde.


In diese Kategorie fällt das TKIP-Verfahren. Dabei wird unter anderem die Länge des Initialisierungsvektors auf 6 Byte verlängert. Insgesamt vergrößert sich der Datenteil bei Nutzung von TKIP also um 11 Byte.

Desweiteren enthält 802.11i einen nach heutigem Stand sicheren Verschlüsselungsalgorithmus (AES: Advanced Encryption Standard) sowie die RADIUS-Authentifizierung als Ersatz für das Shared-Key-Verfahren.

2.4.9 Mehrere Datenraten (Multirate support)

Die verschiedenen 802.11 PHYs unterstützen mehrere Datenraten und erlauben einen dynamischen Wechsel der Datenrate, um sich an unterschiedliche Übertragungsbedingungen anzupassen. Zur Gewährleistung der Interoperabilität von Stationen, die verschiedene Datenraten nutzen, ist die Einhaltung gewisser Regeln erforderlich. Diese wurden schon im Grundstandard festgelegt, in den Nachfolgestandards aber zum Teil abgeändert. Soweit nicht anders gekennzeichnet, beziehen sich die folgenden Aussagen auf die Standards ab 802.11b.

In jedem WLAN sind drei Mengen von Datenraten definiert. Die Menge der \textit{obligatorischen Datenraten}\textsuperscript{12} ist standardspezifisch und gibt die Datenraten an, die WLAN-Hardware unterstützen muß, um sich dem jeweiligen Standard konform nennen zu dürfen.

Die Menge der \textit{Basisdatenraten}\textsuperscript{13} ist ein Konfigurationsparameter des Access Points. Sie gibt die Datenraten an, mit denen eine Station senden und empfangen können muß, damit sie sich mit dem WLAN assozieren kann.

Die Menge der \textit{Betriebsdatenraten}\textsuperscript{14} gibt die Datenraten an, die im BSS genutzt werden dürfen. Es findet keine Übertragung mit einer schnelleren Datenrate statt.

\textsuperscript{11} Wired Equivalent Privacy\textsuperscript{11}
\textsuperscript{12} engl.: mandatory rate set
\textsuperscript{13} engl.: basic rate set
\textsuperscript{14} engl.: operational rate set

Daten- oder Managementframes mit Unicast-Zieladresse können mit einer beliebigen vom Standard und vom Empfänger unterstützten Datenrate gesendet werden, diese wird durch den *Rate Switching Algorithmus* bestimmt.

ACK- und CTS-Frames (sog. *Control response frames*) sind mit der höchsten Rate der Basissatzmenge zu übertragen, die kleiner oder gleich der Rate ist, mit der die vorangegangene Übertragung stattfand.

Besondere Regeln gelten, wenn der Protectionmechanismus von 802.11g genutzt wird. Frames, die dem Protection Mechanismus dienen, werden mit einer der obligatorischen Datenraten des DSSS-Verfahrens übertragen oder, falls die Basisdatensatzmenge des Netzes Datenraten nach 802.11 bzw. 802.11b enthält, mit einer von diesen.
Kapitel 3
Bedienungstheorie

3.1 Einleitung


Wichtige Parameter eines solchen Modells sind die Raten, mit denen Aufträge an den Bedienstationen ankommen und die Leistungsfähigkeit sowie andere Parameter der einzelnen Arbeitsstationen - wie etwa maximale Warteschlangenlänge und Warteschlangendisziplin. Zu den Erkenntnissen, die sich mit Hilfe der Bedienungstheorie gewinnen lassen, gehören Aussagen zur Auslastung einzelner Stationen bzw. des Gesamtsystems, durchschnittliche Warteschlangenlängen oder mittlere Antwortzeiten.


Die Information dieses Kapitels basiert in wesentlichen Teilen auf der Darstellung in [1].

3.1.1 Geschichte


1 Agner Krarup Erlang, 1879-1929
2 Andrei Andrejewitsch Markov, 1856-1922

3.1.2 Begriffe und Bezeichnungen


Die folgenden Bezeichnungen werden im Verlauf dieses Kapitels genutzt:

- $N$ Anzahl der Knoten im Bedienungsnetz
- $K$ Gesamtzahl der Forderungen im Netz
- $(k_1, k_2, \ldots, k_N)$ Zustand des Warteschlangennetzes
- $k_i$ Forderungszahl im i-ten Knoten, $1 \leq i \leq N$
- $m_i$ Anzahl paralleler Bedieneinheiten in Knoten $i$
- $\mu_i$ Bedienrate von Knoten $i$
- $p_{ij}$ Wahrscheinlichkeit, daß ein in Knoten $i$ fertig bedienter Auftrag zu Knoten $j$ wechselt
- $p_{i0}$ Wahrscheinlichkeit, daß ein Auftrag nach Bedienung bei Knoten $i$ das Netz verlässt
- $\lambda_i$ gesamte mittlere Ankunftsrate von Aufträgen bei Knoten $i$
- $\lambda_{0i}$ mittlere Ankunftsrate von außen bei Knoten $i$

3.2 Elementare Wartesysteme

Eine Bedienstation (Abb.3.1) besteht aus einer oder mehreren identischen Bedieneinheiten und einer Warteschlange. Erreicht eine Forderung die Bedienstation, wird sie sofort an einer freien
Bedieneinheit bearbeitet oder reiht sich in die Warteschlange ein, falls alle Bedieneinheiten belegt sind. Sind auch alle Warteplätze belegt, geht die Forderung verloren. Die Rate, mit der die Forderungen die Bedienstation erreichen, wird als \textit{Ankunftsrate} \( \lambda \) bezeichnet. Kehrwert der Ankunftsrate ist die \textit{Zwischenankunftszeit}. Die Rate, mit der die Bedienstation Forderungen fertig bedient wird als \textit{Bedienrate} \( \mu \) bezeichnet, ihr Kehrwert ist die \textit{Bedienzeit}. Zwischenankunfts- und Bedienzeit sind im allgemeinen Zufallsgrößen und durch eine Verteilung gegeben. Die Bedienrate kann auch abhängig von der Anzahl der Forderungen in der Warteschlange der Bedienstation sein.

Bei einer Bedienstation mit \( m \) Bedieneinheiten, von denen jede eine Bedienrate \( \mu \) besitzt, gilt für die Bedienrate der Station \( \mu_{\text{gesamt}} = m \cdot \mu \).

Elementare Wartesysteme besitzen folgende Merkmale:

- Verteilung der Ankunftsrate \( \lambda \)
- Verteilung der Bedienrate \( \mu \)
- Anzahl der Bedieneinheiten
- maximale Länge der Warteschlange (kann auch null betragen oder unendlich sein)
- Warteschlangenstrategie

Die Ausprägungen dieser Merkmale bestimmen Vorhandensein und Komplexität einer Lösung für Modelle mit verschiedenen Typen von Bedienstationen.

### 3.2.1 Die Kendall-Notation

Zur Beschreibung von Bedienstationen wird die sogenannte \textit{Kendall-Notation}\(^7\) genutzt. Die oben genannten Merkmale einer Bedienstation werden auf ein Quintupel abgebildet, wie in Abbildung 3.2 dargestellt. Für die Verteilungen von Ankunfts- und Bedienrate sind folgende Kürzel üblich:

- **M** Exponentialverteilung
- **D** Deterministische Verteilung, d.h. konstante Zwischenankunfts- bzw. Bedienzeit
- **E\(_k\)** Erlang-Verteilung mit \( k \) Phasen

---

\(^7\) benannt nach Edward Calvin Kendall (1886-1972), us-amerikan. Biochemiker
Die Anzahl der Bedieneinheiten kann Werte von 1 bis \( \infty \) annehmen. Für die Anzahl der Warteplätze sind die Werte 0 bis \( \infty \) möglich. Sind unendlich viele Bedieneinheiten vorhanden, wird keine Warteschlange benötigt. Für jeden ankommenden Auftrag ist sofort eine Bedieneinheit verfügbar. Dieser Typ Bedienstation wird als \emph{Infinite Server (IS)} bezeichnet. Einige gebräuchliche Bedienstrategien sind die folgenden:

- **FCFS (First Come First Served):** entspricht dem FIFO-Prinzip
- **LCFS (Last Come First Served):** entspricht dem LIFO-Prinzip
- **RR (Round Robin):** jede Forderung bekommt eine Zeitscheibe fester Größe zur Bearbeitung zugeteilt. Kann die Forderung nicht fertiggestellt werden, reiht sie sich wieder in die Warteschlange ein. Diese wird nach FCFS abgearbeitet.
- **PS (Processor Sharing):** entspricht Round Robin, wobei die Länge der Zeitscheibe gegen null geht. Kann zur Modellierung von Multitasking-Systemen genutzt werden.
- **SIRO (Service In Random Order):** Die Abarbeitung der Warteschlange erfolgt in zufälliger Reihenfolge.


### 3.2.2 Exponentialverteilung: Markov- und Poisson-Eigenschaft

vom gegenwärtigen Zustand, nicht jedoch von der Vergangenheit abhängt. Formal wird dieser Sachverhalt durch Formel 3.1 für diskrete und durch Formel 3.2 für kontinuierliche stochastische Prozesse ausgedrückt.

\[
P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \ldots, X(t_1) = x_1) = P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n)
\]

(3.1)

\[
P(T \leq s + t | T > s) = P(T \leq t)
\]

(3.2)

Eine Exponentialverteilung liegt immer dann vor, wenn die einzelnen Ereignisse eines Ankunft- oder Bedienprozesses voneinander unabhängig sind. Diese Annahme ist nicht immer realistisch.


Es existieren eine Reihe von Verteilungen, um nichtexponentielle Verteilungen durch „Zusammenschaltung“ mehrerer exponentieller Bestandteile mit beliebiger Genauigkeit zu approximieren, z.B. die Hyperexponential-Verteilung und die Erlang-k-Verteilung.
3.2.3 Leistungsgrößen

Auslastung $\rho$  
$$\rho = \frac{\text{Ankunftsrate}}{\text{Bedienrate}} = \frac{\lambda}{\mu},$$  
bzw. bei $m$ Bedieneinheiten $\rho = \frac{\lambda}{m \mu}$

Durchsatz $\lambda$  
$$\lambda$$

Wartezeit $w$  
Zeit, die eine Forderung in der Warteschlange verbringt, bis die Bearbeitung beginnt. $\bar{w}$ = mittlere Wartezeit.

Antwortzeit $t$  
Gesamtzeit, die eine Forderung im Wartesystem verbringt (auch Verweilzeit genannt)  
mittlere Antwortzeit $\bar{t} = \overline{t} + \frac{1}{\mu}$

Warteschlangenlänge $Q$  
Anzahl der Aufträge in der Warteschlange

Füllung $k$  
Anzahl der Aufträge im Wartesystem  
$$\bar{k} = \sum_{k=1}^{\infty} k \cdot p(k)$$

Gesetz von Little  
$$\bar{Q} = \lambda \bar{w}$$

3.3 Bedienungsnetze


3.3.1 Ein- und Mehrklassennetze

In Bedienungsnetzen mit einer Forderungsklasse werden alle Forderungen prinzipiell gleich behandelt. Um verschiedene Auftragsprioritäten zu modellieren, sind Mehrklassennetze anwendbar. Hier existieren mehrere Forderungsklassen, und die Bedienstationen weisen für verschiedene Forderungsklassen unterschiedliche Bedienraten auf.
Ein Nachteil von Mehrklassennetzen ist, daß sie wesentlich empfindlicher auf Parameterschwankungen reagieren als Einklassennetze. Bei Einklassennetzen wirken sich Schwankungen der Systemparameter etwa linear auf die Ergebnisse des Modells aus. Bei Mehrklassennetzen hingegen können 5% Parameterschwankung zu 50% Ergebnisabweichung führen ([1]). Die Anwendung von Mehrklassennetzen ist daher besonders kritisch zu sehen, wenn Parameter des Systems geschätzt werden müssen und größere Abweichungen wahrscheinlich sind.

### 3.3.2 Offene und geschlossene Netze


### 3.3.3 Markovketten, Zustandsübergangsdiagramm


\[ p_{ij}(t) = P(X_{t+1} = x_j | X_t = x_i) \text{ für } i, j = 1\ldots m \]
als Übergangsmatrix dargestellt werden:

\[
(p_{ij}(t)) = M = \begin{pmatrix}
    p_{11}(t) & \cdots & p_{1m}(t) \\
    \vdots & \ddots & \vdots \\
    p_{m1}(t) & \cdots & p_{mm}(t)
\end{pmatrix}
\]

Übliches Mittel zur Darstellung von Markovketten sind Zustandsübergangsdiagramme. Die Knoten sind die Zustände der Kette, die Kanten die Zustandsübergänge, beschriftet mit den entsprechenden Übergangswahrscheinlichkeiten. Übergänge mit Übergangswahrscheinlichkeit \( p = 0 \) werden nicht dargestellt.


Bei geschlossenen Netzen ist der Zustandsraum der zugehörigen Markov-Kette endlich. Für ein Netz mit \( N \) Knoten und insgesamt \( K \) Forderungen ist der Zustandsraum folgende Menge von \( N \)-Tupeln:

\[
Z = \{ (k_1, k_2, \ldots, k_N) | \sum_{n=1}^{N} k_n = K \}.
\]

Bei offenen Netzen hingegen ist die Zustandsmenge unendlich, da \( K \) in diesem Fall nicht konstant ist, sondern unendlich viele Werte annehmen kann.

Eine im Zusammenhang mit Bedienungstheorie wichtige Eigenschaft, die Markovketten aufweisen können, ist die der Ergodizität. Eine Markovkette ist ergodisch, wenn sie folgende Voraussetzungen erfüllt:

- die Kette muß irreduzibel sein, d.h. jeder Zustand muß von jedem anderen Zustand aus erreichbar sein.
- die Kette muß aperiodisch, d.h. alle Zustände i haben die Periode 1. Dabei ist die Periode der ggT aller Werte \( n \) für die gilt:

\[
P(X_n = i, X_0 = i, X_k \neq i \text{ für } k = 1 \ldots n-1) > 0
\]

3.3.4 Gleichgewicht

Bedienmodelle sind dynamische Modelle. Das heißt, daß sich im Laufe der Zeit der Systemzustand und die Leistungsgrößen des Systems ändern. Lösungen für Momentaufnahmen des dynamischen Ablaufs sind oft nur schwer oder gar nicht zu bestimmen.

Im allgemeinen sind aber Lösungen für den \textit{Gleichgewichtszustand} des Systems ausreichend. Im Gleichgewichtszustand sind die Einschwingvorgänge eines Warteschlangensystems abgeklungen, die Zustandswahrscheinlichkeiten und die Leistungskenngrößen sind dann zeitunabhängig und unabhängig vom Anfangszustand des Prozesses.
Voraussetzung für die Existenz des Gleichgewichtszustandes ist, daß die das System beschreibende Markov-Kette ergodisch ist. (Siehe Abschnitt 3.3.3.)

### 3.3.5 Globale Gleichgewichtsgleichungen

Die *globalen Gleichgewichtsgleichungen* können aus dem Gleichungssystem zur Bestimmung der Gleichgewichts-Zustandswahrscheinlichkeiten, \( \mathbf{p} \cdot \mathbf{Q} = \mathbf{0} \), hergeleitet werden. Sie besagen, daß für jeden Zustand \( S_i \) des Bedienungsnetzes die Wahrscheinlichkeit für das Entstehen des Zustandes gleich der Wahrscheinlichkeit für das Verschwinden des Zustandes sein muß. Formal:

\[
\sum_{S_j} p(S_j) q_{ji} = p(S_i) \sum_{S_j} q_{ij}.
\]

Zur Veranschaulichung soll ein geschlossenes Bedienungssystem mit \( N = 2 \) Knoten und \( K = 3 \) Forderungen dienen. Das zugehörige Zustandsübergangsdiagramm ist in Abbildung 3.6 dargestellt. Die globale Gleichgewichtsgleichung für Zustand \((2,1)\) - im Diagramm gelb hervorgehoben - lautet

\[
(\mu_1 + \mu_2) \cdot P(2,1) = \mu_1 \cdot P(3,0) + \mu_2 \cdot P(1,2).
\]

Globale Gleichgewichtsgleichungen lassen sich nur für geschlossene Bedienungsnetze aufstellen, da in jeder Gleichung alle Systemzustände vorkommen. Offene Bedienungsnetze besitzen jedoch einen unendlichen Zustandsraum.

Zur Lösung der globalen Gleichgewichtsgleichungen können numerische Verfahren genutzt werden, z.B.

- die iterative numerische Methode,
- die direkte numerische Methode und
- die rekursive numerische Methode.

Diese Methoden sind jedoch nur für kleine Netze mit wenigen Knoten und wenigen Forderungen praktikabel.
3.3.6 Lokales Gleichgewicht

Die globalen Gleichgewichtsgleichungen können in lokale Gleichgewichtsgleichungen zerlegt werden. Diese besagen, daß die Rate, mit der ein Zustand \( S \) eines Warteschlangenmodells aufgrund des Abgangs einer Forderung an einem Knoten verlassen wird, gleich ist der Rate, mit der er durch Zugang einer Forderung an diesem Knoten erreicht wird. Ein Bediennetz, für das die lokalen Gleichgewichtsgleichungen lösbar sind, besitzt die local-balance Eigenschaft.

Für das Beispiel in Abb. 3.6 lassen sich folgende lokalen Gleichgewichtsgleichungen aufstellen:

\[
\begin{align*}
P(2,1) \cdot \mu_1 &= P(1,2) \cdot \mu_2 \\
P(2,1) \cdot \mu_2 &= P(3,0) \cdot \mu_1 \\
P(1,2) \cdot \mu_1 &= P(0,3) \cdot \mu_2
\end{align*}
\]

Besitzt ein Netz die local-balance Eigenschaft, ist es zur Bestimmung der Gleichgewichtszustandswahrscheinlichkeiten gar nicht erforderlich, die Gleichgewichtsgleichungen zu lösen. Die Lösung für den Zustand des Netzes kann aus Faktoren zusammengesetzt werden, die sich auf einzelne Knoten beziehen:

\[
p(S_1,S_2,\ldots,S_N) = \frac{1}{G} p(S_1) \cdot S_2 \cdot \ldots \cdot S_N.
\]

Man spricht deshalb auch von Produktformnetzen.

Existiert eine eindeutige Lösung der lokalen Gleichgewichtsgleichungen, ist diese gleichzeitig Lösung der globalen Gleichgewichtsgleichungen.

Die local-balance Eigenschaft ist gegeben für Netze, die ausschließlich aus den folgenden Typen elementarer Wartesysteme bestehen:

- **M/M/m-FCFS** (Typ 1). Dabei müssen die Bedienraten für Aufträge verschiedener Klassen identisch sein.
- **M/G/1-PS(RR)** (Typ 2)
- **M/G/\infty** (Typ 3)
- **M/G/1-LCFS PR** (Typ 4)

3.3.7 Jackson Theorem

Mit dem Jackson-Theorem lassen sich offene Warteschlangennetze mit einer Forderungsklasse untersuchen. Weitere Voraussetzungen bzw. Eigenschaften sind:

- Gesamtzahl der Forderungen \( K \) ist nicht beschränkt
- jeder Knoten kann Ankünfte von außen haben (exponentiell verteilte Zwischenankunftszeiten)
- Abgänge von Aufträgen sind an jedem Knoten möglich
3.3 Bedienungsnetze

- exponentielle Bedienzeiten
- es treten ausschließlich FCFS-Knoten auf
- es sind mehrere Bedieneinheiten pro Knoten möglich
- die Bedienraten und die Ankunftsrate $\lambda_0$ können von der Auftragszahl im jeweiligen Knoten abhängen

**Theorem von Jackson:** Wenn für alle Knoten $i = 1 \ldots N$ im offenen Netz die Stabilitätsbedingung $\lambda_i < \mu_i \cdot m_i$ erfüllt ist, wobei sich die Ankunftsrate $\lambda_i$ aus der Gleichung

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} \lambda_j p_{ji} \text{ für } i = 1, \ldots, N$$

ergeben, dann ist die Wahrscheinlichkeit für den Gleichgewichtszustand des Netzes durch das Produkt der Zustandswahrscheinlichkeiten der einzelnen Knoten gegeben:

$$p(k_1, k_2, \ldots, k_N) = p_1(k_1) \cdot p_2(k_2) \cdot \ldots \cdot p_N(k_N).$$

Die Analyse eines Bediennetzes nach dem Jackson-Theorem gliedert sich in 3 Schritte:

1. **Schritt**
   - Bestimmung der Ankunftsrate $\lambda_i$ an den einzelnen Knoten
   
   $$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} \lambda_j \cdot p_{ij}$$

2. **Schritt**
   - Überprüfung der Stabilitätsbedingung $\rho < 1$ und Bestimmung der Zustandswahrscheinlichkeiten und Leistungsgrößen für jeden Knoten einzeln
   
   $$p_i = (1 - p_i) \rho_i^2$$

3. **Schritt**
   - Bestimmung der Zustandswahrscheinlichkeiten für das gesamte Netz:

   $$p(k_1, k_2, \ldots, k_N) = p_1(k_1) \cdot p_2(k_2) \cdot \ldots \cdot p_N(k_N)$$

### 3.3.8 Weitere Verfahren

Das Jackson-Theorem wurde durch Gordon/Newell (Gordon/Newell-Theorem) und Baskett, Chandy, Muntz und Palacios (BCMP-Theorem) auf geschlossene Netze bzw. Mehrklassennetze und verschiedene Warteschlangenstrategien erweitert.

Trotz formelmäßiger Einfachheit schränkt die aufwändige Berechnung der sog. Normalisierungsconstanten bei geschlossenen Netzen die Anwendbarkeit der Verfahren ein. Abhilfe schaffen effiziente Algorithmen, wie z.B. der Faltungsalgorithmus und die Mittelwertanalyse.

Außerdem existieren eine Vielzahl approximativer Verfahren, z.B. die Summationsmethode und die Grenzwertanalyse. Es existieren auch approximative Verfahren zur Lösung von Nicht-Produktformnetzen (Dekompositionsverfahren, Erweiterungen der Mittelwertanalyse, Produktformapproximationen).
Kapitel 4
Modellierung von WLAN-Netzen

In diesem Kapitel soll ein Modell zur Kapazitätsanalyse von WLAN-Funknetzen vorgestellt sowie Überlegungen und Entscheidungen auf dem Weg zu dessen Entstehung dargelegt werden.

Da die Ergebnisse dieser Arbeit in ein Kapazitätsanalyse-Werkzeug für allgemeine Rechnernetze einfließen sollen, wird immer wieder auf diesen allgemeinen Kontext Bezug genommen.

Grundsatz der Modellierung war, zunächst ein einfach handhabbares Modell zu erstellen. Es besteht nicht der Anspruch, im Nachkommabereich exakte Ergebnisse zu erhalten. Dies wäre angesichts der Unsicherheit der Eingabegrößen ohnehin unrealistisch (siehe Abschnitt 4.2). Ziel ist vielmehr ein praktikables Modell, welches für die Netzwerkplanung brauchbare Ergebnisse schnell bereitstellen kann. Hier geht es also eher um das Abschätzen von Größenordnungen.

Das Modell soll flexibel genug sein, um den Einsatz verschiedener Protokolle auf den höheren Netzwerkschichten sowie die Nutzung der verschiedenen WLAN-Standards (802.11, 802.11b, a, h, g) abzubilden.

Der Modellierungsprozeß gliedert sich in mehrere Teilaufgaben. Diese werden in Abbildung 4.1 dargestellt.

Bei der Lastanalyse und -modellierung sind die im realen System auftretenden Datenlasten auszuwerten und in einer Form darzustellen, die im Modell genutzt werden kann. Abschnitt 4.2 widmet sich dieser Problematik.

Abbildung 4.1: Teilaufgaben bei der Modellierung

Bei der anschließenden *Parametrisierung* (Abschnitt 4.4) steht die Bestimmung der Parameter der einzelnen Bedienstationen sowie der Verzweigungswahrscheinlichkeiten im Mittelpunkt.

Zunächst werden im folgenden Abschnitt die Methoden mathematische Modellierung und Simulation gegenübergestellt.

### 4.1 Vergleich Mathematische Modellierung und Simulation

Bei der Neuplanung von Rechnernetzen bietet sich neben der hier im Mittelpunkt stehenden mathematischen Modellierung mit Hilfe der Bedienungstheorie die Simulation als alternatives Verfahren zur Kapazitätsanalyse an. Beide Methoden besitzen Vor- und Nachteile, die es gegeneinander abzuwägen gilt.

Die Simulation liefert in der Regel exaktere Ergebnisse, da das Verhalten des modellierten Systems detailliert nachgebildet wird und weniger Abstraktionen als bei mathematischer Modellierung erforderlich sind. Dies bedingt zwei weitere Eigenschaften der Simulation: sie kann prinzipiell einfach realisiert werden, wobei allerdings die Schaffung eines Simulators in der Regel einen erheblichen Aufwand darstellt.

Mit steigender Leistungsfähigkeit der Rechentechnik verkürzt sich die Zeit, die zur Durchführung von Simulationen erforderlich ist, dennoch liefert ein mathematisches Modell schneller Ergebnisse.

Der Vorteil der größeren Genauigkeit wird dadurch relativiert, daß das Ergebnis der Simulation - wie auch der mathematischen Modellierung - in kritischer Weise von den Datenlasten abhängt, die der Modellierung zugrunde gelegt werden.

### 4.2 Lastmodellierung


4.2 Lastmodellierung


Bei Nutzung *statistischer Verteilungen* wird eine zu den gewonnenen Meßwerten passende Verteilungsfunktion bestimmt. Diese wird dann genutzt, um Parameter der Last im Modell zu bestimmen.

Da bei beiden Verfahren die Gewinnung von Meßwerten am Anfang steht, kommt es darauf an, daß die der Messung zugrunde gelegte Situation repräsentativ ist. Dabei spielt nicht nur das Datenvolumen eine Rolle, sondern unter anderem auch die Anzahl der Nutzer, die Häufigkeit, mit der diese das Netzwerk in Anspruch nehmen sowie die verschiedenen auftretenden Anwendungs- typen, bezogen auf die die Art der Last, die sie erzeugen (Bursts, Streams, ...).


[2] nennt vier Merkmale zur Beschreibung des Datenverkehrsaufkommens von Netzwerkanwendungen:

- Verteilung der Zeit zwischen Anwendungsaufträgen
- Dauer einer Verbindung (Sitzung)
- Anzahl Bytes, die während einer Verbindung übertragen werden
- Zwischenankunftszeit der Pakete innerhalb einer Verbindung

Für Parameter verschiedener allgemeiner Anwendungen liegen Vorschläge von Verteilungen zu deren Modellierung vor. Zum Beispiel wurden für die Modellierung der Zwischenankunftszeiten zwischen FTP- und Telnet-Sitzungen die Exponentialverteilung und die Weibull-Verteilung vorgeschlagen ([2]). Die Größen der Dateien, die bei HTTP-Nutzung übertragen werden, wurden mit der Pareto-Verteilung charakterisiert ([30]).

Kap. 4 Modellierung von WLAN-Netzen

stiegene globale Verflechtung der Unternehmen einen gesteigerten Kommunikationsbedarf über Intranet-Grenzen hinweg.


Im Fall von Firmen-Intranets ist diese Aussage zu relativieren. Hier kann der Datenverkehr durch spezielle firmenintern entwickelte Software oder branchentypische Software dominiert werden. Für Datentransfers derartiger Software sowie neuer Software sind statistische Verteilungen erst noch zu bestimmen.

Oftmals ist es nicht möglich, eine statistische Verteilung für Datenverkehrsmerkmale zu bestimmen, dann kann auf ein empirisches Modell zurückgegriffen werden. [3] gibt folgendes Vorgehen zur Erstellung eines solchen Modells an:

1. Festlegung eines Blickwinkels, aus dem die Last untersucht werden soll
2. Auswahl einer Menge von Parametern, die die relevanten Merkmale der zu untersuchenden Last abbilden
3. Messung am System, um Performance-Rohdaten zu gewinnen
4. Analyse und Reduzierung der Performancedaten
5. Modell konstruieren
6. Modell überprüfen


So wurde etwa festgestellt, daß die Verteilung der Größen per HTTP übertragenen Dateien heavy tailed\(^1\) ist ([30]). Die Aggregation von heavy-tailed-Datenquellen führt zu einem long-range-dependent Datenstrom. Der besitzt im Gegensatz zu aggregierten Poisson-Strömen die Eigenschaft, daß zeitlich weit voneinander entfernte Parameter des Stromes miteinander korreliert sein können.

Auch andere Parameter wie Burst-Längen, Leerlaufzeiten und Verbindungsdauern ([2]) wurden als heavy-tailed charakterisiert.

Zur Selbstähnlichkeit von Internet-Traffic liegen widersprüchliche Untersuchungsergebnisse vor ([30]). Selbstähnlichkeit bedeutet, daß ein Prozeß über mehrere Zeitskalen hinweg die gleichen statistischen Merkmale aufweist.

\(^1\) heavy tailed: innerhalb einer Reihe von Werten gibt es einige, die zwar nach ihrer Anzahl vernachlässigbar wären, die jedoch so sehr vom Durchschnitt abweichen, daß sie das Ergebnis entscheidend beeinflussen.
4.3 Strukturelle Modellierung


Für das Funktionieren des Netzwerkes notwendige Software ist das Betriebssystem, besonders Netzwerktreiber und -protokollimplementierungen. Außerdem sind die Anwendungen und Middleware, die die bereitgestellte Netz-Infrastruktur nutzen, als Bestandteil des Netzwerks zu sehen.

Die Netzwerkgeräte werden im Bedienungsmodell durch Bedienstationen dargestellt. Die Eigenschaften der genutzten Netzwerkprotokolle sowie der Treibersoftware fließen in die Bedienraten der Stationen ein.

Abbildung 4.3 zeigt verschiedene denkbare Detailstufen der Abbildung. Bei Modellierung einzel-


Deswegen sollte die Detailstufe des Modells entsprechend den Anforderungen des jeweiligen Gerätes gewählt werden und nicht generell für das gesamte Modell festgelegt werden.


Auch die Möglichkeit, den Access Point und jeden Client durch einzelne Bedienstationen zu repräsentieren (Detailstufe 2) wurde verworfen. Der Durchsatz, den eine einzelne WLAN-Station erzielen kann, ist in starkem Maße von den anderen Stationen im BSS abhängig - sowohl von deren Kanalbedingungen, als auch den durch sie erzeugten Datenströmen. Im Bedienmodell sind solche Abhängigkeiten nicht dynamisch abbildbar, da die Bedienrate einer Bedienstation zwar von deren Warteschlangenlänge abhängen darf, nicht aber von der anderer Stationen. Die Wechselwirkungen könnten zwar bei der Bestimmung fester Bedienraten berücksichtigt werden. Das würde jedoch

<table>
<thead>
<tr>
<th>Detailstufe 3</th>
<th>Teilnetze, Netzelemente</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z.B. WLAN1, WLAN2, Ethernet1, Server1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detailstufe 2</th>
<th>Einzelne Geräte im Netzwerk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z.B. Switch, Gateway, Workstation, Notebook, Access Point</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detailstufe 1</th>
<th>Einzelne Hardware-Komponenten</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z.B. Prozessor, Hauptspeicher, Netzwerkkarte</td>
</tr>
</tbody>
</table>

**Abbildung 4.3:** Mögliche Detailstufen der Abbildung eines Rechnernetzes


Deswegen sollte die Detailstufe des Modells entsprechend den Anforderungen des jeweiligen Gerätes gewählt werden und nicht generell für das gesamte Modell festgelegt werden.


Auch die Möglichkeit, den Access Point und jeden Client durch einzelne Bedienstationen zu repräsentieren (Detailstufe 2) wurde verworfen. Der Durchsatz, den eine einzelne WLAN-Station erzielen kann, ist in starkem Maße von den anderen Stationen im BSS abhängig - sowohl von deren Kanalbedingungen, als auch den durch sie erzeugten Datenströmen. Im Bedienmodell sind solche Abhängigkeiten nicht dynamisch abbildbar, da die Bedienrate einer Bedienstation zwar von deren Warteschlangenlänge abhängen darf, nicht aber von der anderer Stationen. Die Wechselwirkungen könnten zwar bei der Bestimmung fester Bedienraten berücksichtigt werden. Das würde jedoch
zu Problemen mit der Interpretation der Ergebnisse führen. Würde das Modell eine Überlastung einer Bedienstation anzeigen, wäre die Aussage auch nur, daß in dem betreffenden WLAN-BSS ein Problem besteht, da die Ursache für die Überlastung bei einer anderen Station liegen kann.


4.3.1 Andere mögliche Konstellationen

Neben dem in Abbildung 4.4 betrachteten Fall, daß ein Access Point mobile Clients mit einem kabelgebundenen LAN verbindet, sind noch zwei weitere Einsatzszenarien im Rahmen des WLAN-Infrastrukturmodus möglich, die bei der Modellierung gesonderter Berücksichtigung bedürfen.


Es gibt Access Points, die über zwei unabhängige WLAN-Interfaces verfügen. Dadurch eröffnen sich verschiedene Nutzungsmöglichkeiten:

1. Realisierung einer Wireless Bridge mit doppeltem Datendurchsatz durch Parallelschaltung
Abbildung 4.6: Beispiel zur Modellierung eines AP mit 2 Interfaces

beider Kanäle

2. Anbindung weiter entfernter Clients an ein LAN. Das eine WLAN-Interface bildet mit einem am LAN angeschlossenen normalen AP eine Wireless Bridge, am anderen können sich die Clients anmelden.


Dualband Access Points, also solche, die über je ein Interface verschiedener WLAN-Technologien verfügen (z.B. 802.11g und 802.11a), sind von der Modellierung her genauso zu behandeln. Die praktische, programmtechnische Umsetzung der hier dargestellten Modellierungsprobleme wird in Kapitel 5 angesprochen.

4.3.2 Typ der Bedienstation


Da eine Bedienstation mehrere sich einen Funkkanal teilende WLAN-NICs modelliert, und dieser Kanal nur für eine Übertragung gleichzeitig genutzt werden kann, wird $n = 1$ gewählt.
4.3 Strukturelle Modellierung


4.3.3 Forderungen

Bei der strukturellen Abbildung ist noch die Frage zu klären, welches Objekt des realen Systems einer Forderung im Modell entsprechen soll. Wünschenswert kann dabei eine möglichst intuitive Abbildung sein, d.h., daß eine Forderung einem natürlichen Objekt des realen Systems entspricht. Für die Handhabbarkeit des Modells kann es jedoch wichtiger sein, eine Forderung so zu wählen, daß die bei den Berechnungen auftretenden Zahlen in akzeptablen Größenordnungen bleiben.

Bei der Modellierung von Rechnernetzen spiegeln die Forderungen die übertragenen Daten wider. Es sind verschiedene Datenmengen als Einheit der Forderung denkbar:

- 1 Bit
- 1 Byte
- 1 Frame/Paket einer bestimmten Protokollschicht
- 1 „abstraktes“ Paket einer bestimmten Größe

4.4 Parametrisierung des Modells

Eine der Hauptschwierigkeiten bei der Modellierung liegt in der Parametrisierung des Modells. Es geht dabei hauptsächlich um die Bestimmung der Bedienraten der einzelnen Bedienstationen, so daß sie das Verhalten des modellierten Systems widerspiegeln.


- **Sättigungs-Annahme**: Jede Station hat immer mindestens ein Frame zur Aussendung bereit. Jede Station konkurriert nach erfolgreicher Übertragung eines Frames sofort wieder um die Kanalnutzung.
- Alle Stationen besitzen eine identische Verteilung der Paket-Zwischenankunftszeiten und der Paketgrößen.
- Es liegt ein idealer Kanal vor, d.h. jede Station kann die Übertragungen aller anderen Stationen empfangen.

Diese Annahmen können gemacht werden, da es um generelle Protokolleigenschaften geht.


Dabei wird die Bedienrate aus Einzelausdrücken zusammengesetzt, von denen jeder einen Faktor

\[ \text{engl.: saturation} \]
4.5 Zu berücksichtigende Aspekte

4.5.1 Unterschiedliche Brutto-Datenraten

Auf Bitübertragungsebene setzen die WLAN-Standards (801.11, 802.11b, 802.11a/h und 802.11g) unterschiedliche Modulationen ein, um sich an verschiedene Übertragungsbedingungen anzupassen. Daraus resultieren unterschiedliche Brutto-Datenraten.

Es kann also z.B. in einem 802.11a-BSS eine Station aufgrund starker Dämpfung mit 18 MBit/s arbeiten, während die übrigen die maximale Datenrate 54 MBit/s nutzen können. Eine diese Station vermindert damit jedoch auch den Daten-Durchsatz der anderen Stationen des BSS. Das liegt daran, daß das auf der 802.11 MAC-Schicht genutzte CSMA/CA-Verfahren langfristig allen Stationen ungeachtet ihrer Datenrate die gleiche Chance auf Zugriff auf den Kanal einräumt. Die maximale MAC-Frame-Größe ist ebenfalls unabhängig von der Datenrate der jeweiligen Station einheitlich festgelegt. Nimmt man die Dauer einer Übertragung durch eine schnelle Station als Bezugsgröße, so wird der Kanal bei einer Übertragung einer langsameren Station proportional zum Geschwindigkeitsunterschied länger belegt. (Dabei wird angenommen, daß die Verteilung der MSDU-Größen für alle Stationen gleich ist.)

Wenn man von der Sättigungsannahme ausgeht, wird damit der Geschwindigkeitsvorteil der schnelleren Stationen komplett beseitigt. Der Durchsatz aller Stationen sinkt auf einen Wert in Größenordnung des Durchsatzes der langsamsten Station.


Das beschriebene Phänomen wird im Modell dieser Arbeit berücksichtigt, indem aus den Brutto-Datenraten der Stationen des BSS eine Durchschnittsdatenrate des Kanals berechnet wird. Das folgende Beispiel für OFDM (802.11a) soll das Vorgehen verdeutlichen: zwei Notebooks (N1
<table>
<thead>
<tr>
<th>Bitrate Client1</th>
<th>Durchsatz mit UDP [MBit/s]</th>
<th>Durchsatz mit TCP [MBit/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Client1</td>
<td>Client2</td>
</tr>
<tr>
<td>11</td>
<td>3,09</td>
<td>3,36</td>
</tr>
<tr>
<td>5,5</td>
<td>2,38</td>
<td>2,42</td>
</tr>
<tr>
<td>2</td>
<td>1,30</td>
<td>1,26</td>
</tr>
<tr>
<td>1</td>
<td>0,76</td>
<td>0,76</td>
</tr>
<tr>
<td>11</td>
<td>2,26</td>
<td>2,0</td>
</tr>
<tr>
<td>5,5</td>
<td>2,01</td>
<td>1,56</td>
</tr>
<tr>
<td>2</td>
<td>1,17</td>
<td>0,90</td>
</tr>
<tr>
<td>1</td>
<td>0,74</td>
<td>0,58</td>
</tr>
<tr>
<td>11</td>
<td>1,71</td>
<td>1,41</td>
</tr>
<tr>
<td>5,5</td>
<td>1,66</td>
<td>1,16</td>
</tr>
<tr>
<td>2</td>
<td>0,96</td>
<td>0,84</td>
</tr>
<tr>
<td>1</td>
<td>0,69</td>
<td>0,47</td>
</tr>
</tbody>
</table>

Tabelle 4.1: Durchsatz bei unterschiedlichen Datenraten (Quelle: [26])

und N2) sind an einem Access Point angemeldet, N1 arbeitet mit der maximalen Datenrate 54 MBit/s, N2 kann aufgrund starker Dämpfung nur 6 MBit/s nutzen. Betrachtet wird ausschließlich die Bitübertragungsschicht. Die durchschnittliche Größe der übertragenen Pakete (PSDUs) wird mit 1500 Byte angenommen. Das MAC-Protokoll garantiert langfristig gleiche Kanalzugriffschancen für alle Stationen. Deshalb kann die durchschnittliche Datenrate des Kanals durch Bildung der mittleren Übertragungsdauer für ein Paket jeder Station bestimmt werden. Die Übertragungszeiten betragen für N1 und N2:

\[ t_{tr}^{N1} = \frac{1500\text{Byte} \times 8\text{Bit}}{54\text{MBit/s}} = 0,22\text{ms} \]
\[ t_{tr}^{N2} = \frac{1500\text{Byte} \times 8\text{Bit}}{6\text{MBit/s}} = 2\text{ms} \]

Das arithmetische Mittel daraus ist \( \bar{t}_{tr} = 1,1\text{ms} \). Teilt man die Paketgröße in Bit durch diesen Wert, ergibt sich die mittlere Datenrate \( \bar{r} \) des Kanals:

\[ \bar{r} = \frac{12000\text{Bit}}{1,1\text{ms}} = 10,8\text{MBit/s} \]


Im allgemeinen Fall ergibt sich \( \bar{r} \) als Funktion der Einzeldatenraten \( r_1, \ldots, r_n \) und der Anzahl der Stationen \( n \):

\[ \bar{r}(r_1, \ldots, r_n, n) = \frac{1}{\frac{1}{r_1} + \frac{1}{r_2} + \cdots + \frac{1}{r_n}} = \frac{n}{\frac{1}{r_1} + \frac{1}{r_2} + \cdots + \frac{1}{r_n}}. \]
4.5 Zu berücksichtigende Aspekte

4.5.2 Verluste durch Wettbewerb (Contention)


Die fünfte dargestellte Kurve entspricht dem Mittelwert der anderen.

In [32] findet sich folgende Definition des normalisierten Systemdurchsatzes: „...we consider normalized system throughput as the fraction of time the channel is used to successfully transmit..."
payload bits". Das heißt, daß sich nicht nur der Wettbewerb um den Kanal, sondern auch Protokolloverheads und andere Faktoren mindernd auf die dargestellten Durchsatzzwerte auswirken. Auch die anderen in Abbildung 4.7 dargestellten Ergebnisse entsprechen dieser Voraussetzung.

Werte für weniger als 5 konkurrierende Stationen liegen nur in [17] vor. Für den Fall $n = 1$, wenn eine Station allein an einem Access Point angemeldet ist und nur Daten von der Station zum Access Point fließen (bis auf ACKs), kann angenommen werden, daß der normalisierte Durchsatz nur durch die Overheads von WLAN-MAC und PHY reduziert wird. Bei den gegebenen Parametern belaufen sich diese Overheads auf 10%, der Wert für den normierten Durchsatz ist damit 0,9. Diese Werte werden der Durchschnittskurve hinzugefügt. Die resultierende Kurve wurde im Intervall $1 \leq n \leq 50$ durch die folgende Funktion approximiert:

$$f_{cont}(n) = \frac{6,70 \cdot n + 47,24}{0,02 \cdot n^2 + 7,58 \cdot n + 38,32} - 0,28.$$  

Abbildung 4.8 stellt die Mittelwerte und $f_{cont}(n)$ dar.

Da Faktoren wie Protokolloverheads beim Modell dieser Arbeit separat berücksichtigt werden, sollte für $n = 1$ keine Durchsatzreduzierung bewirkt werden. Dazu wird $f_{cont}$ noch entlang der Y-Achse verschoben, so daß $f_{cont}(1) = 1$ gilt:

$$f_{cont}(n) = \frac{6,70 \cdot n + 47,24}{0,02 \cdot n^2 + 7,58 \cdot n + 38,32} - 0,17.$$  

4.5 Zu berücksichtigende Aspekte

dar. Rechts wurde der OFDM-PHY bei 6 MBit/s genutzt. In beiden Fällen beträgt die MSDU-Größe ebenfalls 1000 Byte.

Es ist ersichtlich, daß der Durchsatz bis zu einer bestimmten Anzahl beteiligter Stationen ansteigt, ohne daß gravierende Wettbewerbsverluste auftreten. Wird ein Schwellwert überschritten, fällt der Durchsatz ab, da sich die Stationen nun vermehrt beim Kanalzugriff gegenseitig behindern.

Es soll nun die insgesamt erzeugte Datenlast betrachtet werden, die an den Punkten maximalen Durchsatzes durch die Stationen erzeugt werden:

\[
\begin{align*}
 l_1 & = (\text{Anzahl Stationen}) \cdot \lambda \cdot g_p \\
 & = 3 \cdot 25 \cdot 1000 = 75000 \text{Byte} = 600000 \text{Bit} \\
 & \quad \text{(Abbildung 4.7, [32])} \\
 l_2 & = 12 \cdot 5 \cdot 1000 = 60000 \text{Byte} = 480000 \text{Bit} \\
 & \quad \text{(Abbildung 4.9, links)} \\
 l_3 & = 15 \cdot 25 \cdot 1000 = 375000 \text{Byte} = 30000000 \text{Bit} \\
 & \quad \text{(Abbildung 4.9), rechts}
\end{align*}
\]

Es zeigt sich, daß die Datenlasten jeweils etwa der Hälfte der genutzten Brutto-Datenrate entsprechen. Damit gilt für die Anzahl Stationen, ab der durch Wettbewerbsverluste eine Durchsatzreduzierung eintritt, folgende Faustformel:

\[
N = \frac{\frac{1}{2} r_{phy}}{\lambda \cdot g_p}
\]

wobei \(r_{phy}\) in \(\text{Bit/s}\) angegeben ist und die Paketgröße \(g_p\) in Bit.

Damit ließe sich der durch Wettbewerb reduzierte Durchsatz approximieren durch

\[
r_{cont} = \begin{cases} 
1 & \text{falls } n \leq N, \\
 r_{phy} \cdot f_{cont}(n - N) & \text{sonst.}
\end{cases}
\]
Dies gilt allerdings nur für den Fall, daß man für alle beteiligten Stationen die gleiche erzeugte Datenlast annimmt. Dies ist jedoch bei der hier zu betrachtenden Aufgabenstellung nicht der Fall.


### 4.5.3 Protokoll-Overhead

Ein wesentlicher Teil der Brutto-Bruttodatenrate wird durch den Overhead verbraucht, der durch die verschiedenen Protokollsichten entsteht. Je nachdem, auf welche Schicht des Protokollstacks sich die Angaben zur vorhandenen Datenlast beziehen, ist bei der Ermittlung der zur Verfügung stehenden Datenrate der Overhead der darunter liegenden Schichten zu berücksichtigen.


**Transportschicht**


UDP ist ein verbindungsloses Transportprotokoll. Es erfolgt keine Bestätigung empfangener Daten. UDP erweitert die übergebenen Daten um einen 8 Byte großen Header.
4.5 Zu berücksichtigende Aspekte

Vermittlungsschicht

Das IP-Protokoll fügt den übergebenen Daten einen 20 Byte großen Header hinzu.

Sicherungsschicht

Die LLC-Teilschicht fügt den an sie übergebenen Daten einen 3 Byte großen Header hinzu, der MAC Header hat zusammen mit der Frame Check Sequence eine Größe von 34 Byte.

Hinzu kommt der 5 Byte große Header des SNAP\textsuperscript{3}-Protokolls.


Bitübertragungsschicht

Die verschiedenen PHY-Realisierungen nutzen unterschiedlich lange PLCP-Präambeln und -Header. Deren Übertragungsdauer ist für die einzelnen Standards in Tabelle 2.3 (Seite 13) aufge-
listet.

Beim OFDM-Verfahren ist zusätzlich zu berücksichtigen, daß die Länge des Datenteils ein ganzzahliges Vielfaches der Anzahl der Datenbits pro OFDM-Symbol sein muß. Um dies zu gewährleisten, wird eine entsprechende Anzahl Pad-Bits eingefügt. Der potenzielle Einfluß der Pad-Bits auf die Übertragungsdauer eines Frames ist in Tabelle 4.2 dargestellt.

\( N_{\text{PAD}}^{\text{max}} \) gibt an, wieviele Pad-Bits bei der gegebenen Datenrate maximal aufgefüllt werden. Der Wert ist immer gerade, da der Datenteil - ohne die Pad-Bits - aus vollen Bytes besteht. Dadurch ist dieser Wert auch theoretisch, die tatsächlich möglichen Werte sind in der Regel etwas kleiner. \( t_{\text{max}} \) gibt die Übertragungsdauer für die maximale Anzahl Pad-Bits an und \( t_{\text{avg}} \) die Dauer zur Übertragung von \( \frac{1}{2} \cdot N_{\text{PAD}}^{\text{max}} \) Pad-Bits. Dieser Wert kann als Durchschnittswert angesetzt werden, wenn auf eine exakte Bestimmung verzichtet werden soll.

<table>
<thead>
<tr>
<th>Datenrate</th>
<th>( N_{\text{PAD}}^{\text{max}} )</th>
<th>( t_{\text{max}} [\mu s] )</th>
<th>( t_{\text{avg}} [\mu s] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>22</td>
<td>3,66</td>
<td>1,83</td>
</tr>
<tr>
<td>9</td>
<td>34</td>
<td>3,77</td>
<td>1,89</td>
</tr>
<tr>
<td>12</td>
<td>46</td>
<td>3,83</td>
<td>1,92</td>
</tr>
<tr>
<td>18</td>
<td>70</td>
<td>3,89</td>
<td>1,94</td>
</tr>
<tr>
<td>24</td>
<td>94</td>
<td>3,92</td>
<td>1,96</td>
</tr>
<tr>
<td>36</td>
<td>142</td>
<td>3,94</td>
<td>1,97</td>
</tr>
<tr>
<td>48</td>
<td>190</td>
<td>3,96</td>
<td>1,98</td>
</tr>
<tr>
<td>54</td>
<td>214</td>
<td>3,96</td>
<td>1,98</td>
</tr>
</tbody>
</table>

Tabelle 4.2: Auswirkung der Pad-Bits auf die Übertragungsdauer

Beispiel

Im folgenden sollen die besprochenen Overheads, die durch die einzelnen Protokollschichten entstehen, für ein Beispiel ermittelt werden. Es wird von 1460 Byte Daten ausgegangen, die als ein TCP-Segment übertragen werden sollen. Das TCP-Acknowledgement wird außer Betracht gelassen, da - wie oben erwähnt - die Zuordnung zu einem TCP-Segment problematisch ist.
4.5 Zu berücksichtigende Aspekte

<table>
<thead>
<tr>
<th>Protokoll</th>
<th>Nutzdaten</th>
<th>MAC-Acknowledgement</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>= 1460 Byte</td>
<td>= 34 Byte</td>
</tr>
<tr>
<td>IP</td>
<td>= 1480 Byte</td>
<td>= 272 Bit</td>
</tr>
<tr>
<td>SNAP</td>
<td>= 1500 Byte</td>
<td>= 294 Bit</td>
</tr>
<tr>
<td>LLC</td>
<td>= 1505 Byte</td>
<td>= 432 Bit</td>
</tr>
<tr>
<td>MAC</td>
<td>= 1508 Byte</td>
<td>= 0,232 ms</td>
</tr>
<tr>
<td>OFDM (54 MBit/s)</td>
<td>+ 16 Bit Service Field</td>
<td>+ 16 Bit Service Field</td>
</tr>
<tr>
<td></td>
<td>= 12336 Bit</td>
<td>= 12358 Bit</td>
</tr>
<tr>
<td></td>
<td>+ 6 Tail Bits</td>
<td>+ 6 Tail Bits</td>
</tr>
<tr>
<td></td>
<td>= 12358 Bit</td>
<td>= 294 Bit</td>
</tr>
<tr>
<td></td>
<td>+ 170 Pad Bits</td>
<td>+ 138 Pad Bit</td>
</tr>
<tr>
<td></td>
<td>= 12528 Bit</td>
<td>= 432 Bit</td>
</tr>
<tr>
<td>Übertragungsdauer:</td>
<td>0,232 ms</td>
<td>Übertragungsdauer: 0,048 ms</td>
</tr>
<tr>
<td>+ 4 µs PLCP Präambel</td>
<td>+ 16 µs PLCP Präambel</td>
<td></td>
</tr>
<tr>
<td>= 0,252 ms</td>
<td>= 0,068 ms</td>
<td></td>
</tr>
</tbody>
</table>

Zur Übertragung des MAC-Acknowledgement wurden 9 MBit/s als Basisdatenrate angenommen.

Damit ergibt sich unter Einbeziehung eines SIFS und eines DIFS als Dauer für den gesamten Übertragungsvorgang

\[ t = t_{DATA} + t_{SIFS} + t_{ACK} + t_{DIFS} = 252 \mu s + 16 \mu s + 68 \mu s + 34 \mu s = 370 \mu s \]

Die zur Übertragung der Nutzdaten benötigte Zeit beträgt

\[ t_{nutz} = \frac{1460 \cdot 8 \text{Bit}}{54 \text{MBit/s}} = 216,3 \mu s. \]

Der Overhead ergibt sich demnach für das Beispiel zu

\[ o = \frac{t_{nutz}}{t} = 41,54\% \]

Da die Protokolloverheads in Abhängigkeit von den verwendeten Netzwerkprotokollen, der betrachteten Netzschicht und der durchschnittlichen Größe der übertragenen Datenpakete sehr un-
Die Bitfehlerrate von WLANs - $10^{-3}$ bis $10^{-5}$ - erreicht noch nicht die niedrigen Werte drahtgebundener Netze. Aufgrund der zusätzlichen Störquellen, die das Funkmedium mit sich bringt, werden beim WLAN auf MAC-Ebene Korrekturmechanismen realisiert, die eine fehlerfreie Übertragung gewährleisten. Diese Mechanismen brauchen jedoch Zeit, die zum Ablaufen von TCP-Timern führen kann. TCP reagiert - in diesem Fall unnötig - mit einer Verkleinerung seines Sendefensters und der Durchsatz sinkt.


Ein weiterer Effekt tritt auf, wenn kontinuierliche Datenströme über eine WLAN-Strecke über-

Deshalb wird im Modell nur ein konstanter Faktor vorgesehen, der die durchschnittliche Reduzierung des Durchsatzes durch die Wechselwirkungen zwischen dem TCP-Protokoll und dem WLAN-MAC repräsentieren soll. Ein realistischer Wert für diesen Faktor ist durch Messungen zu bestimmen, die im Rahmen dieser Arbeit nicht erfolgen können.

4.6 Bedienrate

Die Bedienrate setzt sich aus den Teilausdrücken zusammen, die die in den vorangegangenen Abschnitten besprochenen Einflußfaktoren repräsentieren. Der Gesamtausdruck ergibt sich wie folgt:

\[
\mu = \frac{r \cdot (1 - v_p) \cdot (1 - v_c(n)) \cdot (1 - v_{tcp}) \cdot (1 - k_m) \cdot (1 - k_g \cdot n) + k_a}{g_p}
\]

wobei

\[
\begin{align*}
 r & \quad \text{Brutto-Datenrate des Kanals} \\
 v_p & \quad \text{Verluste durch Protokolloverhead} \\
 v_c(n) & \quad \text{Verluste durch Wettbewerb um den Kanal} (n=Anzahl der konkurrierenden Stationen) \\
 v_{tcp} & \quad \text{Verluste durch Wechselwirkungen zwischen TCP und dem WLAN-MAC} \\
 g_p & \quad \text{durchschnittliche Paketgröße} \\
 k_m & \quad \text{Freiheitsgrade zur Korrektur des Ergebnisses} \\
 k_g & \quad \text{multiplikativer Korrekturwert zur relativen Veränderung der Bedienrate} \\
 k_a & \quad \text{beeinflußt die Bedienrate ebenfalls relativ, jedoch ist das Ausmaß der Veränderung proportional zur Anzahl der Stationen im jeweiligen BSS} \\
 k_a & \quad \text{konstanter Wert}
\end{align*}
\]

Es wurden einige Freiheitsgrade in das Modell integriert, die eine Beeinflussung des Ergebnisses erlauben. Zum einen können so weitere Einflußfaktoren auf die Datenrate berücksichtigt werden, wie z.B. Overheads höherer Protokollschichten. Außerdem ist eine Korrektur der Werte des Modells möglich, sollten diese einem Vergleich mit der Realität (Meßergebnisse) nicht standhalten.

\( k_m \) ist ein multiplikativer Korrekturwert zur relativen Veränderung der Bedienrate.

\( k_g \) beeinflußt die Bedienrate ebenfalls relativ, jedoch ist das Ausmaß der Veränderung proportional zur Anzahl der Stationen im jeweiligen BSS.

\( k_a \) wird der ermittelten Bedienrate hinzuaddiert und verändert diese somit um einen konstanten Wert.
4.7 Auslastung

Eine Überlastung des modellierten Systems wird im Modell in zwei Formen dargestellt. Bei einem Wert von \( \rho \geq 1 \) liegt auf jeden Fall eine Überlastung vor. Bei \( \rho \leq 1 \) kann auch eine Überlastung vorliegen. Sie wird durch das Vorhandensein einer bestimmten durchschnittlichen Warteschlangenlänge angezeigt. Für eine Quantifizierung ist ein Vergleich der Modellergebnisse mit Messwerten und Simulationsergebnissen sowie eine Kallibrierung des Modells erforderlich.
Kapitel 5
Das Programm „Kapazitätsanalyse“

Das in Kapitel 4 entwickelte Modell wird durch das Programm „Kapazitätsanalyse“ umgesetzt. Das Programm ist in der Lage, einfache Netzwerke bezüglich der Lastsituation zu analysieren.


5.1 Benutzung des Programmes

5.1.1 Systemvoraussetzungen

Die Software sollte auf jedem handelsüblichen PC lauffähig sein. Voraussetzung ist lediglich ein funktionsfähig installiertes J2SE Runtime Environment 5.0 und eine von Java unterstützte grafische Benutzeroberfläche.

5.1.2 Programmaufruf

Das Vorgehen zum Start des Programmes „Kapazitätsanalyse“ kann sich je nach Betriebssystem unterscheiden. Auf allen von Java unterstützten Plattformen kann das Programm, sofern der Java-Interpreter im Pfad eingetragen ist, durch die Kommandozeile

    java -jar Kapazitaetsanalyse.jar

aufgerufen werden. Unter Windows (ab Win98) wird java.exe bei der Installation normalerweise mit Jar-Dateien verknüpft, so daß das Programm durch Doppelklick auf das Dateisymbol Kapazitaetsanalyse.jar im Explorer oder einer Laufwerksansicht gestartet werden kann.

5.1.3 Funktionen der Programmoberfläche

Nach dem Start des Programmes erscheint das in Abbildung 5.1 dargestellte Hauptfenster. Darin enthalten sind drei Karteikarten, die einzelne Funktionsbereiche der Oberfläche zugänglich machen.
Kap. 5 Das Programm „Kapazitätsanalyse“

Abbildung 5.1: Hauptfenster des Programmes „Kapazitätsanalyse“
5.1 Benutzung des Programmes

Abbildung 5.2: Die Karteikarte „Optionen“


5.1.4 Karteikarte „Optionen“

Die Karteikarte „Optionen“ (Abbildung 5.2) erlaubt es, verschiedene Einstellungen des WLAN-Modells zu verändern. Die meisten Optionen betreffen die Berechnung der Protokolloverheads.

Im Bereich „Betrachtete Protokollschicht“ kann eingestellt werden, auf welche Schicht des Protokollstacks die Lastangaben der NDML-Datei bezogen werden sollen und welches Protokoll genutzt werden soll. Zur Wahl stehen die Schichten eins bis vier des ISO/OSI-Modells. Auf Schicht eins und zwei werden bis jetzt nur die Protokolle der WLAN-Standards unterstützt. Hier ist keine Auswahl einzelner Protokolle möglich, da das zu nutzende Protokoll durch die in der Topologie-

In den darunter befindlichen „TCP-Einstellungen“ kann die Acknowledgement-Rate festgelegt werden. Sie bestimmt, aller wieviel TCP-Segmente (im Durchschnitt) ein TCP-Acknowledgement angenommen wird. Ein Wert von 0,5 bedeutet, daß jedem zweiten Segment ein ACK folgt. Ein Wert von 1 bedeutet, daß jedes Segment einzeln bestätigt wird.


Im Bereich „DSSS-Einstellungen“ kann festgelegt werden, ob lange Header oder das in 802.11b definierte kurze Headerformat genutzt werden sollen.

Für das OFDM-Verfahren kann die bei 802.11g zu nutzende Protection-Methode eingestellt werden. Der Protection-Mechanismus wird automatisch verwendet, wenn am betrachteten 802.11g-AP mindestens ein 802.11b- oder 802.11-Client angemeldet ist.

Der ganz unten befindliche Bereich „Allgemeine Einstellungen“ faßt einige Optionen zusammen, die in keine der bisherigen Kategorien paßten. Zum einen ist das die anzunehmende SDU-Größe. Diese wird auf das oben eingestellte Protokoll bezogen.

Die drei Werte darunter dienen der Manipulation der im Modell ermittelten Bedienraten. Die folgende Gleichung gibt an, wie die drei Werte die Bedienrate verändern:

\[ \mu = \mu \cdot (1 - k_m) \cdot (1 - n \cdot k_g) + k_a \]

Die ersten beiden Werte wirken also multiplikativ, wobei der zweite abhängig von der Zahl der Teilnehmer ist, die über den betrachteten Funkkanal kommunizieren. Der letzte Wert wird dem Ergebnis hinzuaddiert.

5.1 Benutzung des Programmes

5.1.5 Karteikarte „Modellmanipulation“


Die Funktionen dieser Karteikarte sind erst aufrufbar, nachdem für das aktuelle Szenario bereits ein Analyselauf durchgeführt wurde.

Im oberen Bereich der Arbeitsfläche listet eine Tabelle alle Access Points und die zu nutzende Technologie auf. Voreinstellung ist „gemäß NDML“. Ein Klick auf die Einträge der rechten Spalte bringt eine Auswahlliste zur Anzeige, mit der die Einstellung geändert werden kann (siehe Abbildung 5.3: Die Karteikarte „Modellmanipulation“)
Kap. 5 Das Programm „Kapazitätsanalyse“

Abbildung 5.4: Ändern der Technologie-Einstellung

Abbildung 5.4). Die unter der Tabelle befindliche Schaltfläche setzt alle Einträge auf den Standardwert zurück.

Die Tabelle im unteren Bereich der Arbeitsfläche zeigt für alle Access Points die voreingestellten WLAN-Technologien. Mögliche Werte sind 802.11, 802.11b, 802.11a, 802.11g_only und 802.11g_mixed. 802.11g_only bezeichnet ein BSS ausschließlich aus 802.11g-Geräten, während 802.11g_mixed besagt, daß an dem 802.11g Access Point mindestens ein 802.11- oder 802.11b-Gerät angemeldet ist.

5.1.6 Laden eines Projektes


5.1.7 Durchführen der Analyse

Der Button „Analyse starten“ der Karteikarte „Analyse“ setzt die Auswertung der gewählten NDML-Dateien und die Durchführung der Lastanalyse in Gang. Im Verlauf der Analyse werden im Ausgabebereich Informationen zu den Modelleinstellungen und schließlich die Ergebnisse zur Anzeige gebracht.
5.1 Benutzung des Programmes

Abbildung 5.5: Auswahl der genutzten Datenrate


5.1.8 Ausgabe der Analyseergebnisse


Verwendete Einstellungen
========================

<table>
<thead>
<tr>
<th>Basic-Datei</th>
<th>/szenarios/szenario2/szenario2_basic.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load-Datei</td>
<td>/szenarios/szenario2/szenario2_load.xml</td>
</tr>
<tr>
<td>Topology-Datei</td>
<td>/szenarios/szenario2/szenario2_topology.xml</td>
</tr>
<tr>
<td>Allgemein</td>
<td>SDU-Größe . . . . . . . . . . : 1000</td>
</tr>
<tr>
<td></td>
<td>Betrachtete Schicht/Protokoll . : DATALINK/MACWLAN</td>
</tr>
<tr>
<td>QNAccessPoint</td>
<td>Multiplikative Korrektur . . . : 0.0</td>
</tr>
<tr>
<td></td>
<td>Protokoll-Interaktions-Verlust . : 0.1</td>
</tr>
<tr>
<td></td>
<td>Additive Korrektur . . . . : 0.0</td>
</tr>
</tbody>
</table>
Ergebnisse der Analyse

Projektname: Beispiel-Projekt
Projektbeschreibung:

Beispiel-Projekt: Beispiel-Projekt

** QNNode Nr. 5 **

Repr. Gerät . . . . : AccessPoint 1
 Ankunftsrate . . . . : 2523.148148148148
 Bedienrate . . . . . : 3703.703703703704
 Auslastung . . . . . : 0.6812499999999999
durchschn. WS-Länge : 1.4560049019607837

Verbindungen (Datenmenge):
{ QNDestination () = 2500.0, QNNode(6,7) = 23.148148148148145 }

** QNNode Nr. 6 **

Repr. Gerät . . . . : Switch 1
 Ankunftsrate . . . . : 1273.148148148148
 Bedienrate . . . . . : 1273.148148148148
 Auslastung . . . . . : 0.1
durchschn. WS-Länge : 0.011111111111111113

Verbindungen (Datenmenge):
{ QNDestination () = 23.148148148148145, QNNode(5,6) = 1250.0 }

** QNSource () **

Verbindungen:
{ QNNode(6,7) = 1250.0, QNNode(5,6) = 1273.148148148148 }
5.2 Interne Funktionsweise

5.2.1 Repräsentation eines Bedienungsnetzes


Zur Repräsentation eines Bedienungsnetzes dient die Klasse QueueingNet. Sie ist Container für die Elemente des Bedienungsnetzes und stellt auch die Methoden bereit, die ein Bedienungs-
netz aus der NDML-Darstellung eines Rechnernetzes generieren und das Netz analysieren.

```java
QueueingNet(String basicFileName, String loadFileName,
        String topologyFileName);
void buildQueueingNet();
void setLayer(Overheads.Layer layer, Overheads.Protocol protocol);
void setPacketSize(int size);
void queueingAnalysis();
```


Die Methode `queueingAnalysis()` schließlich berechnet die Leistungsgrößen des Bedienungsnetzes mit Hilfe des Jackson-Theorems.


Die Klasse `Tools.DijkstraRouting` wird benötigt, um aus den Angaben von Start- und Zielpunkt im Load-Viewpoint die Wege zu ermitteln, die die Forderungsströme im Bedienungsnetz nehmen müssen. Wie der Name vermuten lässt, kommt dabei der Dijkstra-Algorithmus zum Einsatz.

Die Klasse `Model` ist als Modell im Sinne des Model-View-Controller-Schemas aus der Softwaretechnologie zu sehen. Sie kapselt die relevanten Schnittstellen der anderen beteiligten Klassen. Sie ist nur zur Verwendung in Zusammenhang mit den Oberflächenklassen bestimmt.
5.2 Aufbau des Bedienungsnetzes


Der Aufbau eines Bedienungsnetzes wird durch die Methode buildQueueingNet() der Klasse QueueingNet realisiert und gliedert sich in folgende Schritte:

1. Auslesen der relevanten NDML-Dateien
2. Auswertung des Viewpoints Basic
3. Erzeugung der QNNodes
4. Routing-Tabelle aufbauen
5. Verbindungen im Bedienungsnetz erzeugen
6. QNNodes parametrisieren
7. überflüssige QNNodes entfernen, Quelle und Senke integrieren

Im ersten Schritt werden die drei hier relevanten NDML Projekdateien - Viewpoints Basic, Load und Topology - mittels JAXB (siehe Abschnitt 5.3.1) ausgelesen. Die enthaltenen Informationen liegen anschließend programmintern als Objekthierarchie vor, aus der in den folgenden Schritten die für die Lastanalyse relevanten Informationen gewonnen werden.

Der nächste Schritt ist die Auswertung des Viewpoints Basic. Hier werden nur Name und Beschreibung des Projektes ausgelesen.

Im dritten Schritt beginnt der eigentliche Aufbau des Bedienungsnetzes. Aus dem Topology-Viewpoint werden durch die Methode QueueingNet.createQNNodes() die Devices des Rechnernetzes ausgelesen und für jedes ein Objekt der passenden QNNode-Unterklasse erzeugt:

```
// für jedes Device d
switch (d.getCategory()) {
    case SWITCH:
        q = new QNSwitch(d, this); break;
    case ACCESSPOINT:
        q = new QNAccessPoint(d, this); break;
    default:
```

Im folgenden Schritt wird eine Routingtabelle berechnet, welche die Struktur des Netzwerks für das Bedienungsnetz abbildet. Sie enthält für jedes Paar Knoten die Zwischenstationen, über die die zugehörigen Devices miteinander verbunden sind. Zur Erzeugung dieser Tabelle wird der Dijkstra-Algorithmus verwendet.


Im sechsten Schritt werden die einzelnen Knoten des Modells parametriert. Diese Anpassungen sind Device-spezifisch und finden daher in den parametrise() Methoden der jeweiligen Unterklassen von QNNode statt. Diese werden von der Methode QueueingNet.parametriseNodes() aufgerufen:

```java
for (QNNode node : qNNodes) {
    if (!node.isTemporary()&&!node.isLocked())
        switch (node.getDevice().getCategory()) {
            case ACCESSPOINT: ((QNAccessPoint)n).parametrise(topology); break;
            case SWITCH: ((QNSwitch)n).parametrise(topology); break;
            default: n.parametrise(topology);
        }
}
```

5.2 Interne Funktionsweise

5.2.3 Parametrisierung der Knoten im Detail

Die Parametrisierung der Knoten bzw. die Verfeinerung des Modells findet in den `parametris()` Methoden der Unterklassen von QNNode statt, nachdem in den vorherigen Schritten nur ein Knoten pro Device des Netzwerks erzeugt wurde und diese miteinander entsprechend der Netzstruktur verbunden wurden. Im einfachsten Fall wird die Bedienrate des Knotens bestimmt. Es kann jedoch auch erforderlich sein, ein Netzwerkgerät detaillierter zu modellieren und durch mehr als eine Bedienstation darzustellen. In diesem Fall sind in der `parametris()` Methode die erforderlichen zusätzlichen Knoten zu erzeugen und zu parametrisieren, geeignet miteinander sowie mit dem bestehenden Netz zu verbinden und dem QueueingNet hinzuzufügen. Letzteres geschieht durch zwei Methoden von QueueingNet:

```java
void addQNNode(QNNode node);
void addQNNodes(QNNode[] nodes);
```

Die erste fügt dem Netz einen einzelnen Knoten hinzu, die zweite mehrere Knoten, die als Array übergeben werden. Die Knoten müssen dem QueueingNet in Reihenfolge ihrer Erzeugung zugefügt werden. Das heißt bezogen auf die zweite Methode, daß der zuerst erzeugte Knoten im Array an Index 0 steht, der danach erzeugte an Index 1, usw.

Um zu verhindern, daß für die in diesem Schritt manuell erzeugten und parametrisierten Knoten eventuell wieder die `parametris()`-Methode aufgerufen wird, ist die Methode `setLocked(true)` aufzurufen. Damit wird erreicht, daß `parametiseNodes()` den betreffenden Knoten übergeht.

Werden hier keine neuen Knoten erzeugt ist dies nicht erforderlich, da `parametiseNodes()` keinen Knoten mehrfach betrachtet.
5.2.4 Ermittlung der Protokoll-Overheads

Das Paket `LastAnalyse.Overheads` dient zur Ermittlung der durch den Protokollstack entstehenden Overheads. In der jetzigen Fassung wird die Bitübertragungs- und Sicherungsschicht von 802.11, 802.11b, 802.11a und 802.11g modelliert. Als Protokoll der Vermittlungsschicht ist IP vertreten. An Transportprotokollen stehen TCP und UDP zur Verfügung.

Das in Abbildung 5.8 dargestellte Klassendiagramm gibt eine Übersicht über die Architektur des Paketes. Zentrale Schnittstelle ist die Klasse `Overheads`: 
Die Methode `computeOverhead(...)` ermittelt die Protokolloverheads für die gegebene WLAN-Technologie, Netzwerkschicht, Protokoll, Datenrate und SDU-Größe. `computeAverageOverhead(...)` ermittelt den durchschnittlichen Overhead für eine Menge von Datenraten, die übrigen Parameter sind die gleichen. Mit den `get-`Methoden kann auf die Instanz der jeweiligen Protokoll-Repräsentation zugegriffen werden, um Einstellungen zu verändern.


Die Klasse `Mac` modelliert Overheads der gesamten WLAN-Sicherungsschicht, inklusive LLC und SNAP.

Jede Protokoll-Repräsentation hält eine Referenz auf eine Instanz der darunter liegenden Schicht. In `Layer` ist die Methode `double getTransmissionTime(int pkgSize, double rate)` definiert, die in den Unterklassen überschrieben wird. Diese ermittelt die Zeit, die zur Übertragung einer SDU der angegebenen Größe mit der angegebenen Datenrate benötigt wird, indem sie Overheads des repräsentierten Protokolls berücksichtigt und die gleichnamige Methode der Instanz aufruft, die die darunterliegende Protokollschicht repräsentiert. Als Beispiel soll die Methode der Klasse `IP` dienen:

```java
public double getTransmissionTime(int pkgSize, double rate)
{
    // 20 Byte IP Header
    return mac.getTransmissionTime(pkgSize+20, rate);
}
```

Beim TCP-Modell kann mittels `setAcknowledgementRate(double)` festgelegt werden, aller wieviel TCP-Segmente durchschnittlich ein Acknowledgement gesendet wird. Beispielswei-
se bedeutet ein Wert von 1, daß jedes Segment bestätigt wird. Bei einem Wert von 0,5 erfolgt für jedes zweite Segment ein Acknowledgement.

5.2.5 Durchführung der Analyse


Im zweiten Schritt erfolgt keine explizite Überprüfung der Stabilitätsbedingung $\rho < 1$. Eine Verletzung dieser Bedingung ist bei der Auswertung der Analyseergebnisse zu berücksichtigen. Ist für eine Bedienstation $\rho \geq 1$, also die Ankunftsrate größer als die Bedienrate, so ist diese Station überlastet. Die für die anderen Stationen ermittelten Werte sind dann als falsch zu betrachten, da die zugrundegelegten Datenströme aufgrund der Überlastsituation so nicht zustande kommen. Sind mehrere Stationen überlastet, kann diejenige mit der größten Auslastung als primärer Engpaß betrachtet werden.

Lediglich wenn die Auslastung aller Stationen kleiner als eins ist, können die ermittelten Leistungskenngrößen als korrekt betrachtet werden. Diese Einschränkung ist akzeptabel, da sie dem primären Zweck des Programmes, Engpässe und Überlastungen zu erkennen, nicht zuwider läuft.

Zur Ermittlung der Leistungsgrößen ruft die Methode `QueueingNet.queueingAnalysis()` `QNNode.compute()` für jeden Knoten des Netzes auf. Dort werden die Auslastung und die durchschnittliche Warteschlangenlänge ermittelt.

Als Zeichen einer Überlastung ist bereits das Entstehen einer Warteschlange zu werten, auch wenn $\rho$ noch kleiner als eins ist.

5.2.6 Einschränkungen des Programmes

Das Programm „Kapazitätsanalyse“ ist als Prototyp zu sehen. Es wurden einige vereinfachende Annahmen gemacht und Einschränkungen in Kauf genommen, speziell was die Auswertung der NDML-Dateien und die modellierbaren Szenarien betrifft. Diese werden im folgenden aufgelistet:


Beim Einsatz der NDML3 in CANDY werden diese Datenbestände sinnvoller Weise durch eine Datenbank bereitgestellt, nicht durch einzelne XML-Dateien.

an zwei Stellen war eine Umbenennung im NDML-Schema erforderlich, da keine korrekte Abbildung mit JAXB gelang (siehe Abschnitt 5.3.1). Im Viewpoint Load mussten die Typen Device und Devices in LDevice und LDevices umbenannt werden, um einen Konflikt mit gleichnamigen Typen im Topology-Viewpoint aufzulösen.

Einige weitere Besonderheiten bestehen für den Load-Viewpoint. Diese sind in den Abschnitten 5.4.1 und 5.4.2 näher erläutert.

5.3 Genutzte Technologien

Zur Entwicklung des Programmes „Kapazitätsanalyse“ wurde folgende Software genutzt:

- Java SDK 5.0
- JAXB 2.0ea3 zur XML-Verarbeitung
- Gnu Emacs mit JDE als Entwicklungsumgebung
- diverse nützliche Unix-Werkzeuge

5.3.1 Java Architecture for XML Binding (JAXB)


Abbildung 5.9: JAXB-Architektur (Quelle: SUN (java.sun.com))


```xml
<schemaBindings>
  <nameXmlTransform>
    <typeName prefix="..."/>
  </nameXmlTransform>
</schemaBindings>
```

Der xjc-Compiler-bricht dennoch mit Verweis auf die Namenskonflikte ab. Wird eines der konfliktierenden Schemata weggelassen, findet die beabsichtigte Umbenennung korrekt statt. Vermutlich ist das Problem auf einen Bug in JAXB zurückzuführen. Da deshalb davon ausgegangen werden kann, daß es sich um ein temporäres Problem handelt, wurde an der Verwendung von JAXB festgehalten und ein Weg zur zwischenzeitlichen Umgehung des Problems gesucht. Dazu wurden die betroffenen Typen im Load-Schema umbenannt, indem den Namen ein „L“ vorangestellt wur-
Für einen Prototypen scheint diese Maßnahme vertretbar, da die Umbenennung auch leicht in einem Vorverarbeitungsschritt automatisch durchgeführt werden kann.

5.4 Integration in CANDY


5.4.1 Anforderungen an NDML

Die Sprache NDML muß bestimmte Informationen widergeben können, damit sie einem Kapazitätsanalyse-Werkzeug zur Bereitstellung der erforderlichen Eingabedaten dienen kann. Im folgenden soll NDML 3.0 anhand des bereits in der Einleitung genutzten Beispiels (Abbildung 5.10) dahingehend betrachtet werden.

Zur vollständigen Unterstützung des Analyseprozesses müssen alle in der Abbildung dargestellten Informationen bzw. zu deren Erlangung notwendige Vorinformationen in NDML abbildbar sein:

- Topologiebeschreibung
- Gebäude-/Geometriebeschreibung
- Signalstärken
- erweiterte Interface-Informationen zu WLAN-Hardware
- im WLAN genutzte Datenraten
• Lastangaben

Mit der Topologiebeschreibung gibt es keine Probleme. Der in NDML 3.0 enthaltene Topology-Viewpoint ist mächtig und flexibel genug, um verschiedenste Netzwerk-Konstellationen zu beschreiben. Auch die Beschreibung von WLAN-Teilnetzen ist durch die Medien-Kategorie *air* vorgesehen:

```
<medium id="5">
  <description>Verbindung von Notebook 1 mit Router</description>
  <category>air</category>
  <plugs>
    <plugType>wireless</plugType>
    <plugType>wireless</plugType>
  </plugs>
  ...
</medium>
```

Für eine WLAN-Kapazitätsanalyse ist es notwendig, die tatsächlich nutzbaren Datenraten auf den einzelnen Funkstrecken zu kennen. Diese ergeben sich aus der vorhandenen Signalstärke und der hardwarespezifischen Zuordnung, welches Modulationsverfahren bei welcher Signalstärke genutzt wird.


Ein Repository zur Beschreibung der im Netz eingesetzten Netzwerkprodukte ist in NDML 3.0 ebenfalls vorhanden. Zur Speicherung der Signalstärke-Schwellwerte für WLAN-Hardware ist allerdings bisher keine Möglichkeit vorgesehen.

Zur Lastdefinition sieht NDML 3.0 eine hierarchische Darstellung der Komponenten des Rechnernetzes vor. Zunächst werden die zu berücksichtigenen PCs, Notebooks etc. als *Devices* definiert. Dann werden *DeviceSets* definiert, wobei ein DeviceSet aus mindestens einem Device besteht. Die auftretenden Lasten selbst werden als *Connections* definiert:

```
<connection id="1">
  <description>Verbindung zwischen Mitarbeitern und dem Server</description>
  <connects>
    <deviceSet inDocument="this">1</deviceSet>
    <deviceSet inDocument="this">2</deviceSet>
  </connects>
  <type>wired</type>
  <traffic unit="GByte" per="day">2</traffic>
  <trafficTypes>
    <trafficType>http</trafficType>
  </trafficTypes>
</connection>
```
Dabei sind ausschließlich Verbindungen zwischen DeviceSets vorgesehen. Für jede Connection wird das pro Zeiteinheit auftretende Datenvolumen, der Traffic Type sowie der Typ der Verbindung (wired oder wireless) festgelegt.


5.4.2 Umgehung der Probleme mit NDML 3.0

Zur Umgehung der in Abschnitt 5.4.1 geschilderten Probleme mit NDML 3.0 wurden einige Konventionen eingeführt, die nicht im Einklang mit den Absichten stehen, die bei der Entwicklung von NDML 3.0 zugrunde gelegt wurden, die jedoch für die den hier verfolgten Zweck erforderlich waren.


Da für die Lastanalyse Datenlasten zwischen konkreten Rechnern - und nicht Gruppen von Rechnern - berücksichtigt werden sollen, besteht im Load-Viewpoint jedes DeviceSet aus genau einem

Das type-Attribut von Connection im Load-Viewpoint wird ignoriert, da Ende-zu-Ende-Verbindungen ungeachtet der für die Verbindung genutzten Netzwerktechnologien spezifiziert werden sollen.

Da NDML bisher keine Möglichkeit zur Speicherung der bei einer WLAN-Verbindung tatsächlich genutzten Datenrate vorsieht, wird diese Information im Programm für jede WLAN-Strecke im Dialog vom Anwender erfragt.

### 5.4.3 Graphische Lastdarstellung

Kapitel 6
Leistungsbetrachtungen zu WLAN in der Literatur

Im Rahmen dieser Arbeit wurde nach vorhandenen Arbeiten aus dem Themenbereich Kapazitätsanalysen für Wireless LANs recherchiert.


In Form von wissenschaftlichen Papieren sind vielfältige Betrachtungen zur WLAN-Leistungsanalyse zu finden. Die meisten dieser Arbeiten lassen sich in eine der folgenden Kategorien einordnen:

- Untersuchung verschiedener MAC-Varianten im Vorfeld der Standardisierung von 802.11 WLAN
- Performance-Untersuchungen zur 802.11 MAC-Schicht
- Vorschläge zur Verbesserung der Performance der 802.11 MAC-Schicht

Eine dieser Arbeiten soll im Folgenden genauer vorgestellt werden, anschließend wird ein Überblick weiterer Arbeiten gegeben.

6.1 G. Bianchi: „Performance Analysis of the IEEE 802.11 Distributed Coordination Function“

In dieser Arbeit ([17]) wird ein analytisches Modell für die 802.11 DCF entwickelt, mit dessen Hilfe Aussagen über den Durchsatz von WLANs sowohl für den einfachen Zugriffsmechanismus als auch für das RTS/CTS-Verfahren gewonnen werden. Der Kern des erstellten Modells soll hier vorgestellt werden, da der Ansatz interessant und elegant ist und eine Vielzahl anderer Arbeiten darauf aufbauen.

Für das Modell werden einige vereinfachende Annahmen gemacht:

- idealer Kanal
- feste Anzahl Stationen
- Sättigungsannahme; Damit geht jeder Übertragung eine Backoffphase voraus.
- die Kollisionswahrscheinlichkeit beträgt für jede Übertragung $p \equiv const$, egal wie viele Kollisionen vorausgegangen sind
Abbildung 6.1: Markov-Kette für Backoff-Prozeß (Quelle: [17])

Zunächst wird ein Ausdruck für die Wahrscheinlichkeit entwickelt, daß eine Station in einer willkürlich gewählten Zeitscheibe eine Übertragung startet. Dazu wird der Backoff-Mechanismus als Markovkette modelliert. Dies ist aufgrund der Annahme konstanter Kollisionswahrscheinlichkeiten möglich. Es werden zwei stochastische Prozesse betrachtet:

- \( b(t) \) sei der Backoff-Counter der betrachteten Station zum Zeitpunkt \( t \)
- \( s(t) \) sei die Backoff-Ebene der betrachteten Station zum Zeitpunkt \( t \)

Der zweidimensionale Prozess \( \{s(t), b(t)\} \) wird als Markov-Kette modelliert (Abbildung 6.1). Im Folgenden sind die Zustandsübergangswahrscheinlichkeiten zusammengefaßt:

\[
\begin{align*}
P\{i, k | i, k + 1\} &= 1 \quad k \in (0, W_i - 1) \quad i \in (0, m) \\
P\{0, k | i, 0\} &= (1 - p) / W_0 \quad k \in (0, W_0 - 1) \quad i \in (0, m) \\
P\{i, k | i - 1, 0\} &= p / W_i \quad k \in (0, W_i - 1) \quad i \in (1, m) \\
P\{m, k | m, 0\} &= p / W_m \quad k \in (0, W_m - 1)
\end{align*}
\]

(6.1) \( \quad \) (6.2) \( \quad \) (6.3) \( \quad \) (6.4)

Gleichung 6.1 trägt der Tatsache Rechnung, daß zu Beginn jeder Zeitscheibe der Backoff-Counter um eins heruntergezählt wird. Gleichung 6.2 behandelt den Fall einer erfolgreichen Übertragung.
Die Backoff-Ebene wird zurückgesetzt und der Backoff-Counter aus dem Bereich \((0, W_0 - 1)\) gewählt. Die folgende Gleichung spiegelt die Erhöhung der Backoff-Ebene nach einer kollidierten/gestörten Übertragung wider. War die ursprüngliche Backoff-Ebene \(i - 1\), wird nun der Backoff-Counter aus dem Bereich \((0, W_i)\) gewählt. Gleichung 6.4 spiegelt den Fall wider, daß eine Übertragung scheitert, wenn schon die maximale Backoff-Ebene erreicht war.

Mit \(b_{i,k} = \lim_{t \to \infty} P\{s(t) = i, b(t) = k\}, i \in (0, m), k \in (0, W_i - 1)\) ist die stationäre Verteilung der Kette gegeben. Es werden alle \(b_{i,k}\) auf \(b_{0,0}\) zurückgeführt und \(b_{0,0}\) durch die Normalisierungsbedingung

\[
1 = \sum_{i=0}^{m} W_{i-1} \sum_{k=0}^{b_{i,k}}
\]

bestimmt. Im folgenden sollen nur die Ergebnisse dargestellt werden:

\[
b_{i,k} = \frac{W_i - k}{W_i} p^i b_{0,0},
b_{0,0} = \frac{2(1 - 2p)(1 - p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^m)}
\]

Davon ausgehend wird die Wahrscheinlichkeit \(\tau\), daß eine betrachtete Station in einer willkürlich gewählten Zeitscheibe sendet

\[
\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1 - p} = \frac{2(1 - 2p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^m)}
\]

sowie die Kollisionswahrscheinlichkeit \(p = 1 - (1 - \tau)^{n-1}\) bestimmt.

Als nächstes wird der normalisierte Systemdurchsatz \(S\) bestimmt, d. h. der Zeitanteil, den der Kanal zur Übertragung von Nutzdatenbits verwendet wird. Dazu wird betrachtet, was in einer willkürlich ausgewählten Zeitscheibe passieren kann. \(P_{tr}\) ist die Wahrscheinlichkeit, daß mindestens eine Station in der betrachteten Zeitscheibe sendet. Mit \(n\) konkurrierenden Stationen, wobei jede Station mit Wahrscheinlichkeit \(\tau\) sendet, ergibt sich

\[
P_{tr} = 1 - (1 - \tau)^n.
\]

\(P_s\) ist die Wahrscheinlichkeit, daß eine erfolgreiche Übertragung stattfindet, d. h. die Wahrscheinlichkeit, daß exakt eine Station sendet unter der Bedingung, daß mindestens eine Station sendet:

\[
P_s = \frac{n\tau(1 - \tau)^{n-1}}{P_{tr}} = \frac{n\tau(1 - \tau)^{n-1}}{1 - (1 - \tau)^n}.
\]

\(S\) wird nun ausgedrückt als

\[
S = \frac{E[\text{innerhalb einer Zeitscheibe übertragene Nutzdaten}]}{E[\text{Länge der Zeitscheibe}]}.
\]
$E[P]$ sei die durchschnittliche Nutzdatengröße pro Paket. Die durchschnittliche Nutzdatenmenge, die in einer Zeitscheibe übertragen wird, ist dann $P_{tr}P_{s}E[P]$, da eine erfolgreiche Übertragung mit Wahrscheinlichkeit $P_{tr}P_{s}$ stattfindet. Die durchschnittliche Länge einer Zeitscheibe wird bestimmt unter Betrachtung der drei Möglichkeiten, daß innerhalb einer Zeitscheibe

- mit Wahrscheinlichkeit $(1 - P_{tr})$ keine Übertragung
- mit Wahrscheinlichkeit $P_{tr}P_{s}$ eine erfolgreiche Übertragung
- mit Wahrscheinlichkeit $P_{tr}(1 - P_{s})$ eine Kollision

stattfinden können. Damit ergibt sich

$$S = \frac{P_{s}P_{tr}E[P]}{(1 - P_{tr})\sigma + P_{tr}P_{s}T_{s} + P_{tr}(1 - P_{s})T_{c}}.$$  

$\sigma$ ist die Slottime gemäß der genutzten Bitübertragungsschicht. $T_{s}$ ist die durchschnittliche Zeit, die der Kanal aufgrund einer erfolgreichen Übertragung belegt ist, $T_{c}$ die entsprechende Dauer während einer Kollision. $T_{s}$ und $T_{c}$ sind in Abhängigkeit vom genutzten Zugriffsverfahren zu bestimmen. Dabei wird im Folgenden von einer konstanten Nutzdatengröße pro Paket ausgegangen.

Zuerst wird der Fall des einfachen Zugriffsverfahrens betrachtet. Sei $H = PHY_{hdr} + MAC_{hdr}$ die Dauer zur Übertragung der Paket-Header und $\delta$ die bei der Ausbreitung von Funkwellen entstehende Verzögerung. Es ergibt sich

$$T_{s}^{bas} = H + E[P] + SIFS + \delta + ACK + DIFS + \delta$$
$$T_{c}^{bas} = H + E[P^*] + DIFS + \delta,$$

wobei $E[P^*]$ die durchschnittliche Länge des längsten in eine Kollision verwickelten Nutzdatenteiles ist.

Bei Einsatz des RTS/CTS-Verfahrens ergeben sich folgende Werte:

$$T_{s}^{rts} = RTS + SIFS + \delta + CTS + SIFS + \delta + H + E[P] + SIFS + \delta + ACK + DIFS + \delta$$
$$T_{c}^{rts} = RTS + DIFS + \delta.$$

In [17] werden noch die Fälle variabler Paketgrößen und Systeme im hybriden Modus betrachtet, wo RTS/CTS nur bei Überschreiten einer bestimmten Paketgröße eingesetzt wird. Auch dafür werden Ausdrücke für $T_{s}$ und $T_{c}$ bestimmt, auf deren Darstellung hier aufgrund ihrer Komplexität aber verzichtet wird.
6.2 Übersicht weiterer Papiere

O. Abu-Sharkh, A. H. Tewfik:
„Troughput Evaluation and Enhancement in 802.11 WLANs with Access Point“
(2005, IEEE 0-7803-8887-9/05)

Gegenstand: Auswirkungen der Nutzung verschiedener Datenraten innerhalb eines 802.11-BSS auf den Durchsatz.

Inhalt:
- Ausgangspunkt: Nutzung unterschiedlicher Datenraten innerhalb eines BSS reduziert den Durchsatz aller Stationen
- analytisches Modell zur Bestätigung des Problems
- Vorschlag einer Protokolländerung, um das Problem zu lösen: Framegröße wird Funktion der Übertragungsgeschwindigkeit
- Simulation des Effekts mit NS/2

G. Bianchi:
„Performance Analysis of the IEEE 802.11 Distributed Coordination Function“
(2000, IEEE 0733-8716/00)

Gegenstand: 802.11 MAC Protokoll, speziell DCF

Annahmen, Vereinf.
- Sättigung
- feste Anzahl Stationen
- idealer Kanal
- konstante Kollisionswahrscheinlichkeit

Ergebnisse
- analytisches Modell zur Bestimmung des Durchsatzes im Sättigungsfall
- Vorschlag einer Methode, für jede Netzkonstellation maximalen Durchsatz zu erreichen

Thesen
- maximal erreichbarer Durchsatz der einfachen Zugriffsmethode liegt sehr nahe bei dem bei RTS/CTS-Nutzung
- maximaler Durchsatz ist unabhängig von der Anzahl der Stationen im Netz
G. Bianchi, L. Frata, M. Oliveri:
„Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs“

Gegenstand  Performance der 802.11 DCF; Vorschlag einer Protokollerweiterung zur Verbesserung
Inhalt  • Simulative Untersuchung der Performanz der 802.11 DCF
• Vorschlag einer Protokollerweiterung: adaptive Anpassung des Backoff Fensters an die Anzahl der Stationen, die um den Kanal konkurrieren
• Berücksichtigung von RTS/CTS und versteckten Terminals
Ergebnisse  • Simulationsergebnisse für Sättigungsdurchsatz bei Variation von Paketgröße, Anzahl der Stationen und der Wahrscheinlichkeit von Hidden Terminals
• Simulationsergebnisse für die durchschnittliche Verzögerung beim Kanalzugriff in Abhängigkeit von der angebotenen Last
Thesen  • Saturation Throughput hängt stark vom Anfangswert für $CW_{\text{min}}$ und der Anzahl der Stationen ab
• optimaler Wert von $CW_{\text{min}}$ hängt vom Netzwerkszenario, d.h. der Anzahl Stationen im BSS ab. Bei wenigen Stationen ist ein kleiner Wert besser, wogegen bei vielen Stationen ein großer Wert bessere Ergebnisse liefert. (Dieser Tatsache wird in dem entwickelten adaptiven Algorithmus Rechnung getragen.)

Sheng-Tzong Cheng, Mingzoo Wu:
„Performance Evaluation of Ad-Hoc WLAN by M/G/1 Queueing Model“
(2005, IEEE 0-7695-2315-3/05)

Gegenstand  Analytische Bestimmung von Durchsatz und Antwortzeit für 802.11 Ad-Hoc-Netze
Annahmen, Vereinf.  • konstante Kollisionswahrscheinlichkeiten für alle Stationen, unabhängig von der Anzahl der Übertragungswiederholungen
• Varianz der MSDU-Größe $E[P] = 0$
• identische Ankunftsverteilung für alle Stationen
Modelleigenschaften

- Modellierung jeder Station als M/G/1-Bedienstation
- Modellparameter: Anzahl der Stationen, Ankunftsrate $\lambda$, durchschnittliche MSDU-Größe $P$

Ergebnisse

Analytische Ausdrücke für Durchsatz, Durchsatz bei Sättigung, Antwortzeit und Warteschlangenlänge

B. P. Crow, I. Widjaja, J. G. Kim, P. Sakei:
„Investigation of the IEEE 802.11 Medium Access Control (MAC) Sublayer Functions“
(1997, IEEE 0-8186-7780-5/97)

Gegenstand

Untersuchung der im Entwurf des 802.11 Standard spezifizierten MAC Teilschicht mittels diskreter Simulation.

Annahmen, Vereinf.

- Ausbreitungsverzögerung wird vernachlässigt
- „hidden terminal“-Problem wird nicht berücksichtigt
- ausschließliche Nutzung von DSSS mit 1 MBit/s
- bei keiner Station ist der Stromsparmodus aktiviert
- es tritt keine Interferenz mit anderen BSS auf
- eine identische Traffic-Verteilung für alle Stationen wird angenommen

Modelleigenschaften

- Berücksichtigung von DCF und PCF
- Modellierung von Ad-Hoc-Netz (DCF) und Infrastrukturnetz (DCF+PCF)
- asynchroner Datenverkehr und Packet Voice Anwendung

Ergebnisse

- Durchsatz in Abhängigkeit von Signalqualität, MSDU-Größe
- Untersuchung des optimalen Fragmentation Threshold
- Untersuchung des optimalen RTS/CTS-Threshold

Thesen

- Echtzeit-Dienste können mit der 802.11 PCF realisiert werden.
- Packet-Voice über WLAN erfordert Einsatz eines Echo Cancelers.
M. Heusse, F. Rousseau, G. Berger-Sabbatel A. Duda:
„Performance Anomaly of 802.11b“
(2003, IEEE 0-7803-7753-2/03)

Inhalt

- Performance Anomaly = Einbruch des Durchsatzes aller Stationen eines BSS, wenn mindestens eine Station mit einer niedrigeren Datenrate arbeitet.
- Analytisches Modell zur Bestätigung des Phänomens
- Bestätigung des Modells durch Simulation
- diverse Meßergebnisse zur Illustration des Effekts

J. V. Sudarev, L. B. White:
„Performance Analysis of 802.11 CSMA/CA for Infrastructure Networks under Finite Load Conditions“

Gegenstand

Analytische Untersuchung der Performanz der 802.11 DCF in Infrastruktur-Netzwerken bei endlicher Last

Annahmen, Vereinf.

- endliche Anzahl Stationen
- idealer Kanal
- keine Saturation-Annahme
- die Kollisionswahrscheinlichkeit, die Wahrscheinlichkeit für einen belegten Kanal und die Wahrscheinlichkeit für eine leere Queue sind unabhängig von Backoff-Ebene und dem Retry Counter der Stationen

Modelleigenschaften

- beliebige Paketankunftsrate (aber für alle STAs gleich)
- beliebige Paketlängenverteilung (aber für alle STAs gleich)
- jede Station wird als M/G/1-Queue modelliert
- Verifikation des Modells durch Vergleich mit NS2-Simulationsergebnissen

Ergebnisse

- geschlossener Ausdruck für normalisierten Systemdurchsatz (=Zeitanteil, der zur erfolgreichen Übertragung von Nutzdatenbits verwendet wird)
- Ergebnisse für Durchsatz in Abhängigkeit von der Anzahl der Stationen bei verschiedenen Ankunftsrate
Y. C. Tay, K. C. Chua:
„A Capacity Analysis for the IEEE 802.11 MAC Protocol“

Gegenstand
Analyse von Durchsatz und Kollisionswahrscheinlichkeit in 802.11 Ad-Hoc Netzwerken

Annahmen, Vereinf.
- DCF, RTS/CTS wird nicht verwendet
- Saturation-Annahme
- Kollisionswahrscheinlichkeit < 0,5

Ergebnisse
- geschlossene Ausdrücke für Kollisionswahrscheinlichkeit, maximalen Durchsatz und den Grenzwert der Anzahl Stationen pro BSS

Thesen
- Halbierung von $CW_{\min}$ hat gleichen Effekt auf maximalen Durchsatz wie eine Verdopplung der Anzahl der Stationen
- optimales $CW_{\min}$ ist proportional zur Quadratwurzel der durchschnittlichen Paketgröße

O. Tickoo, B. Sikdar:
„A Queueing Model for Finite Load IEEE 802.11 Random Access MAC“
(2004, IEEE 0-7803-8533-0/04)

Gegenstand
Analytische Untersuchung von Antwortzeit und Warteschlangenlängen bei Einsatz der DCF

Annahmen, Vereinf.
- feste Anzahl Knoten
- RTS/CTS wird genutzt
- beliebiger (aber für alle Knoten identischer) Paketankunftsprozeß
- beliebige Paketlängenverteilung

Modelleigenschaften
- jeder Knoten als G/G/1-Queue modelliert
- explizite Berücksichtigung des Einflusses endlicher Last auf die Kollisionswahrscheinlichkeit

Ergebnisse
- geschlossene Ausdrücke für mittlere Antwortzeit und Warteschlangenlänge

O. Tickoo, B. Sikdar:
„Queueing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC based Wireless Networks“
(2004, IEEE 0-7803-8355-9/04)

Gegenstand
Analytische Untersuchung von Queueing Delays für 802.11 DCF
Annahmen, Vereinf.
- RTS/CTS wird genutzt
- beliebiger Ankunftsprozeß, aber für alle Stationen gleich
- beliebige Paket-Länge
- feste Anzahl Stationen

Modelleigenschaften
- G/G/1-Modellierung der Knoten des Netzes
- Berücksichtigung der 802.11e Mac-Erweiterung (EDCF)

Ergebnisse
analytische Ausdrücke für Delay und Warteschlangenlänge

Guoping Zeng, Imrich Chlamantac:
„A Finite Queueing Model For IEEE 802.11 Mac Protocol“

Gegenstand
Analytische Untersuchung der MAC Queueing Delays sowie der Blockierungswahrscheinlichkeit (Abweisung von Daten wegen voller MAC-Sendequeue).

Annahmen, Vereinf.
- Infrastrukturnetz mit fester Anzahl Stationen
- nur der Verkehr von den Stationen zum Access Point wird berücksichtigt
- Poisson-Ankunftsprozesse an den Stationen
- Paketlängen exponentiell verteilt

Ergebnisse
- analytische Ausdrücke für Queueing Delay und Blockierungswahrscheinlichkeit
- vergleichende Simulationsergebnisse
Kapitel 7
Validierung mathematischer Modelle

Im Rahmen dieser Arbeit wurden verschiedene Methoden untersucht, mit denen sich die Ergebnisse eines mathematischen Modells überprüfen oder zumindest vergleichen lassen.

Eine wirkliche Überprüfung kann nur an der Realität erfolgen, d.h. die Modellergebnisse müssten mit Meßergebnissen aus realen Rechnernetzen verglichen werden. Aussagekräftige Ergebnisse sind nur durch Messungen vielfältiger Szenarien zu erlangen. Aufgrund der Komplexität dieses Unterfangens ist eine Überprüfung des Modells durch Messungen im Rahmen dieser Arbeit nicht möglich.

Es wurde auch nach publizierten Messergebnissen gesucht, bei denen die mitveröffentlichten Kontextinformationen es erlauben würden, das Szenario nachzumodellieren und die Ergebnisse zu vergleichen. Es konnten allerdings keine ausreichend detaillierten Veröffentlichungen gefunden werden.


7.1 NS-2


\(^1\)http://www.isi.edu/nsnam/ns
\(^2\)General Public License
7.1.1 Prinzip der Simulation mit NS-2

Das Paket NS-2 besteht aus mehreren Programmen. Die wichtigsten davon sind:

- der Netzwerksimulator selbst
- das Animationswerkzeug NAM (Network Animator)
- der Plotter XGraph

Die eigentliche Simulation eines Netzwerkes läuft vorm Anwender verborgen ab. Es können allerdings verschiedene Trace-Dateien erzeugt werden. Diese können anschließend genutzt werden, um den Ablauf der Simulation mit NAM zu visualisieren bzw. mit XGraph Diagramme zu den aufgezeichneten Größen wie z.B. der Netzwerkauslastung zu erzeugen.

Der eigentliche Simulator ist in C++ geschrieben. Er wird über eine OTcl³-Schnittstelle mit Eingabedaten versorgt. Diese „Zweisprachigkeit“ gewährleistet einerseits eine effiziente Ausführung der Simulation und andererseits eine einfache Möglichkeit, Simulationsszenarien zu erstellen und zu verändern.


```sh
# Simulator-Objekt erzeugen
set ns [new Simulator]

# Tracefile festlegen
set nf [open out.nam w]
$ns namtrace-all $nf

# Aktionen am Ende der Simulation
³Objectoriented Tool Command Language
7.1 NS-2

```plaintext
proc finish () {
    global ns nf
    $ns flush-erase
    close $nf
    exec nam out.nam &
    exit 0
}

# Knoten erzeugen
set n0 [$ns node]
set n1 [$ns node]

# Verbindungen erzeugen
$ns duplex-link $n0 $n1 1Mb 10ms DropTail

# Agenten installieren und verbinden
set udp0 [new Agent /UDP]
$ns attach-agent $n0 $udp0

set null0 [new Agent /Null]
$ns attach-agent $n1 $null0

$ns connect $udp0 $null0

# Applikationen installieren
set cbr0 [new Application /Traffic /CBR]
$cbr0 set packetSize_ 2500
$cbr0 set interval_ 0.1
$cbr0 attach-agent $udp0

# Ereignisse festlegen
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"

# Simulation starten
$ns run
```

7.1.2 Unterstützung der WLAN-Modellierung

ten Zeitpunkt eine Bewegung zu einem Ziel mit einer vorgegebenen Geschwindigkeit initiiert wird.

Das CSMA/CA-Zugriffsverfahren ist in NS-2 implementiert.

Die Simulation von Infrastrukturen (\textit{wired-cum-wireless Szenarios}) wird durch Objekte vom Typ \texttt{BaseStationNode} ermöglicht, die der Simulation von Access Points dienen.

MobileIP wird von NS-2 ebenfalls unterstützt.

### 7.1.3 Funkausbreitungsmodelle

NS-2 unterstützt drei Ausbreitungsmodelle für Funkwellen. Das einfachste ist das \textit{Free space model}. Es geht von idealen Bedingungen aus und berücksichtigt nur eine Ausbreitung entlang der direkten Sichtverbindung.

Das zweite Modell ist das \textit{Two-ray ground reflection model}, welches zusätzlich zur direkten Sichtverbindung einen zweiten Ausbreitungspfad durch Bodenreflexion berücksichtigt. Dieses Modell liefert bei längeren Strecken bessere Ergebnisse als das einfache Modell.

Als drittes wird ein statistisches Modell angeboten, das Umgebungen ohne Sichtverbindungen simulieren kann: das \textit{Shadowing model}. Das Modell kann durch zwei Parameter an unterschiedliche Umgebungen angepasst werden. In der Dokumentation von NS-2 werden Parametersätze für folgende Umgebungen vorgeschlagen:

- Außenbereich, Sichtverbindung
- Außenbereich, städtisches Gebiet ohne Sichtverbindung
- im Gebäude, Sichtverbindung
- im Gebäude, keine Sichtverbindung
- Gebäudetypen: Büro, massiv; Büro, Leichbauweise; Fabrikhalle

Keines der Modelle ist in der Lage, die Ausbreitung von Funkwellen unter Berücksichtigung einer konkreten Gebäudegeometrie zu simulieren.

7.2 OMNeT++


OMNeT++ ist für vielfältige Zwecke geeignet und bei weitem nicht auf die Modellierung von Rechnernetzen beschränkt. Es bietet allerdings von Haus aus auch keine spezielle Unterstützung für diesen Zweck, sondern lediglich ein umfangreiches, gut durchdachtes Rahmenwerk zur Erzeugung ereignisgesteuerter Simulationen.

Mittlerweile existiert eine Vielzahl von Projekten, die auf OMNeT++ aufbauen und Funktionalität für spezielle Anwendungen bereitstellen, unter anderem auch zur Simulation von Netzwerkprotokollen und mobiler Kommunikation.

Dieser Abschnitt soll sowohl Grundzüge von OMNeT++ als auch einige Erweiterungen dafür vorstellen.

7.2.1 Zusammenspiel der Bestandteile von OMNeT++

OMNeT++ ist ein modular aufgebautes Programmpaket, im Folgenden soll ein Überblick über die einzelnen Bestandteile und deren Zusammenspiel gegeben werden. Hier wird nur auf technische Aspekte eingegangen, die eigentliche Modellierung wird in den nächsten Abschnitten beschrieben.


Es folgt eine kurze Beschreibung der Komponenten des OMNeT++-Paketes:

<table>
<thead>
<tr>
<th>Komponente</th>
<th>Funktion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulationsbibliothek</td>
<td>Kernstück von OMNeT++. Stellt Framework von Klassen bereit, auf die bei der Modellierung aufgebaut werden kann (und muß)</td>
</tr>
<tr>
<td>NED-Compiler</td>
<td>erzeugt C++-Quelltext aus NED-Dateien.</td>
</tr>
<tr>
<td>NED-Editor</td>
<td>Editor zur grafischen Erzeugung/Bearbeitung von NED-Dateien</td>
</tr>
<tr>
<td>Oberflächenbibliotheken</td>
<td>stellen Benutzerschnittstellen für Simulationen bereit. OMNeT++ bietet neben der grafischen Schnittstelle auch ein Textmodus-Interface.</td>
</tr>
<tr>
<td>Utilities</td>
<td>diverse Hilfsprogramme, die u.a. den Entwicklungsprozeß mit OMNeT++ automatisieren helfen.</td>
</tr>
</tbody>
</table>

\(^4\)http://www.omnetpp.org
7.2.2 Prinzip der Modellierung mit OMNeT++

Ein OMNeT++-Modell besteht aus hierarchisch geschachtelten Modulen. Die Schachtelungstiefe ist dabei unbegrenzt. Dies erlaubt es, die Strukturen des realen Systems beliebig genau nachzubilden. Prinzipiell wird zwischen zwei Arten von Modulen unterschieden:

- den *einfachen Modulen* (engl.: *simple modules*)
- und den *zusammengesetzten Modulen* (engl.: *compound modules*).


- Untermodule mit ihrem Obermodul
- Untermodule eines Obermoduls untereinander
- zwei Gates eines Moduls
Das bedeutet, daß Verbindungen über mehrere Hierarchieebenen hinweg nicht gestattet sind. Diese Einschränkung ist jedoch positiv zu sehen: strikte Modularisierung und damit die Wiederverwendbarkeit entwickelter Module werden so gefördert.

Nachrichten können beliebig komplexe Objekte enthalten. Wie die Module sind auch Nachrichten streng typisiert.

### 7.2.3 Modellierung von Rechnernetzen mit OMNeT++

Wie schon gesagt, verfügt das OMNeT++-Paket selbst über keine spezielle Unterstützung bei der Modellierung von Rechnernetzen. Lediglich in den mitgelieferten Simulationsbeispielen werden einige Funktionsprinzipien und Protokolle aus dem Rechnernetzbereich modelliert.

Es sind jedoch einige Frameworks entstanden, die OMNeT++ um Funktionen zur Rechnernetzsimulation erweitern.


### 7.3 Vergleich

Beide vorgestellten Simulationswerkzeuge besitzen ihre Stärken. NS-2 ist ein etablierter, ausgefeilter Netzwerksimulator mit Unterstützung für verschiedenste Protokolle. Der Simulator gilt als

OMNeT++ bietet den Vorteil eines klar strukturierten Modellierungsansatzes. Der erleichtert die Einarbeitung und unterstützt die Entwicklung modularer Modelle, die softwaretechnischen Qualitätsmaßstäben genügen. Da OMNeT++ noch ein relativ junges Projekt ist und das Einsatzspektrum gegenüber NS-2 genereller angelegt ist, ist die Unterstützung für die Simulation von Netzwerken noch lückenhaft. Dies betrifft auch Modelle für die gängigen WLAN-Standards. Da der Simulator aber sehr aktiv weiterentwickelt wird und an vielen Universitäten Projekte existieren, die OMNeT++ als ihre Basis nutzen, ist sicherlich mit einer Besserung der Situation zu rechnen. Es wäre zu untersuchen, inwieweit sich die bestehenden Netzwerkmodelle für OMNeT++ nutzen und gegebenenfalls miteinander integrieren lassen, um größere, heterogene Netzwerke zu simulieren.
Kapitel 8
Andere Modelle


Die meisten der vorgestellten Arbeiten entstanden im Rahmen des CANDY-Projektes.

8.1 „Analytische Verfahren für die Kapazitätsplanung von Rechnernetzen auf Basis der Bedienungstheorie“

In seinem 2003 eingereichten Größen Beleg ([10]) untersuchte Wolfgang Fäller die Modellierung von Rechnernetzen mit Hilfe der Bedienungstheorie. Das im Rahmen der Arbeit entstandene Analyseprogramm ist in C++ geschrieben und unterstützt die Netzwerkgeräte Switch und Workstation.

In [10] wird für jedes zu analysierende Szenario ein eigenes Analyseprogramm generiert. Die Beschreibung des zu untersuchenden Szenarios wird aus einer NDML-Datei gewonnen, welche mit Hilfe von XSLT in einen C++-Quelltext transformiert wird, der die main()-Funktion des Analyseprogrammes enthält. Dieser wird anschließend kompiliert und gegen die Programmbibliothek gelinkt, die die zur Modellierung notwendigen Klassen bereitstellt. Das erzeugte Programm generiert dann zur Laufzeit eine XML-Datei, die die ermittelten Leistungsgrößen für die einzelnen Geräte und Medien des Szenarios enthält. Bei den Kenngrößen handelt es sich um

- die Auslastung (für Device und Medium) sowie
- die Leerlaufwahrscheinlichkeit (für Device).


8.2 LAN-Lastmodell von Wang Xuewei

Die Diplomarbeit von Wang Xuewei ([15]) befasst sich mit der Lastanalyse in lokalen Netzen. Basierend auf einer grundsätzlichen Analyse der in einem LAN auftretenden Netzwerklast werden
1. Email
oft ☐ ☐ ☐ ☐ ☐ nie
2. Chatten
oft ☐ ☐ ☐ ☐ ☐ nie
3. Telefonieren (PC to Phone)
onf ☐ ☐ ☐ ☐ ☐ nie
4. Videokonferenzen
oft ☐ ☐ ☐ ☐ ☐ nie
5. Recherche / Suchmaschinen
oft ☐ ☐ ☐ ☐ ☐ nie
6. Einkaufs-/ Verkaufsportale
oft ☐ ☐ ☐ ☐ ☐ nie
7. Foto-Onlineentwicklung
oft ☐ ☐ ☐ ☐ ☐ nie
8. Online Video(TV)
onf ☐ ☐ ☐ ☐ ☐ nie
9. Internetradio
oft ☐ ☐ ☐ ☐ ☐ nie
10. Downloads von Software
oft ☐ ☐ ☐ ☐ ☐ nie
11. Onlinespiele
oft ☐ ☐ ☐ ☐ ☐ nie
12. File sharing
oft ☐ ☐ ☐ ☐ ☐ nie

Abbildung 8.1: Lastermittlungs-Fragebogen (Quelle: [15])

Fragebögen erarbeitet, mit deren Hilfe die durchschnittliche Last im LAN durch Befragung der Mitarbeiter erfasst werden soll - siehe Abbildung 8.1.

Die Antworten auf die Fragen werden zur Gewichtung ermittelte Referenzwerte der Lastanforderungen für einzelnen Anwendungen bzw. Dienste verwendet und ergeben damit die Datenlast, die für die jeweilige Anwendung angesetzt wird.

Desweiteren wurde als Lastmodell der sogenannte Summe-Algorithmus entwickelt. Dabei werden in baumförmigen geswitchten LANs die Datenlasten des Internet-Traffic, beginnend bei den Blättern (Endgeräte: PCs, ...), beim Aufstieg im Topologiebaum entlang der Kanten aufsummiert. Eine Gesamtlast ergibt sich dann an der Wurzel des Baumes - dem Gateway ins Internet.

Außerdem wurde eine grafische Darstellung der Netztopologie mit den ermittelten Datenlasten realisiert, die auf der CANDY-Version von Feiyue Zhou basiert.
8.2.1 Wertung


Der Algorithmus zur Lastanalyse berücksichtigt bisher nur Traffic von den Endgeräten hin zum Gateway ins Internet, jedoch nicht die Datenströme zwischen Geräten des betrachteten LANs.

8.3 Der StandortFinder-Algorithmus

Im Großen Beleg von Thomas Fahnert ([9]) wurde ein Programm zur Hotspot-Platzierung entwickelt: der StandortFinder. Der dazu entwickelte heuristische Algorithmus hat das Ziel, für die jeweils gegebene Situation optimale Access Point Standorte zu ermitteln.


Aus diesen Eingabedaten ermittelt der Algorithmus eine Menge von Standorten für Access Points, so daß der gesamte Empfangsbereich bei minimaler Anzahl Access Points vollständig abgedeckt wird.

8.3.1 Verwendetes Ausbreitungsmodell

Folgende Formel wird zur Ermittlung der Empfangsleistung $P_E$ verwendet:

$$P_E = P_S - 20 \cdot \log_{10} \left( \frac{4\pi f}{c} \sqrt{(h_D - h_T)^2 + d^2} \right) + G_S + G_E - D_W - P_{Sicher}$$


### 8.3.2 Algorithmus

Der Algorithmus des StandortFinder ist in Abbildung 8.2 dargestellt.


Ist die vollständige Abdeckung möglich, wird als initialer Standort für den Access Point die Mitte von $x$- und $y$-Ausdehnung des Empfangsbereiches gewählt.

Als nächstes wird geprüft, ob dieser Standort innerhalb des erlaubten Bereichs für die Access Point-Aufstellung liegt. Falls nicht, wird er zum nächstgelegenen Punkt des erlaubten Bereiches verschoben.


Im negativen Fall wird geprüft, ob eine Verschiebung des Access Point möglich ist. Dies ist zum Beispiel der Fall, wenn in einer Himmelsrichtung am Rand des Empfangsbereiches der geforderte Empfangspegel nicht erreicht wird, in der entgegengesetzten Richtung jedoch ein höherer Wert als gefordert vorliegt.

Ist eine Verschiebung nicht möglich oder nicht ausreichend, bzw. fiel der Anfangstest auf theoretische Abdeckbarkeit negativ aus, muß die Empfangsfläche geteilt werden. Dabei wird die maximale Ausdehnung der Fläche halbiert. Für beide Teilflächen wird der Algorithmus erneut ausgeführt.

### 8.3.3 Programmtechnische Realisierung

Der Algorithmus wurde von Thomas Fahnert in Form des Java-Programmes StandortFinder realisiert.

Zur Erzeugung der als Eingabe erforderlichen Gebäudegrundrisse dient das frei erhältliche Programm PythonCAD. Dieses erzeugt ein XML-Format, welches vom StandortFinder eingelesen wird.

Der StandortFinder kennt die Funkstandards 802.11a,b,g sowie 802.16a. Weitere Standards können hinzugefügt werden.
Abbildung 8.2: StandortFinder: Algorithmus (Quelle: [9])
Nach Ausführung des Algorithmus gibt der StandortFinder eine mit PythonCAD lesbare XML-Datei aus, in der die ermittelten Standorte für Access Points verzeichnet sind.

### 8.3.4 Wertung


### 8.4 Wireless Planer


#### 8.4.1 Lastaspekte

Die im StandortFinder als Empfangsbereich bezeichnete Fläche, also der Bereich, in dem mit der vorgegebenen Signalstärke ein Empfang möglich sein soll, wird beim Wireless Planner durch eine *User Area* ersetzt. Dieser wird eine Benutzerzahl und eine minimale Datenrate zugeordnet, die pro Benutzer zur Verfügung stehen soll. Es wird davon ausgegangen, daß die Nutzer gleichmäßig innerhalb der User Area verteilt sind und daß die Anforderungen an die minimale Datenrate für alle Benutzer gleich sind.

Zunächst wird nach folgender Formel ein initialer Wert für die Anzahl der erforderlichen Access Points $N_{AP}$ ermittelt:

$$N_{AP} = \left\lfloor \frac{n_{user} \cdot DR_{min}}{DR_{AP}} \right\rfloor$$

mit

- $n_{user}$ = Anzahl der Clients
- $DR_{min}$ = minimale Datenrate pro Benutzer
- $DR_{AP}$ = Datenrate, die ein Access Point bereitstellen kann.

Die User Area wird anschließend in $N_{AP}$ gleichgroße Teilflächen unterteilt. Für jede von diesen wird dann der ursprüngliche StandortFinder-Algorithmus ausgeführt, jedoch mit dem im folgenden beschriebenen Unterschied.
8.4.2 Verschiedene Ausbreitungsmodelle

Im Gegensatz zum StandortFinder sind im Wireless Planner zwei Ausbreitungsmodelle implementiert, auch die Integration weiterer Modelle ist denkbar. So kann für Indoor- und Outdoor-Szenarien jeweils ein optimal geeigneter Algorithmus verwendet werden. Für Indoor-Szenarien ist das Multi Wall Model vorgesehen, für Outdoor-Szenarien das COST 231 Walfish-Ikegami-Model.

Für den StandortFinder-Algorithmus spielt es keine Rolle, ob Access Points (Hot Spots) in geschlossenen Räumen oder in städtischen Umgebungen aufgestellt werden sollen - im letzteren Fall können die Dächer von Häusern als erlaubte Bereiche zur Standortwahl vorgesehen werden.

8.4.3 Algorithmus

Der gesamte Algorithmus aus [12] ist in Abbildung 8.3 dargestellt.

8.4.4 Wertung


8.5 Multifarben-Tintenfleck-Algorithmus


Es wird in beiden Arbeiten nicht die Ausbreitung der Funkwellen an sich, sondern die Ausbreitung der verfügbaren Datenrate dargestellt.

Seinen Namen hat der Algorithmus wegen der Art der Visualisierung der Funk-Abdeckung durch die Access Points erhalten. Die Bereiche um einzelne Access Points werden, analog Tintenflecken, mit verschiedenen Farben eingefärbt, wobei die Farben mit zunehmender Entfernung zum Access Point blasser werden, um das Absinken der Datenrate zu verdeutlichen.

Das Modell von A. Geitmann betrachtet dabei eine einzelne Etage, die durch ihren Grundriss und die Eigenschaften der vorhandenen Wände gegeben ist. S. Silberbach ([14]) hat die Betrachtung auf mehrstöckige Gebäude ausgedehnt, wobei zusammen mit der aktuell betrachteten Etage immer jeweils die darüber- und die darunterliegende Berücksichtigung finden. Außerdem hat er das Modell um eine Lastanalyse erweitert.

Die folgenden Aussagen beziehen sich auf das Modell aus [14].
Ermittlung der benötigten Anzahl von AP/BS

Aufteilung der User Area

Für jede Teilfläche

Platzieren der AP/BS

Innerhalb des erlaubten Bereiches?

ja

Nächstgelegenen erlaubten Punkt suchen und dahin versetzen

nein

Empfang in der gesamten Fläche?

ja

Standort gefunden

nein

Empfang in der gesamten Fläche?

ja

Verschieben

nein

Für jede neue Teilfläche

Fläche teilen

Abbildung 8.3: Erweiterter SiteFinder-Algorithmus (Quelle: [12])
8.5.1 Formel

Folgende Formel wird zur Berechnung der verfügbaren Datenrate genutzt:

\[ C_i^r(x,y) = Damp \cdot \frac{BWMax}{Dist_i(x,y)^2} \cdot \frac{1}{n_i} \]

mit

- \( C_i^r(x,y) \) = verfügbare Datenrate in Zelle \( C(x,y) \) von Raum \( r \) bei Bedienung durch Access Point \( i \)
- \( Damp \) = Dämpfungsfaktor
- \( Dist_i(x,y) \) = Entfernung der Zelle zu Access Point \( i \)
- \( n_i \) = Anzahl der Nutzer von Access Point \( i \) in Raum \( r \)

Bei der Ermittlung des Dämpfungsfaktors werden die auf dem Weg von der betrachteten Zelle zum Standort des Access Points liegenden Wände berücksichtigt. Befindet sich der AP auf einer anderen Etage, wird die betrachtete Zelle auf die Etage des APs projiziert. Es wird dann die Dämpfung der Wände dieser Etage und die der zwischen AP und betrachteter Zelle liegenden Decke berücksichtigt.

8.5.2 Analysemodi


Im zweiten Komplex werden konkrete Benutzerprofile berücksichtigt. Diese werden unter Nutzung von Applikationsprofilen erstellt. Dabei können für die benötigten Datenraten der Anwendungen sowohl Durchschnittswerte als auch Maximalwerte (worst case) angenommen werden.


8.5.3 Wertung

Wie schon in anderen Arbeiten ([12],[8]) festgestellt wurde, ist der Ansatz problematisch, die Ausbreitung von Datenrate im Raum darzustellen. Ein Funkausbreitungsmodell kann die Signalstärke


**8.6 Fazit**


Durch die Verwendung verschiedener Modelle können sowohl Indoor- als auch Outdoor-Szenarien betrachtet werden. Die Implementierung weiterer Modelle ist durch den modularen Aufbau problemlos möglich.


Kapitel 9
Zusammenfassung und Ausblick

Im Mittelpunkt dieser Arbeit stand die Kapazitätsanalyse von Wireless LAN durch Modellierung mit Hilfe der Bedienungstheorie.

Zunächst wurden in den einführenden Kapiteln relevante Aspekte der WLAN-Standards und Grundlagen der Bedienungstheorie dargestellt.


Um einen sinnvollen Einsatz des Programms „Kapazitätsanalyse“ zu ermöglichen, ist die Software um Modelle für andere Netzwerktechnologien zu erweitern.

Das sechste Kapitel gibt einen Überblick über publizierte bedienungstheoretische Modelle. Ein Modell wird aufgrund des interessanten Ansatzes und der häufigen Weiterverwendung in anderen Arbeiten näher vorgestellt. Es zeigte sich, daß die betrachteten Modelle für die hier vorliegende Arbeit nicht nutzbar waren, da die dort gewählte analytische Betrachtung der WLAN-Macschicht vereinfachende und pauschalisierende Annahmen bezüglich der Rahmenbedingungen erforderlich
machte, die für diese Arbeit nicht übernommen werden können.


Schließlich wurden im achten Kapitel einige Modelle anderer Diplom- und Belegarbeiten betrachtet. Dabei handelte es sich um Funkausbreitungsmodelle und alternative Modelle zur Kapazitäts- und Lastanalyse.


Ein weiteres Problem ist die realistische Bestimmung der Datenlasten, die der Modellierung eines Rechnernetzes zugrunde gelegt werden. Existiert das betrachtete System noch nicht, bietet sich die Nutzung bekannter statistischer Verteilungen oder die Bestimmung empirischer Verteilungen aus Messwerten eines als repräsentativ betrachteten Referenzsystems an. In beiden Fällen ist die Berücksichtigung der Gegebenheiten des geplanten Systems erforderlich. Lastmodellierung ist ein schwieriger und aufwändiger Prozeß, muß aber hinreichend gründlich betrieben werden, wenn sich die bedienungstheoretische Kapazitätsanalyse lohnen soll.
### Glossar

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>CANDY</td>
<td>Computer Aided Network Design Utility</td>
</tr>
<tr>
<td>CCK</td>
<td>Complementary Code Keying</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Interframe Space</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>EIFS</td>
<td>Extended Interframe Space</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronical Engineers</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control Layer</td>
</tr>
<tr>
<td>NDML</td>
<td>Network Design Markup Language</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer, Bitübertragungsschicht</td>
</tr>
<tr>
<td>PIFS</td>
<td>Point (Coordination Function) Interframe Space</td>
</tr>
<tr>
<td>PLCP</td>
<td>Physical Layer Convergence Procedure</td>
</tr>
<tr>
<td>PMD</td>
<td>Physical Medium Dependent</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>SDU</td>
<td>Service Data Unit</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Interframe Space</td>
</tr>
<tr>
<td>SNAP</td>
<td>Sub Network Access Protocol</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSL</td>
<td>Extensible Stylesheet Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>XSL Transformation</td>
</tr>
<tr>
<td>Tabellenverzeichnis</td>
<td>Seite</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
</tr>
<tr>
<td>2.1 Weitere 802.11-Standarderweiterungen</td>
<td>8</td>
</tr>
<tr>
<td>2.2 OFDM-Datenraten</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Parameter der PHY Realisierungen</td>
<td>13</td>
</tr>
<tr>
<td>4.1 Durchsatz bei unterschiedlichen Datenraten (Quelle: [26])</td>
<td>42</td>
</tr>
<tr>
<td>4.2 Auswirkung der Pad-Bits auf die Übertragungsdauer</td>
<td>48</td>
</tr>
</tbody>
</table>
Abbildungsverzeichnis

1.1 CANDY-Architektur, Komponenten ........................................... 3
1.2 Beispielszenario zur Kapazitätsanalyse .................................... 3

2.1 Hidden Station Problem .......................................................... 16

3.1 Bedienstation ................................................................. 21
3.2 Kendall-Notation ............................................................... 22
3.3 Zusammenfassung und Aufspaltung von Poisson-Prozessen ........... 23
3.4 Offenes Bedienungsnetz ....................................................... 25
3.5 Geschlossenes Bedienungsnetz ............................................... 25
3.6 Beispiel Gleichgewichtsgleichungen ....................................... 27

4.1 Teilaufgaben bei der Modellierung ......................................... 31
4.2 Komponenten im System „Rechnernetz“ ................................ 35
4.3 Mögliche Detailstufen der Abbildung eines Rechnernetzes ............. 36
4.4 WLAN-Modellierungsansatz ................................................. 37
4.5 Wireless Bridge .................................................................... 37
4.6 Beispiel zur Modellierung eines AP mit 2 Interfaces .................... 38
4.7 Durchsatzverluste durch Wettbewerb ..................................... 43
4.8 Approximation der Wettbewerbsverluste durch $f_{\text{con}}(n)$ .......... 44
4.9 Durchsatzverluste durch Wettbewerb 2 (Quelle: [32]) .................. 45
4.10 Ablauf der Übertragung eines MAC-Frames ............................ 47

5.1 Hauptfenster des Programmes „Kapazitätsanalyse“ ....................... 54
5.2 Die Karteikarte „Optionen“ ....................................................... 55
5.3 Die Karteikarte „Modellmanipulation“ ...................................... 57
5.4 Ändern der Technologie-Einstellung ......................................... 58
5.5 Auswahl der genutzten Datenrate ............................................ 59
5.6 Klassendiagramm LastAnalyse ............................................... 61
5.7 Entfernen eines Knotens ....................................................... 65
5.8 Klassendiagramm LastAnalyse.Overheads ................................. 66
5.9 JAXB-Architektur (Quelle: SUN (java.sun.com)) ......................... 70
5.10 Szenario zur Lastanalyse .................................................... 71
Abbildungsverzeichnis

6.1 Markov-Kette für Backoff-Prozeß (Quelle: [17]) .......................... 76
7.1 OMNeT++ Modul-Hierarchie (Quelle: [37]) ............................... 90
7.2 Nachrichtenaustausch zwischen Modulen (Quelle: [37]) ................. 90
8.1 Lastermittlungs-Fragebogen (Quelle: [15]) ............................... 94
8.2 StandortFinder: Algorithmus (Quelle: [9]) ............................... 97
8.3 Erweiterter SiteFinder-Algorithmus (Quelle: [12]) ....................... 100
Literaturverzeichnis


[7] WLAN-Standards (802.11, 802.11b, 802.11g, 802.11a, 802.11h). IEEE.


[38] Kevin Fall, Kannan Varadhan (Editors). The ns manual.

Erklärung

Ich versichere hiermit, daß ich die vorliegende Arbeit selbständig unter ausschließlicher Verwendung der angegebenen Quellen und Hilfsmittel verfaßt habe, sowie Zitate aus anderen Werken mit Quellenangaben kenntlich gemacht habe.

Dresden, den 17. August 2006

Stefan Uhlig

Danksagung

Bedanken möchte ich mich bei meinen Betreuern, Dr. Güter und Dr. Luntovskyy, die mir stets durch Anregungen und Ideen sowie Diskussion von Problemen behilflich waren.
Beiliegende Software

Die im Rahmen dieser Arbeit entwickelte Software sowie die Javadoc-Klassendokumentation befinden sich auf der beiliegenden CD. Näheres zum Inhalt siehe Datei “inhalt.html” auf dem Datenträger.