LEHRBUCH

Christian Baun

Computer Networks / Computernetze

Bilingual Edition: English – German / Zweisprachige Ausgabe: Englisch – Deutsch

Springer Vieweg

Computer Networks / Computernetze

Christian Baun

Computer Networks / Computernetze

Bilingual Edition: English – German / Zweisprachige Ausgabe: Englisch – Deutsch

Springer Vieweg

Christian Baun
Fachbereich 2: Informatik und
Ingenieurwissenschaften
Frankfurt University of Applied Sciences
Frankfurt am Main, Deutschland

ISBN 978-3-658-26355-3 ISBN 978-3-658-26356-0 (eBook)
https://doi.org/10.1007/978-3-658-26356-0

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer Vieweg
© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.
Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.
Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature
Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Preface

This compact book on the broad topic of computer networks intends to provide an overview about essential networking technologies, network devices, protocols, transmission media and tools, and to descriptively explain how computer networks work.

Also, this book is intended to assist readers who wish to improve their knowledge in the subject of computer networks as well as in the English or German language.

The motivation to write this bilingual textbook arose from recent changes in everyday life of universities. Many universities have migrated individual modules or even complete study programs to English to be more attractive for international students and to improve the language skills of the graduates.

The errata list will be published here: `http://www.christianbaun.de`.

At this point, I want to thank my editor Sybille Thelen for her support. Also, many thanks to Bernd Böhm, Eicke Godehardt, Torsten Wiens and especially Katrin Baun for the proofreading.

Frankfurt am Main
March 2019

Vorwort

Dieses kompakte Werk über das breite Thema Computernetze soll einen Überblick über die wichtigsten Vernetzungstechnologien, Netzwerkgeräte, Protokolle, Übertragungsmedien und Werkzeuge bieten und anschaulich erklären, wie Computernetze funktionieren.

Zudem soll dieses Buch diejenigen Leser unterstützen, die sich nicht nur fachlich im Thema Computernetze, sondern auch sprachlich in der englischen oder in der deutschen Sprache weiterbilden möchten.

Die Motivation, dieses zweisprachige Lehrbuch zu schreiben, ergab sich aus den Veränderungen des Hochschulalltags in jüngerer Vergangenheit. Zahlreiche Hochschulen haben einzelne Module oder gar komplette Studiengänge auf die englische Sprache umgestellt, um für ausländische Studenten attraktiver zu sein, und um die sprachlichen Fähigkeiten der Absolventen zu verbessern.

Die Errata-Liste wird hier veröffentlicht: `http://www.christianbaun.de`.

An dieser Stelle möchte ich meiner Lektorin Sybille Thelen für ihre Unterstützung danken. Zudem danke ich Bernd Böhm, Eicke Godehardt, Torsten Wiens und ganz besonders Katrin Baun für das Korrekturlesen.

Prof. Dr. Christian Baun

Contents

1 Introduction **1**

2 Fundamentals of Computer Science **3**
2.1 Bit . . . 3
2.2 Representation of Numbers . . . 4
2.2.1 Decimal System . . . 5
2.2.2 Binary System . . . 5
2.2.3 Octal System . . . 6
2.2.4 Hexadecimal System . . . 6
2.3 File and Storage Dimensions . . . 7
2.4 Information Representation . . . 9
2.4.1 ASCII Encoding . . . 9
2.5 Unicode . . . 12
2.6 Representation of Strings . . . 12

3 Fundamentals of Computer Networking **15**
3.1 History of Computer Networks . . . 15
3.2 Mandatory Components of Computer Networks . . . 17
3.3 Computer Networks distinguished by their Dimensions . . . 18
3.4 Data Transmission . . . 19
3.4.1 Serial and Parallel Transmission . . . 19
3.4.2 Synchronous and Asynchronous Transmission . . . 20
3.4.3 Directional Dependence of Data Transmission . . . 21
3.5 Devices in Computer Networks . . . 21
3.6 Topologies of Computer Networks . . . 24
3.6.1 Bus Network . . . 24
3.6.2 Ring Network . . . 25
3.6.3 Star Network . . . 26
3.6.4 Mesh Network . . . 26
3.6.5 Tree Network . . . 27
3.6.6 Cellular Network . . . 27
3.7 Frequency and Data Signal . . . 28
3.8 Fourier Series and Bandwidth . . . 28
3.9 Bit Rate and Baud Rate . . . 29
3.10 Bandwidth and Latency . . . 29
3.10.1 Bandwidth-Delay Product . . . 30
3.11 Media Access Control . . . 32
3.11.1 Deterministic Media Access Control . . . 32
3.11.2 Non-deterministic Media Access Control . . . 32
3.12 Collision Domain . . . 33

4 Protocols and Reference Models **35**
4.1 TCP/IP Reference Model . . . 36
4.2 Hybrid Reference Model . . . 37
4.2.1 Physical Layer . . . 37
4.2.2 Data Link Layer . . . 38
4.2.3 Network Layer . . . 39
4.2.4 Transport Layer . . . 39
4.2.5 Application Layer . . . 40
4.3 How Communication works . . . 41
4.4 OSI Reference Model . . . 42
4.4.1 Session Layer . . . 42
4.4.2 Presentation Layer . . . 43
4.5 Conclusion on the Reference Models . . . 43

5 Physiscal Layer **45**
5.1 Networking Technologies . . . 45
5.1.1 Ethernet . . . 45
5.1.2 Token Ring . . . 47
5.1.3 Wireless Local Area Network (WLAN) . . . 48
5.1.4 Bluetooth . . . 59
5.2 Transmission Media . . . 62
5.2.1 Coaxial Cables . . . 62
5.2.2 Twisted Pair Cables . . . 64
5.2.3 Fiber-optic Cables . . . 71
5.3 Structured Cabling . . . 72
5.4 Devices of the Physical Layer . . . 73
5.4.1 Impact of Repeaters and Hubs on the Collision Domain . . . 75
5.5 Encoding Data with Line Codes . . . 76
5.5.1 Non-Return to Zero (NRZ) . . . 77
5.5.2 Non-Return to Zero Invert (NRZI) . . . 79
5.5.3 Multilevel Transmission Encoding – 3 Levels (MLT-3) . . . 80
5.5.4 Return-to-Zero (RZ) . . . 80
5.5.5 Unipolar RZ . . . 81
5.5.6 Alternate Mark Inversion (AMI) . . . 82
5.5.7 Bipolar with 8 Zeros Substitution (B8ZS) . . . 82
5.5.8 Manchester . . . 83
5.5.9 Manchester II . . . 84
5.5.10 Differential Manchester Encoding . . . 84
5.6 Line Codes which improve the Payload . . . 85
5.6.1 4B5B Encoding . . . 86
5.6.2 5B6B Encoding . . . 86
5.6.3 8B10B Encoding . . . 88
5.7 Further Line Codes . . . 89
5.7.1 8B6T Encoding . . . 89

6 Data Link Layer **91**
6.1 Devices of the Data Link Layer . . . 91
6.1.1 Learning Bridges . . . 93
6.1.2 Loops on the Data Link Layer . . . 94
6.1.3 Spanning Tree Protocol (STP) . . . 97
6.1.4 Impact of Bridges on the Collision Domain . . . 101
6.2 Addressing in the Data Link Layer . . . 101

6.2.1 Format of MAC Addresses . . . 102
6.2.2 Uniqueness of MAC Addresses . . . 103
6.2.3 Security Aspects of MAC Addresses . . . 103
6.3 Framing . . . 104
6.3.1 Character Count in the Frame Header . . . 104
6.3.2 Byte/Character Stuffing . . . 105
6.3.3 Bit Stuffing . . . 106
6.3.4 Line Code Violations . . . 106
6.4 Framing in current Computer Networks . . . 107
6.4.1 Ethernet Frames . . . 107
6.4.2 WLAN Frames . . . 109
6.4.3 WLAN Control Frames . . . 112
6.5 Maximum Transmission Unit (MTU) . . . 113
6.6 Error-detection Codes . . . 114
6.6.1 Two-dimensional Parity-check Code . . . 114
6.6.2 Cyclic Redundancy Check . . . 115
6.7 Error-correction Codes . . . 117
6.8 Media Access Control Methods . . . 117
6.8.1 Media Access Control Method of Ethernet . . . 118
6.8.2 Media Access Control Method of WLAN . . . 122
6.9 Flow Control . . . 128
6.10 Address resolution with ARP . . . 129

7 Network Layer 131
7.1 Devices of the Network Layer . . . 131
7.1.1 Impact of Routers on the Collision Domain . . . 132
7.1.2 Broadcast Domain . . . 132
7.2 Addressing in the Network Layer . . . 133
7.2.1 Format of IPv4 Addresses . . . 135
7.2.2 Subnets in the IPv4 Address Space . . . 137
7.2.3 Private IPv4 Address Spaces . . . 139
7.2.4 Structure of IPv4 Packets . . . 140
7.2.5 Fragmentation of IPv4 Packets . . . 142
7.2.6 Representation and Structure of IPv6 Addresses . . . 143
7.2.7 Representation of Networks in the IPv6 Address Space . . . 145
7.2.8 Selected IPv6 Address Spaces . . . 146
7.2.9 Embed IPv4 Addresses into the IPv6 Address Space . . . 146
7.2.10 Structure of IPv6 Packets . . . 147
7.3 Forwarding and Path Determination . . . 148
7.4 Routing Information Protocol (RIP) . . . 150
7.4.1 Count-to-Infinity . . . 151
7.4.2 Split Horizon . . . 152
7.4.3 Conclusion on RIP . . . 153
7.5 Open Shortest Path First (OSPF) . . . 155
7.5.1 Routing Hierarchy with OSPF . . . 157
7.5.2 Functioning of OSPF . . . 157
7.5.3 Structure of OSPF Messages . . . 159
7.5.4 Conclusion on OSPF . . . 161
7.6 Internetworking . . . 162
7.7 Network Address Translation . . . 164
7.8 Diagnostic and Error Messages via ICMP . . . 167

8 Transport Layer **171**
8.1 Characteristics of Transport Layer Protocols . . . 171
8.2 Addressing in the Transport Layer . . . 172
8.3 User Datagram Protocol (UDP) . . . 174
8.3.1 Format of UDP Segments . . . 174
8.4 Transmission Control Protocol (TCP) . . . 175
8.4.1 Format of TCP Segments . . . 176
8.4.2 Functioning of TCP . . . 179
8.4.3 Flow Control . . . 183
8.4.4 Congestion Control . . . 187

9 Application Layer **195**
9.1 Domain Name System (DNS) . . . 195
9.1.1 Functioning of TCP . . . 196
9.1.2 Example of a Domain Name Resolution . . . 197
9.2 Dynamic Host Configuration Protocol (DHCP) . . . 199
9.2.1 Functioning of DHCP . . . 199
9.2.2 Structure of DHCP Messages . . . 201
9.2.3 DHCP Relay . . . 202
9.3 Telecommunication Network (Telnet) . . . 203
9.4 Hypertext Transfer Protocol (HTTP) . . . 204
9.5 Simple Mail Transfer Protocol (SMTP) . . . 207
9.6 Post Office Protocol Version 3 (POP3) . . . 208
9.7 File Transfer Protocol (FTP) . . . 209

10 Network Virtualization **213**
10.1 Virtual Private Networks (VPN) . . . 213
10.1.1 Technical types of VPNs . . . 214
10.2 Virtual Local Area Networks (VLAN) . . . 215

11 Command Line Tools **219**
11.1 ethtool . . . 222
11.2 mii-tool . . . 223
11.3 iwconfig . . . 223
11.4 iwlist . . . 224
11.5 arp . . . 224
11.6 ifconfig . . . 225
11.7 ip . . . 225
11.8 iftop . . . 227
11.9 tcpdump . . . 227
11.10 route . . . 228
11.11 ping . . . 228
11.12 traceroute . . . 229
11.13 netstat . . . 229
11.14 nmap . . . 230
11.15 dhclient . . . 231
11.16 dig . . . 231
11.17 ftp . . . 232
11.18 nc . . . 232
11.19 netperf . . . 233
11.20 nslookup . . . 233
11.21 ssh . . . 234

11.22 telnet 234

Technical Terms **237**

Glossary **245**

References **255**

Index **257**

Inhaltsverzeichnis

1 Einleitung **1**

2 Grundlagen der Informationstechnik **3**
2.1 Bit . . . 3
2.2 Repräsentation von Zahlen . . . 4
2.2.1 Dezimalsystem . . . 5
2.2.2 Dualsystem . . . 5
2.2.3 Oktalsystem . . . 6
2.2.4 Hexadezimalsystem . . . 6
2.3 Datei- und Speichergrößen . . . 7
2.4 Informationsdarstellung . . . 9
2.4.1 ASCII-Kodierung . . . 9
2.5 Unicode . . . 12
2.6 Darstellung von Zeichenketten . . . 12

3 Grundlagen der Computervernetzung **15**
3.1 Entwicklung der Computernetze . . . 15
3.2 Zwingend nötige Elemente für Computernetzwerke . . . 17
3.3 Räumliche Ausdehnung von Computernetzen . . . 18
3.4 Datenübertragung . . . 19
3.4.1 Serielle und parallele Übertragung . . . 19
3.4.2 Synchrone und asynchrone Übertragung . . . 20
3.4.3 Richtungsabhängigkeit der Übertragung . . . 21
3.5 Geräte in Computernetzen . . . 21
3.6 Topologien von Computernetzen . . . 24
3.6.1 Bus-Topologie . . . 24
3.6.2 Ring-Topologie . . . 25
3.6.3 Stern-Topologie . . . 26
3.6.4 Maschen-Topologie . . . 26
3.6.5 Baum-Topologie . . . 27
3.6.6 Zellen-Topologie . . . 27
3.7 Frequenz und Datensignal . . . 28
3.8 Fourierreihe und Bandbreite . . . 28
3.9 Bitrate und Baudrate . . . 29
3.10 Bandbreite und Latenz . . . 29
3.10.1 Bandbreite-Verzögerung-Produkt . . . 30
3.11 Zugriffsverfahren . . . 32
3.11.1 Deterministisches Zugriffsverfahren . . . 32
3.11.2 Nicht-deterministisches Zugriffsverfahren . . . 32
3.12 Kollisionsgemeinschaft . . . 33

4 Protokolle und Schichtenmodelle **35**
4.1 TCP/IP-Referenzmodell . . . 36
4.2 Hybrides Referenzmodell . . . 37
4.2.1 Bitübertragungsschicht . . . 37
4.2.2 Sicherungsschicht . . . 38
4.2.3 Vermittlungsschicht . . . 39
4.2.4 Transportschicht . . . 39
4.2.5 Anwendungsschicht . . . 40
4.3 Ablauf der Kommunikation . . . 41
4.4 OSI-Referenzmodell . . . 42
4.4.1 Sitzungsschicht . . . 42
4.4.2 Darstellungsschicht . . . 43
4.5 Fazit zu den Referenzmodellen . . . 43

5 Bitübertragungsschicht **45**
5.1 Vernetzungstechnologien . . . 45
5.1.1 Ethernet . . . 45
5.1.2 Token Ring . . . 47
5.1.3 Wireless Local Area Network (WLAN) . . . 48
5.1.4 Bluetooth . . . 59
5.2 Übertragungsmedien . . . 62
5.2.1 Koaxialkabel . . . 62
5.2.2 Twisted-Pair-Kabel . . . 64
5.2.3 Lichtwellenleiter . . . 71
5.3 Strukturierte Verkabelung . . . 72
5.4 Geräte der Bitübertragungsschicht . . . 73
5.4.1 Auswirkungen von Repeatern und Hubs auf die Kollisionsdomäne . . . 75
5.5 Kodierung von Daten in Netzwerken . . . 76
5.5.1 Non-Return to Zero (NRZ) . . . 77
5.5.2 Non-Return to Zero Invert (NRZI) . . . 79
5.5.3 Multilevel Transmission Encoding – 3 Levels (MLT-3) . . . 80
5.5.4 Return-to-Zero (RZ) . . . 80
5.5.5 Unipolares RZ . . . 81
5.5.6 Alternate Mark Inversion (AMI) . . . 82
5.5.7 Bipolar with 8 Zeros Substitution (B8ZS) . . . 82
5.5.8 Manchester . . . 83
5.5.9 Manchester II . . . 84
5.5.10 Differentielle Manchesterkodierung . . . 84
5.6 Nutzdaten mit Leitungscodes verbessern . . . 85
5.6.1 4B5B-Kodierung . . . 86
5.6.2 5B6B-Kodierung . . . 86
5.6.3 8B10B-Kodierung . . . 88
5.7 Weitere Leitungscodes . . . 89
5.7.1 8B6T-Kodierung . . . 89

6 Sicherungsschicht **91**
6.1 Geräte der Sicherungsschicht . . . 91
6.1.1 Learning Bridges . . . 93
6.1.2 Kreise auf der Sicherungsschicht . . . 94
6.1.3 Spanning Tree Protocol (STP) . . . 97
6.1.4 Auswirkungen von Bridges auf die Kollisionsdomäne . . . 101
6.2 Adressierung in der Sicherungsschicht . . . 101

6.2.1 Format der MAC-Adressen . . . 102
6.2.2 Eindeutigkeit von MAC-Adressen . . . 103
6.2.3 Sicherheit von MAC-Adressen . . . 103
6.3 Rahmen abgrenzen . . . 104
6.3.1 Längenangabe im Header . . . 104
6.3.2 Zeichenstopfen . . . 105
6.3.3 Bitstopfen . . . 106
6.3.4 Verstöße gegen Regeln des Leitungscodes . . . 106
6.4 Rahmenformate aktueller Computernetze . . . 107
6.4.1 Rahmen bei Ethernet . . . 107
6.4.2 Rahmen bei WLAN . . . 109
6.4.3 Spezielle Rahmen bei WLAN . . . 112
6.5 Maximum Transmission Unit (MTU) . . . 113
6.6 Fehlererkennung . . . 114
6.6.1 Zweidimensionale Parität . . . 114
6.6.2 Zyklische Redundanzprüfung . . . 115
6.7 Fehlerkorrektur . . . 117
6.8 Medienzugriffsverfahren . . . 117
6.8.1 Medienzugriffsverfahren bei Ethernet . . . 118
6.8.2 Medienzugriffsverfahren bei WLAN . . . 122
6.9 Flusskontrolle . . . 128
6.10 Adressauflösung mit ARP . . . 129

7 Vermittlungsschicht 131
7.1 Geräte der Vermittlungsschicht . . . 131
7.1.1 Auswirkungen von Routern auf die Kollisionsdomäne . . . 132
7.1.2 Broadcast-Domäne (Rundsendedomäne) . . . 132
7.2 Adressierung in der Vermittlungsschicht . . . 133
7.2.1 Aufbau von IPv4-Adressen . . . 135
7.2.2 Subnetze im IPv4-Adressraum . . . 137
7.2.3 Private IPv4-Adressbereiche . . . 139
7.2.4 Aufbau von IPv4-Paketen . . . 140
7.2.5 Fragmentieren von IPv4-Paketen . . . 142
7.2.6 Darstellung und Aufbau von IPv6-Adressen . . . 143
7.2.7 Darstellung von Netzen im IPv6-Adressraum . . . 145
7.2.8 Einige IPv6-Adressbereiche . . . 146
7.2.9 IPv4-Adressen im IPv6-Adressraum einbetten . . . 146
7.2.10 Aufbau von IPv6-Paketen . . . 147
7.3 Weiterleitung und Wegbestimmung . . . 148
7.4 Routing Information Protocol (RIP) . . . 150
7.4.1 Count-to-Infinity . . . 151
7.4.2 Split Horizon . . . 152
7.4.3 Fazit zu RIP . . . 153
7.5 Open Shortest Path First (OSPF) . . . 155
7.5.1 Routing-Hierarchie mit OSPF . . . 157
7.5.2 Arbeitsweise von OSPF . . . 157
7.5.3 Aufbau von OSPF-Nachrichten . . . 159
7.5.4 Fazit zu OSPF . . . 161
7.6 Netzübergreifende Kommunikation . . . 162
7.7 Network Address Translation . . . 164
7.8 Diagnose und Fehlermeldungen mit ICMP . . . 167

8 Transportschicht **171**
8.1 Eigenschaften von Transportprotokollen . . . 171
8.2 Adressierung in der Transportschicht . . . 172
8.3 User Datagram Protocol (UDP) . . . 174
8.3.1 Aufbau von UDP-Segmenten . . . 174
8.4 Transmission Control Protocol (TCP) . . . 175
8.4.1 Aufbau von TCP-Segmenten . . . 176
8.4.2 Arbeitsweise von TCP . . . 179
8.4.3 Flusskontrolle . . . 183
8.4.4 Überlastkontrolle . . . 187

9 Anwendungsschicht **195**
9.1 Domain Name System (DNS) . . . 195
9.1.1 Arbeitsweise des DNS . . . 196
9.1.2 Auflösung eines Domainnamens . . . 197
9.2 Dynamic Host Configuration Protocol (DHCP) . . . 199
9.2.1 Arbeitsweise von DHCP . . . 199
9.2.2 Aufbau von DHCP-Nachrichten . . . 201
9.2.3 DHCP-Relay . . . 202
9.3 Telecommunication Network (Telnet) . . . 203
9.4 Hypertext Transfer Protocol (HTTP) . . . 204
9.5 Simple Mail Transfer Protocol (SMTP) . . . 207
9.6 Post Office Protocol Version 3 (POP3) . . . 208
9.7 File Transfer Protocol (FTP) . . . 209

10 Netzwerkvirtualisierung **213**
10.1 Virtual Private Networks (VPN) . . . 213
10.1.1 Technische Arten von VPNs . . . 214
10.2 Virtual Local Area Networks (VLAN) . . . 215

11 Kommandozeilenwerkzeuge **219**
11.1 ethtool . . . 222
11.2 mii-tool . . . 223
11.3 iwconfig . . . 223
11.4 iwlist . . . 224
11.5 arp . . . 224
11.6 ifconfig . . . 225
11.7 ip . . . 225
11.8 iftop . . . 227
11.9 tcpdump . . . 227
11.10 route . . . 228
11.11 ping . . . 228
11.12 traceroute . . . 229
11.13 netstat . . . 229
11.14 nmap . . . 230
11.15 dhclient . . . 231
11.16 dig . . . 231
11.17 ftp . . . 232
11.18 nc . . . 232
11.19 netperf . . . 233
11.20 nslookup . . . 233
11.21 ssh . . . 234

11.22 telnet 234

Fachbegriffe **237**

Glossar **245**

Literatur **255**

Index / Stichwortverzeichnis **257**

1

Introduction

This book intends to provide a compact and not comprehensive overview of computer networks and their components. The aim is to assist its readers in gaining a basic understanding of the way modern computer networks and their components work. No prior technical knowledge is required.

The chapters 2 and 3 provide an introduction to the fundamentals of information technology (IT) and computer networking. These topics are necessary to understand computer networks and the contents of this book.

Chapter 4 deals with the basics of protocols and their classification into established reference models. The chapter also describes the communication flow and the way user data travels through the individual protocol layers.

The chapters 5 to 9 describe the individual protocol layers from the bottom layer, which is the Physical Layer, to the top layer, which is the Application Layer.

Chapter 10 provides an introduction to the different variants of network virtualization.

Chapter 11 contains an overview of relevant command line tools for network configuration and the analysis of network-related issues.

Einleitung

Dieses Buch will einen Überblick über Computernetze und deren Komponenten schaffen, ohne dabei den Anspruch auf Vollständigkeit zu erheben. Das Ziel ist es, den Leserinnen und Lesern ein grundlegendes Wissen über die Funktionsweise moderner Computernetze und deren Komponenten zu vermitteln. Technische Vorkenntnisse sind dabei nicht erforderlich.

In den Kapiteln 2 und 3 findet eine Einführung in die Grundlagen der Informationstechnik (IT) und der Computervernetzung statt. Dies ist nötig, um die Thematik Computernetze und den Inhalt dieses Buchs verstehen zu können.

Kapitel 4 beschäftigt sich mit den Grundlagen von Protokollen und deren Einordnung in etablierte Referenzmodelle. Das Kapitel beschreibt auch den Ablauf der Kommunikation und den Weg der Nutzdaten durch die einzelnen Protokollschichten.

In den Kapiteln 5 bis 9 werden die einzelnen Protokollschichten von der untersten Schicht, der Bitübertragungsschicht, bis zur obersten Schicht, der Anwendungsschicht, behandelt.

In Kapitel 10 findet eine Einführung in die Varianten der Netzwerkvirtualisierung statt.

Kapitel 11 enthält eine Übersicht über wichtige Kommandozeilenwerkzeuge zur Netzwerkkonfiguration und Analyse von Netzwerkproblemen.

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_1

2

Fundamentals of Computer Science

A basic understanding of information technology (IT) is required to understand how computer networks work. These fundamentals include the different variants of information representation and representation of numbers, orders of magnitude and the way information (especially textual information) is represented in computers.

Grundlagen der Informationstechnik

Um die Funktionsweise der Computernetze zu verstehen, ist ein grundlegendes Verständnis der Informationstechnik (IT) nötig. Bei diesen Grundlagen handelt es sich um die Möglichkeiten der Informationsdarstellung und Repräsentation von Zahlen, Größenordnungen und die Art und Weise, wie Informationen (speziell Texte) in Rechnern dargestellt werden.

2.1

Bit

A *Bit* is the smallest possible unit of information, and every piece of information is bound to an information carrier [5]. An information carrier, which can be in precisely one of two states, can represent 1 bit of data. The value of one or more bits is called *state*. One bit can represent two states. Different scenarios can represent 1 bit of data. Examples are:

- The position of a switch with two states
- The switching state of a transistor
- The presence of an electrical voltage or charge
- The presence of a magnetization
- The value of a variable with logical truth values

Bit

Ein *Bit* ist die kleinstmögliche Einheit der Information und jede Information ist an einen Informationsträger gebunden [5]. Ein Informationsträger, der sich in genau einem von zwei Zuständen befinden kann, kann die Datenmenge 1 Bit darstellen. Den Wert eines oder mehrerer Bits nennt man *Zustand*. Ein Bit kann zwei Zustände darstellen. Verschiedene Sachverhalte können die Datenmenge 1 Bit darstellen. Beispiele sind:

- Die Stellung eines Schalters mit zwei Zuständen
- Der Schaltzustand eines Transistors
- Das Vorhandensein einer elektrischen Spannung oder Ladung
- Das Vorhandensein einer Magnetisierung
- Der Wert einer Variable mit den logischen Wahrheitswerten

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_2

Table 2.1: The number of representable states doubles with each additional Bit

Bits	States	Bits	States	Bits	States
1	$2^1 = 2$	9	$2^9 = 512$	17	$2^{17} = 131,072$
2	$2^2 = 4$	10	$2^{10} = 1,024$	18	$2^{18} = 262,144$
3	$2^3 = 8$	11	$2^{11} = 2,048$	19	$2^{19} = 524,288$
4	$2^4 = 16$	12	$2^{12} = 4,096$	20	$2^{20} = 1,048,576$
5	$2^5 = 32$	13	$2^{13} = 8,192$	21	$2^{21} = 2,097,152$
6	$2^6 = 64$	14	$2^{14} = 16,384$	22	$2^{22} = 4,194,304$
7	$2^7 = 128$	15	$2^{15} = 32,768$	23	$2^{23} = 8,388,608$
8	$2^8 = 256$	16	$2^{16} = 65,536$	24	$2^{24} = 16,777,216$

If more than two states are required to store information, multiple bits (*bit sequences*) are needed. With n bits, it is possible to represent 2^n different states (see Table 2.1). With 2 bits, $2^2 = 4$ different states can be represented, namely `00`, `01`, `10` and `11`. With 3 bits, even $2^3 = 8$ different states (`000`, `001`, `010`, `011`, `100`, `101`, `110` and `111`) can be represented. With each additional bit the number of possible representable states (bit sequences) doubles [5].

Benötigt man zur Speicherung einer Information mehr als zwei Zustände, sind Folgen von Bits (*Bitfolgen*) nötig. Mit n Bits kann man 2^n verschiedene Zustände darstellen (siehe Tabelle 2.1). Also kann man mit 2 Bits $2^2 = 4$ verschiedene Zustände repräsentieren, nämlich `00`, `01`, `10` und `11`. Mit 3 Bits kann man schon $2^3 = 8$ verschiedene Zustände (`000`, `001`, `010`, `011`, `100`, `101`, `110` und `111`) repräsentieren. Jedes zusätzliche Bit verdoppelt die Anzahl der möglichen darstellbaren Zustände (Bitfolgen) [5].

2.2

Representation of Numbers

Numbers can be represented in various ways. One task of IT is to map numbers from the *real world* into the computer. In this context, the distinction between *value* and *representation* is significant.

In mathematics, numbers are distinguished as elements of different sets of values (natural numbers, integers, real numbers, complex numbers,...). The value of a number is also called *abstract number*, and the value is independent of the representation (for example $0.5 = 1/2$).

However, operations of a computer are not executed on values, but on bit sequences. This is why the representation of numbers is particularly interesting for IT. The representation is determined by the place value system (positional notation) used. Relevant for IT are the decimal system, the binary numeral system, the octal system, and the hexadecimal system.

Repräsentation von Zahlen

Zahlen kann man auf unterschiedliche Arten darstellen. Eine Aufgabe der IT ist es, Zahlen aus der *realen Welt* im Computer abzubilden. Wichtig ist dabei die Unterscheidung zwischen *Wert* und *Darstellung*.

In der Mathematik unterscheidet man Zahlen als Elemente verschiedener Wertemengen (natürliche Zahlen, ganze Zahlen, reelle Zahlen, komplexe Zahlen, usw.). Den Wert einer Zahl nennt man auch *abstrakte Zahl* und der Wert ist unabhängig von der Darstellung (zum Beispiel $0{,}5 = 1/2$).

Operationen eines Rechners werden aber nicht auf Werten, sondern auf Bitfolgen ausgeführt. Darum ist für die IT besonders die Darstellung der Zahlen interessant. Die Darstellung wird vom verwendeten Stellenwertsystem (Positionssystem) bestimmt. Die für die IT wichtigen Stellenwertsysteme sind das Dezimalsystem, das Dualsystem, das Oktalsystem und das Hexadezimalsystem.

2.2.1

Decimal System

The decimal system uses the number 10 as the base. Each digit D at the position i has the value $D \times 10^i$. One example is:

Dezimalsystem

Das Dezimalsystem verwendet als Basis die Zahl 10. Jede Ziffer D an der Stelle i hat den Wert $D \times 10^i$. Ein Beispiel ist:

$$2013 = 2 \times 10^3 + 0 \times 10^2 + 1 \times 10^1 + 3 \times 10^0$$

Computer systems distinguish between two electrical states. Therefore choosing the number 2 as the base and thus the binary system is ideal for IT.

Computer-Systeme unterscheiden prinzipiell zwischen zwei elektrischen Zuständen. Darum ist aus Sicht der IT als Basis die Zahl 2 und damit das Dualsystem optimal geeignet.

2.2.2

Binary System

The binary system uses the number 2 as base. Numbers are represented by using the digits of the values zero and one. Number representations in the binary numeral system are called binary numbers. All positive natural numbers including zero can be represented by sequences of symbols from the set $\{1, 0\}$. If n is the number of bits, x_0 is the *least significant bit* (LSB)) and x_{n-1} is the *most significant bit* (MSB).

Since long series of zeros and ones quickly become confusing to humans, the octal system or the hexadecimal system are often used to represent bit sequences.

The conversion of numbers between different place value systems is simple. For clarification, the place value system of the respective number is subscripted in the following examples.

When converting binary numbers to decimal numbers, the digits are multiplied by their place values, and the results are summed up.

Dualsystem

Das Dualsystem verwendet als Basis die Zahl 2. Zahlen werden nur mit den Ziffern der Werte Null und Eins dargestellt. Zahldarstellungen im Dualsystem heißen Dualzahlen oder Binärzahlen. Alle positiven natürlichen Zahlen inklusive der Null können durch Folgen von Symbolen aus der Menge $\{1, 0\}$ repräsentiert werden. Wenn n der Anzahl der Bits entspricht, ist x_0 das niederwertigste Bit (*Least Significant Bit* – LSB) und x_{n-1} das höchstwertigste Bit (*Most Significant Bit* – MSB).

Da lange Reihen von Nullen und Einsen für Menschen schnell unübersichtlich werden, verwendet man zur Darstellung von Bitfolgen häufig das Oktalsystem oder das Hexadezimalsystem.

Die Umrechnung der Stellenwertsysteme ist einfach möglich. Zur Verdeutlichung ist das Stellenwertsystem der jeweiligen Zahl in den folgenden Beispielen tiefgestellt beigefügt.

Bei der Umwandlung von Dualzahlen in Dezimalzahlen werden die Ziffern mit ihren Stellenwertigkeiten ausmultipliziert und die Ergebnisse addiert.

$$10100100_2 = 2^7 + 2^5 + 2^2 = 164_{10}$$

Different ways exist to convert decimal numbers to binary numbers. One method presents Table 2.2. The decimal number is divided by the base value 2, and the result, as well as the remainder (value zero or one), are noted. In the

Die Umwandlung von Dezimalzahlen in Dualzahlen ist unter anderem mit dem in Tabelle 2.2 gezeigten Verfahren möglich. Dabei wird die Dezimalzahl durch die Basis 2 dividiert und das Ergebnis und der Rest (Wert Null oder Eins)

next round (row of the table), the result of the division is again divided by the base value, and the result, as well as the remainder, are both noted. This algorithm continues until the result of the division is zero.

werden notiert. Das Ergebnis der Division wird in der nächsten Runde (Zeile der Tabelle) erneut durch die Basis dividiert und erneut werden das Ergebnis und der Rest notiert. Dieser Restwertalgorithmus wird so lange weitergeführt, bis das Ergebnis der Division Null ist.

Table 2.2: Converting the decimal number 164_{10} to the binary number 10100100_2

k	*Quotient* k DIV 2	*Remainder* k MODULO 2
164	82	$0 = x_1$
82	41	$0 = x_2$
41	20	$1 = x_3$
20	10	$0 = x_4$
10	5	$0 = x_5$
5	2	$1 = x_6$
2	1	$0 = x_7$
1	0	$1 = x_8$

2.2.3

Octal System

The octal system uses the number 8 as the base and can represent groups of 3 bits with a single character.

When converting from binary numbers to octal numbers, the bit sequence is subdivided beginning at the least significant bit in groups of three. Each group of three corresponds to a single position of the octal number.

Oktalsystem

Das Oktalsystem verwendet als Basis die Zahl 8 und kann Gruppen von 3 Bits mit einem Zeichen darstellen.

Bei der Umwandlung von Dualzahlen in Oktalzahlen wird die Bitkette vom niederwertigsten Bit beginnend in Dreiergruppen unterteilt. Jede Dreiergruppe entspricht einer Stelle der Oktalzahl.

$$164_{10} = 10|100|100_2 = 244_8$$

The conversion of octal numbers to binary numbers is analogous. One digit in the octal system corresponds to three digits in the binary system.

Die Umwandlung von Oktalzahlen in Dualzahlen erfolgt analog. Eine Stelle im Oktalsystem entspricht drei Stellen im Dualsystem.

2.2.4

Hexadecimal System

The hexadecimal system uses the number 16 as base. The representation of positive natural numbers is done with the 16 number symbols from the set $\{0, 1, \ldots 8, 9, A, B, C, D, E, F\}$. A single character can represent a group of 4 bits (tetrade, half-byte or nibble).

Hexadezimalsystem

Das Hexadezimalsystem verwendet als Basis die Zahl 16. Die Darstellung positiver natürlicher Zahlen erfolgt mit den 16 Ziffernsymbolen aus der Menge $\{0, 1, \ldots 8, 9, A, B, C, D, E, F\}$. Ein Zeichen kann eine Gruppe von 4 Bits (Tetrade, Halbbyte oder Nibble) darstellen.

Table 2.3: Different Representations of positive natural Numbers

Decimal representation	Binary representation	Octal representation	Hexadecimal representation
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

When converting binary numbers to hexadecimal numbers, the bit sequence is subdivided into tetrades, starting with the least significant bit. Each tetrade corresponds to one digit of the hexadecimal number.

Bei der Umwandlung von Dualzahlen in Hexadezimalzahlen wird die Bitkette vom niederwertigsten Bit beginnend in Tetraden unterteilt. Jede Tetrade entspricht einer Stelle der Hexadezimalzahl.

$$164_{10} = 1010|0100_2 = A4_{16}$$

The conversion of hexadecimal numbers to binary numbers is done analogously. One digit in the hexadecimal system corresponds to four digits in the binary system.

Table 2.3 contains an overview of the various representations of the first 16 positive natural numbers in the decimal system, the binary system, the octal system, and the hexadecimal system.

Die Umwandlung von Hexadezimalzahlen in Dualzahlen geschieht analog. Eine Stelle im Hexadezimalsystem entspricht vier Stellen im Dualsystem.

Tabelle 2.3 enthält eine Übersicht der verschiedenen Darstellungen der ersten 16 positiven natürlichen Zahlen im Dezimalsystem, Dualsystem, Oktalsystem und Hexadezimalsystem.

2.3

File and Storage Dimensions

For performance reasons, computers usually do not carry out read and write operations on single bits, but instead, work with bit sequences whose lengths are multiples of eight. A group of 8 bits is called *byte*. The value of a byte can be

Datei- und Speichergrößen

Rechner lesen und schreiben aus Geschwindigkeitsgründen meist nicht einzelne Bits, sondern arbeiten mit Bitfolgen, deren Längen Vielfache von Acht sind. Eine Gruppe von 8 Bits nennt man *Byte*. Den Wert eines Bytes kann man ent-

represented either by 8 bits or two hexadecimal digits.

A file is an arbitrarily long sequence of bytes that contain related data. All information (numbers, texts, music, programs, ...) a computer should work with must to be represented as a sequence of bytes and stored as a file [5].

Since most files have a size of several thousand or millions of bytes, different size units for shortened number representation exist. For data storage with binary addressing, storage capacities of 2^n bytes (powers of two) are used (see Table 2.4).

weder durch 8 Bits oder zwei Hexadezimalziffern darstellen.

Eine Datei ist eine beliebig lange Folge von Bytes und enthält inhaltlich zusammengehörende Daten. Alle Informationen (Zahlen, Texte, Musik, Programme, usw.), mit denen ein Rechner arbeiten soll, müssen als Folge von Bytes repräsentiert werden können und als Datei gespeichert werden [5].

Da sich die Größenordnungen der meisten Dateien im Bereich mehrerer Tausend oder Millionen Bytes befinden, gibt es verschiedene Größeneinheiten zur verkürzten Zahlendarstellung. Für Datenspeicher mit binärer Adressierung ergeben sich Speicherkapazitäten von 2^n Byte, also Zweierpotenzen (siehe Tabelle 2.4).

Table 2.4: File and Memory Sizes

Name	Symbol	Meaning
Kilobyte	kB	$2^{10} = 1.024$ bytes
Megabyte	MB	$2^{20} = 1.048.576$ bytes
Gigabyte	GB	$2^{30} = 1.073.741.824$ bytes
Terabyte	TB	$2^{40} = 1.099.511.627.776$ bytes
Petabyte	PB	$2^{50} = 1.125.899.906.842.624$ bytes
Exabyte	EB	$2^{60} = 1.152.921.504.606.846.976$ bytes
Zettabyte	ZB	$2^{70} = 1.180.591.620.717.411.303.424$ bytes
Yottabyte	YB	$2^{80} = 1.208.925.819.614.629.174.706.176$ bytes

The units in Table 2.4 are typically used by operating systems to specify the storage capacities of main memory and storage devices. However, the manufacturers of hard disk drives, CD/DVDs and USB storage drives prefer using decimal prefixes for the calculation of the storage capacity and its specification on the packaging. This means, for example, 10^9 instead of 2^{30} for GB is used, and 10^{12} instead of 2^{40} for TB. For this reason, for example, when using a DVD-R disc with a specified capacity of 4.7 GB, several applications will show its correct capacity of only 4.38 GB.

Die Maßeinheiten in Tabelle 2.4 haben sich für die Größenangabe von Hauptspeicher und Speichermedien in Betriebssystemen eingebürgert. Die Hersteller von Festplatten, CD/DVDs und USB-Speichermedien bevorzugen zur Berechnung der Kapazität und zur Angabe auf der Verpackung aber lieber Dezimal-Präfixe, also zum Beispiel den Faktor 10^9 anstatt 2^{30} für GB und 10^{12} anstatt 2^{40} für TB. Aus diesem Grund wird zum Beispiel bei einem DVD-Rohling mit einer angegebenen Kapazität von 4,7 GB in vielen Anwendungen korrekterweise nur die Kapazität 4,38 GB angezeigt.

$$10^9 = 1,000,000,000\,, \quad 2^{30} = 1,073,741,824$$

In this case, the difference in capacity between the power of two and the power of ten is approximately 7.37 %.

For larger storage systems, the difference is even more significant. For example, of a hard

Der Kapazitätsunterschied zwischen Zweierpotenz und Zehnerpotenz ist in diesem Fall ca. 7,37 %.

Bei größeren Speichern ist der Unterschied noch größer. So können von einer Festplatte mit

disk drive with a propagated storage capacity of 1 TB, only about 930 GB can be used.

angeblich 1 TB Speicherkapazität tatsächlich nur etwa 930 GB verwendet werden.

$$10^{12} = 1,000,000,000,000\,, \quad 2^{40} = 1,099,511,627,776$$

The difference in capacity between the power of two and the power of ten, in this case, is already approximately 9.95 % and with each further unit of measurement (PB, EB, ZB, ...) the difference in capacity between the power of two and the power of ten grows.

The International Electrotechnical Commission (IEC) proposed in 1996 to label the popular size factors that base on powers of two with the lowercase letter "i" and to reserve the established designations for the powers of 10. This proposal did not succeed so far, and the alternative names Kibibyte (KiB), Mebibyte (MiB), Gibibyte (GiB), Tebibyte (TiB), Pebibyte (PiB), Exbibyte (EIB) and Zebibyte (ZiB) did not gain much popularity outside the academic world.

Der Kapazitätsunterschied zwischen Zweierpotenz und Zehnerpotenz ist in diesem Fall schon ca. 9,95 % und mit jeder weiteren Maßeinheit (PB, EB, ZB, usw.) wächst der Kapazitätsunterschied zwischen Zweierpotenzen und Zehnerpotenzen weiter.

Die International Electrotechnical Commission (IEC) schlug 1996 vor, die populären Größenfaktoren, die auf den Zweierpotenzen basieren, mit einem kleinen „i“ zu kennzeichnen und die etablierten Bezeichnungen der Maßeinheiten für die Zehnerpotenzen zu reservieren. Dieser Vorschlag konnte sich bislang nicht durchsetzen und die daraus resultierenden alternativen Bezeichnungen Kibibyte (KiB), Mebibyte (MiB), Gibibyte (GiB), Tebibyte (TiB), Pebibyte (PiB), Exbibyte (EiB) und Zebibyte (ZiB) sind außerhalb des akademischen Bereichs nicht besonders populär.

2.4

Information Representation

Data is represented as sequences of zeroes and ones that encode arbitrary information. In order to represent text and numbers with data, the characters of the alphabet (uppercase and lowercase), punctuation marks such as period, comma and semicolon, as well as some special characters like `+`, `%`, `&` and `$` are encoded as bit sequences. Special characters such as space (`SP`), carriage return (`CR`) and tabulator (`TAB`) are also required. The most established encoding is the *American Standard Code for Information Interchange (ASCII)*.

Informationsdarstellung

Daten sind Folgen von Nullen und Einsen, die beliebige Informationen repräsentieren. Um Text und Zahlen durch Daten darzustellen, kodiert man die Zeichen des Alphabets (Groß- und Kleinschreibung), Satzzeichen wie Punkt, Komma und Semikolon, sowie einige Spezialzeichen wie zum Beispiel `+`, `%`, `&` und `$` in Bitfolgen. Zudem sind Sonderzeichen wie Leerzeichen (`SP`), Wagenrücklauf (`CR`) und Tabulator (`TAB`) nötig. Die am besten etablierte Kodierung ist der *American Standard Code for Information Interchange (ASCII)*.

2.4.1

ASCII Encoding

The ASCII encoding, also called US-ASCII, is a 7-bit character encoding. I.e., each character is assigned to a bit sequence of 7 bits. So there are $2^7 = 128$ different bit sequences, which is

ASCII-Kodierung

Die ASCII-Kodierung, häufig auch US-ASCII genannt, ist eine 7-Bit-Zeichenkodierung. Das heißt, dass jedem Zeichen eine Bitfolge aus 7 Bits zugeordnet ist. Es existieren also $2^7 = 128$

also the number of characters defined by this character encoding (see Table 2.5). Of these 128 characters, 95 characters are printable and 33 characters are not. The printable characters are (starting with space):

verschiedene Bitfolgen und exakt so viele Zeichen definiert die Zeichenkodierung (siehe Tabelle 2.5). Von den 128 Zeichen sind 95 Zeichen druckbar und 33 Zeichen nicht druckbar. Die druckbaren Zeichen sind (beginnend mit dem Leerzeichen):

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

Table 2.5: The ASCII character encoding (US-ASCII)

Dez	Hex	Char.	Dez	Hex	Char.	Dez	Hex	Char.	Dez	Hex	Char.
000	00	NUL	032	20	Space	064	40	@	096	60	`
001	01	SOH	033	21	!	065	41	A	097	61	a
002	02	STX	034	22	"	066	42	B	098	62	b
003	03	ETX	035	23	#	067	43	C	099	63	c
004	04	EOT	036	24	$	068	44	D	100	64	d
005	05	ENQ	037	25	%	069	45	E	101	65	e
006	06	ACK	038	26	&	070	46	F	102	66	f
007	07	BEL	039	27	'	071	47	G	103	67	g
008	08	BS	040	28	(	072	48	H	104	68	h
009	09	TAB	041	29	)	073	49	I	105	69	i
010	0A	LF	042	2A	*	074	4A	J	106	6A	j
011	0B	VT	043	2B	+	075	4B	K	107	6B	k
012	0C	FF	044	2C	,	076	4C	L	108	6C	l
013	0D	CR	045	2D	-	077	4D	M	109	6D	m
014	0E	SO	046	2E	.	078	4E	N	110	6E	n
015	0F	SI	047	2F	/	079	4F	O	111	6F	o
016	10	DLE	048	30	0	080	50	P	112	70	p
017	11	DC1	049	31	1	081	51	Q	113	71	q
018	12	DC2	050	32	2	082	52	R	114	72	r
019	13	DC3	051	33	3	083	53	S	115	73	s
020	14	DC4	052	34	4	084	54	T	116	74	t
021	15	NAK	053	35	5	085	55	U	117	75	u
022	16	SYN	054	36	6	086	56	V	118	76	v
023	17	ETB	055	37	7	087	57	W	119	77	w
024	18	CAN	056	38	8	088	58	X	120	78	x
025	19	EM	057	39	9	089	59	Y	121	79	y
026	1A	SUB	058	3A	:	090	5A	Z	122	7A	z
027	1B	ESC	059	3B	;	091	5B	[	123	7B	{
028	1C	FS	060	3C	<	092	5C	\	124	7C	|
029	1D	GS	061	3D	=	093	5D	]	125	7D	}
030	1E	RS	062	3E	>	094	5E	^	126	7E	~
031	1F	US	063	3F	?	095	5F	_	127	7F	DEL

The non-printable characters 00_{16} to 20_{16} and $7F_{16}$, for example, backspace (BS) and carriage return (CR), are control characters originally

Die nicht druckbaren Zeichen 00_{16} bis 20_{16} und $7F_{16}$, zum Beispiel Backspace (BS) und Carriage Return (CR), sind Steuerzeichen, die

used to control a teleprinter. ASCII is therefore not only a standard for storing data, but also suitable for data transmission. The beginning and the end of a data transmission are highlighted with Start of Text (`STX`) and End of Text (`ETX`). The transmission can be controlled with non-printable characters such as Acknowledge (`ACK`) and Negative Acknowledge (`NAK`). With Bell (`BEL`), a sender can transmit an alarm signal to the receiver, e.g., in the event of an error.

The 8th bit can be used as a parity bit for error detection. In this case it has a value of 0, if the number of one-bits at the remaining seven bit positions is even. Otherwise, it has a value of 1.

Due to improved protocols for data transmission, the 8th bit is no longer required for error detection when transmitting ASCII-encoded texts. Therefore, in order to be able to encode additional characters, US-ASCII was upgraded with numerous extensions and became an 8-bit character encoding. If each character gets assigned a sequence of 8 bits, $2^8 = 256$ different bit sequences are available. This resulted in 128 additional characters in comparison to the standard US-ASCII. Since these 128 additional characters are not sufficient to encode all internationally required special characters, several ASCII extensions for different languages and regions exist.

These extensions are compatible with the original US-ASCII. All characters specified in US-ASCII are encoded in the different extensions by the same bit sequences. The first 128 characters of all ASCII extensions are therefore identical with the original ASCII table. Extensions such as ISO Latin 9 (ISO 8859-15) contain language-specific characters (e.g., German umlauts) and special characters (e.g., the Euro symbol €), which are not included in the standard Latin alphabet.

One disadvantage of these ASCII extensions is that not all extensions are available in all operating systems. If two communication partners do not use the identical extension, the special characters in the text are displayed incorrectly.

ursprünglich zur Steuerung eines Fernschreibers verwendet wurden. ASCII ist also nicht nur ein Standard zur Datenablage, sondern auch zur Datenübertragung geeignet. Den Anfang und das Ende einer Datenübertragung markiert man mit Start of Text (`STX`) bzw. End of Text (`ETX`). Die Steuerung der Übertragung ist mit nicht druckbaren Zeichen wie Acknowledge (`ACK`) und negative Acknowledge (`NAK`) möglich. Mit Bell (`BEL`) kann ein Sender, zum Beispiel bei einem Fehler, ein Alarmsignal an den Empfänger senden.

Das 8. Bit kann als Paritätsbit zur Fehlererkennung verwendet werden. In diesem Fall hat es den Wert 0, wenn die Anzahl der Einsen an den übrigen sieben Bitpositionen gerade ist und ansonsten den Wert 1.

Durch verbesserte Protokolle zur Datenübertragung benötigt man das 8. Bit bei der Datenübertragung von ASCII-kodierten Texten nicht mehr zur Fehlererkennung. Darum wurde, um zusätzliche Zeichen kodieren zu können, US-ASCII mit zahlreichen Erweiterungen zu einer 8-Bit-Zeichenkodierung erweitert. Wird jedem Zeichen eine Bitfolge aus 8 Bits zugeordnet, sind $2^8 = 256$ verschiedene Bitfolgen verfügbar. Es sind also 128 Zeichen mehr verfügbar, als bei US-ASCII. Da diese 128 zusätzlichen Zeichen nicht ausreichen, um alle international benötigten Sonderzeichen zu kodieren, existieren verschiedene ASCII-Erweiterungen für die verschiedenen Sprachen und Regionen.

Die Erweiterungen sind mit dem ursprünglichen US-ASCII kompatibel. Alle im US-ASCII definierten Zeichen werden in den verschiedenen Erweiterungen durch die gleichen Bitfolgen kodiert. Die ersten 128 Zeichen einer ASCII-Erweiterung sind also mit der ursprünglichen ASCII-Tabelle identisch. Die Erweiterungen wie zum Beispiel ISO Latin 9 (ISO 8859-15) enthalten sprachspezifische Zeichen (zum Beispiel Umlaute) und Sonderzeichen (zum Beispiel das Euro-Symbol €), die nicht im lateinischen Grundalphabet enthalten sind.

Ein Nachteil der ASCII-Erweiterungen ist, dass nicht in allen Betriebssystemen alle Erweiterungen verfügbar sind. Wenn zwei Kommunikationspartner nicht die identische Erweiterung verwenden, werden unter anderem die Sonderzeichen im Text falsch angezeigt.

2.5

Unicode

In order to avoid problems caused by different character encodings, the multi-byte encoding Unicode (ISO 10646) was developed. It is extended continuously and is supposed to contain all known characters in the future.

Several Unicode standards exist. The most popular one is UTF-8. The first 128 characters are encoded with a single byte and are identical to US-ASCII. The encodings of the other characters use between 2 and 6 bytes. Currently, UTF-8 contains over 100,000 characters.

With UTF-8, each byte beginning with a `0`-bit corresponds to a 7-bit US-ASCII character. Each byte beginning with a `1`-bit belongs to a character encoded with multiple bytes. If a character that is encoded with UTF-8 consists of $n \geq 2$ bytes, the first byte begins with n `1`-bits and each of the following $n-1$ bytes begins with the bit sequence `10` (see Table 2.6).

Unicode

Um die Probleme durch unterschiedliche Zeichenkodierungen zu vermeiden, wurde die Mehrbyte-Kodierung Unicode (ISO 10646) entwickelt. Diese wird laufend erweitert und soll in Zukunft alle bekannten Schriftzeichen enthalten.

Es existieren verschiedene Unicode-Standards. Am populärsten ist UTF-8. Die ersten 128 Zeichen werden mit einem Byte codiert und sind mit US-ASCII identisch. Die Kodierungen der anderen Zeichen verwenden zwischen 2 und 6 Bytes. Aktuell enthält UTF-8 über 100.000 Zeichen.

Bei UTF-8 entspricht jedes mit `0` beginnende Byte einem 7-Bit US-ASCII-Zeichen. Jedes mit `1` beginnende Byte gehört zu einem aus mehreren Bytes kodierten Zeichen. Besteht ein mit UTF-8 kodiertes Zeichen aus $n \geq 2$ Bytes, beginnt das erste Byte mit n Einsen und jedes der $n-1$ folgenden Bytes mit der Bitfolge `10` (siehe Tabelle 2.6).

Table 2.6: Multi-byte character encoding with UTF-8

Code length	Bits for the character encoding	Format
1 byte	7 bits	`0xxxxxxx`
2 bytes	11 bits	`110xxxxx 10xxxxxx`
3 bytes	16 bits	`1110xxxx 10xxxxxx 10xxxxxx`
4 bytes	21 bits	`11110xxx 10xxxxxx 10xxxxxx 10xxxxxx`
5 bytes	26 bits	`111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx`
6 bytes	31 bits	`1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx`

2.6

Representation of Strings

To encode continuous text, the individual characters are concatenated to a string. The text "`Netzwerke kompakt.`" becomes the following string.

Darstellung von Zeichenketten

Um einen fortlaufenden Text zu kodieren, werden die einzelnen Zeichen zu einer Zeichenkette (*String*) aneinandergefügt. Der Text „`Netzwerke kompakt.`“ wird zur nachstehenden Zeichenfolge.

```
N, e, t, z, w, e, r, k, e,
 , k, o, m, p, a, k, t, .
```

All characters (including the space) are replaced by the decimal character numbers of the ASCII table.

Alle Zeichen (auch das Leerzeichen) werden durch die dezimalen Zeichennummern der ASCII-Tabelle ersetzt.

```
078 101 116 122 119 101 114 107 101
032 107 111 109 112 097 107 116 046
```

Alternatively, the hexadecimal character numbers of the ASCII table can be specified.

Alternativ kann man die hexadezimalen Zeichennummern der ASCII-Tabelle angeben.

```
4E 65 74 7A 77 65 72 6B 65
20 6B 6F 6D 70 61 6B 74 2E
```

The conversion of the characters to binary numbers results in their representation as bit sequence.

Konvertiert man die Zeichennummern in Dualzahlen, erhält man die Repräsentation als Bitfolge.

```
01001110 01100101 01110100 01111010
01110111 01100101 01110010 01101011
01100101 00100000 01101011 01101111
01101101 01110000 01100001 01101011
01110100 00101110
```

3

Fundamentals of Computer Networking

Grundlagen der Computervernetzung

This chapter deals with fundamental terms of computer networks. It includes the classification of networks according to their maximum physical dimension and topologies, as well as aspects of data transmission and various network devices.

Dieses Kapitel behandelt grundlegende Begriffe der Computervernetzung. Dazu gehört die Einteilung der Netze anhand ihrer räumlichen Ausdehnung und Topologien sowie Aspekte der Datenübertragung und die verschiedenen Netzwerkgeräte.

3.1

History of Computer Networks

Entwicklung der Computernetze

When studying the generations of computer systems whose temporal boundaries are less static than Table 3.1 suggests, it becomes evident why the first developments in the field of computer networks did not occur until the 1960s and 1970s. Computers of the first generation were inefficient and not used for universal purposes but only for particular tasks such as flight path calculations.

The first decentralized computer network was the Advanced Research Projects Agency Network (Arpanet), which was developed in the 1960s and connected four universities in the United States by December 1969. The focus of Arpanet was on robustness against the failure of individual nodes. This goal was achieved by its decentralized structure and packet switching. In the following years, several universities and research institutes in the United States joined Arpanet. Numerous application protocols like

Betrachtet man die Generationen von Computersystemen, deren zeitliche Grenzen unschärfer sind, als es Tabelle 3.1 vermuten lässt, wird deutlich, warum es erst in den 1960er und 1970er Jahren erste Entwicklungen im Bereich Computernetze gab. Die Rechner der ersten Generation waren leistungsschwach und wurden nicht für universelle Zwecke, sondern nur für spezielle Aufgaben wie zum Beispiel Flugbahnberechnungen verwendet.

Das erste dezentrale Computernetz war das Advanced Research Projects Agency Network (Arpanet), das in den 1960er Jahren entwickelt wurde und bis Dezember 1969 vier amerikanische Hochschulen verband. Der Fokus des Arpanet war Robustheit gegenüber dem Ausfall einzelner Knoten. Dieses Ziel wurde durch die dezentrale Struktur und Paketvermittlung erreicht. In den folgenden Jahren schlossen sich zahlreiche amerikanische Universitäten und Forschungsinstitute dem Arpanet an. Im Rahmen

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_3

Telnet and FTP were also developed during the Arpanet project.

It is popular belief that the fear of a nuclear war motivated the construction of the Arpnet. However, this remains an unproven myth. Arpanet aimed to set up a robust computer network in order to make better use of the then limited computer resources of the associated universities. Especially the unreliable telephone lines, which served as transmission medium (solution: packet switching) and the desire for scalability (solution: decentralized structure) have influenced the development.

From the 1970s, computer networks became available for connecting terminals to mainframes via serial lines. This technology made time-sharing possible. In time-sharing mode, multiple users work at the same time and in a competitive way via terminals on a single computer by sharing the available computing time of the processor. The distribution of the computing time is done via *time slices*.

One example of a networking technology from this era is IBM's Systems Network Architecture (SNA), which was available as from 1974. From 1975 on, DECnet was also available, which allowed the construction of a computer network of two directly connected PDP-11 computers and, from 1976 on it was possible to connect up to 32 computers. While these early computer networks were proprietary technologies, later networks mostly based on open standards and protocols.

des Arpanet wurden auch zahlreiche Anwendungsprotokolle wie Telnet und FTP entwickelt.

Dass die Angst vor den Folgen eines Atomkriegs der Grund für den Aufbau des Arpanet war, ist eine gerne erzählte Geschichte. Dabei handelt es sich aber um einen unbewiesenen Mythos. Das Ziel des Arpanet war der Aufbau eines robusten Computernetzes, um die damals knappen Rechenkapazitäten der verbundenen Hochschulen besser nutzbar zu machen. Besonders die Unzulänglichkeiten der Telefonleitungen, die als Übertragungsmedium dienten (Lösung: Paketvermittlung) und der Wunsch nach Skalierbarkeit (Lösung: dezentrale Struktur) haben die Entwicklung beeinflusst.

Ab den 1970er Jahren wurden Netzwerke verfügbar, um Terminals mit Großrechnern über serielle Leitungen zu verbinden und so Dialogbetrieb zu ermöglichen. Beim Dialogbetrieb (*Time Sharing*) arbeiten mehrere Benutzer über Dialogstationen (*Teminals*) an einem Computer gleichzeitig und konkurrierend, indem sie sich die verfügbare Rechenzeit des Hauptprozessors teilen. Die Verteilung der Rechenzeit geschieht mit Zeitscheiben (*Time Slices*).

Ein Beispiel für eine Netzwerktechnologie aus dieser Epoche ist die ab 1974 verfügbare Systems Network Architecture (SNA) von IBM. Ab 1975 war auch DECnet verfügbar, das die Vernetzung von zwei direktverbundenen PDP-11-Rechnern und ab 1976 von bis zu 32 Rechnern ermöglichte. Handelte es sich bei diesen frühen Computernetzen um proprietäre Technologien, basierten spätere Netzwerke überwiegend auf offenen Standards und Protokollen.

Table 3.1: Generations of Computer Systems

	Timeframe	Technological progress
0	until 1940	(Electro-)mechanical calculating machines
1	1940 – 1955	Elelectron tubes, relays, jack panels
2	1955 – 1965	Transistors, batch processing
3	1965 – 1980	Integrated circuits, time sharing
4	1980 – 2000	Very large-scale integration, PCs/Workstations
5	2000 until ?	Distributed systems, *the network is the computer*

The rise of Personal Computers (PCs) and workstations in the 1980s and the 1990s and the availability of low-cost network technologies,

Das Aufkommen von Personal Computern (PCs) und Workstations in den 1980er und 1990er Jahren sowie die Verfügbarkeit preisgüns-

such as Ethernet and Token Ring caused the number of computer networks in universities and companies to grow.

In the 1980s, Arpanet switched over to the protocols TCP/IP and became part of the Internet. It was finally shut down in the late 1980s.

From the mid-1990s, low-cost Internet access and mobile systems such as notebooks, tablet computers, and mobile phones also led to the establishment of computer networks in private environments. Increasing computing power and storage capacity, as well as powerful and affordable networking technologies such as WLAN (WiFi), LTE (4G) and 5G, ensure that computer networks will remain a hot topic in the future.

tiger Netzwerktechnologien wie Ethernet und Token Ring führten zur zunehmenden Vernetzung in Hochschulen und Unternehmen.

Das Arpanet wurde in den 1980er Jahren durch den Wechsel hin zu den Protokollen TCP/IP ein Teil des Internet und schließlich Ende der 1980er Jahre abgeschaltet.

Ab Mitte der 1990er Jahre führten preisgünstige Internetverbindungen und mobile Systeme wie Notebooks, Tablet-Computer und Mobiltelefone auch dazu, dass sich Computernetze im privaten Umfeld etablieren konnten. Steigende Rechenleistung und Speicherkapazität sowie leistungsfähige und preisgünstige Vernetzungstechnologien wie WLAN (WiFi), LTE (4G) und 5G machen Computernetze auch in Zukunft zu einem aktuellen Thema.

3.2

Mandatory Components of Computer Networks

The construction and operation of a computer network requires at least two terminal devices with at least one *network service* for communication or shared resources usage, a *transmission medium* for data exchange and *network protocols*.

Common transmission media for wired networks are electrical conductors with copper wires (twisted pair cables or coaxial cables) and fiber-optic cables. Also, wireless transmission is possible.

Network protocols are rules to specify how network devices communicate with each other. These rules are mandatory for the communication partners to understand each other. To illustrate this, it is useful to imagine a telephone call to a foreign country. A connection is established, but no participant understands the other's language. Only if both communication partners speak the same language, they can communicate with each other [23].

Zwingend nötige Elemente für Computernetzwerke

Für den Aufbau und Betrieb eines Computernetzes sind mindestens zwei Endgeräte mit mindestens einem *Netzwerkdienst* (Service) zur Kommunikation oder gemeinsamen Ressourcennutzung, ein *Übertragungsmedium* zum Datenaustausch und *Netzwerkprotokolle* nötig.

Gängige Übertragungsmedien für leitungsgebundene Netze sind elektrische Leiter mit Kupferleitungen (Twisted-Pair-Kabel oder Koaxialkabel) und Lichtwellenleiter. Zudem ist auch nicht-leitungsgebundene (drahtlose) Übertragung möglich.

Netzwerkprotokolle sind Regeln, die festlegen, wie die Netzwerkgeräte miteinander kommunizieren können. Diese Regeln sind zwingend nötig, denn ansonsten können sich die Kommunikationspartner nicht verstehen. Zur Verdeutlichung kann man sich einen Telefonanruf ins Ausland vorstellen, bei der die Verbindung zustande kommt, aber kein Teilnehmer die Sprache des anderen versteht. Nur wenn beide Kommunikationspartner die gleiche Sprache sprechen, kommt eine Kommunikation zustande [23].

3.3

Computer Networks distinguished by their Dimensions

Depending on their dimensions, a distinction is made between different groups of computer networks.

Networks of small mobile devices such as cell phones are called *Personal Area Network* (PAN). These networks can be cable-based (USB or FireWire for example) or wireless (for example WLAN, Bluetooth or IrDA). Their maximum expansion is just a few meters.

A local network – *Local Area Network* (LAN) – stretches across an apartment, a building, a company site or a university campus. Cable-based LANs today mostly base on variants of the Ethernet standard (IEEE 802.3). Token Ring (IEEE 802.5) has also been popular until the 1990s. Local radio networks are called Wireless LAN (WLAN) and usually base on standards of the IEEE 802.11 family of standards. Depending on the transmission medium used, their maximum range is 500 to 1,000 m. Especially in wireless networks, the environment and the transmission power have a substantial impact on the maximum range.

A *Metropolitan Area Network* (MAN) connects LANs. Due to their low attenuation and high data transfer rate, fiber-optic cables are preferred here as transmission medium. For wireless networks, the Wireless Metropolitan Area Networks (WiMAX) standard from the IEEE 802.16 family is relevant. MANs extend across an metropolitan area or a big city and have a maximum expansion of up to 100 km.

Networks, which cover large geographic areas within a nation or continent and connect MANs are called *Wide Area Network* (WAN). In the past, they were mostly based on the Asynchronous Transfer Mode (ATM) standard. Current installations often use Ethernet with a data rate of 10 Gbit/s. The maximum expansion of WANs is up to 1.000 km.

A *Global Area Network* (GAN) that connects WANs covers unlimited geographic distances. The Internet is an example of a GAN, but not

Räumliche Ausdehnung von Computernetzen

Abhängig von der räumlichen Ausdehnung unterscheidet man verschiedene Gruppen von Computernetzen.

Netze aus mobilen Kleingeräten wie Mobiltelefonen heißen *Personal Area Network* (PAN). Diese können kabelgebunden (zum Beispiel via USB oder FireWire) sein oder Funknetze (zum Beispiel via WLAN, Bluetooth oder IrDA). Die Reichweite beträgt nur wenige Meter.

Ein lokales Netz – *Local Area Network* (LAN) – erstreckt sich über eine Wohnung, ein Gebäude, ein Firmengelände oder einen Campus. Kabelgebundene LANs basieren heute meist auf Varianten des Ethernet-Standards (IEEE 802.3). Bis in die 1990er Jahre war auch Token Ring (IEEE 802.5) populär. Lokale Funknetze heißen Wireless LAN (WLAN) und basieren in der Regel auf Standards der Normenfamilie IEEE 802.11. Je nach verwendetem Übertragungsmedium beträgt die maximale Ausdehnung 500 bis 1.000 m. Speziell bei Funknetzen spielen die räumlichen Gegebenheiten und Sendeleistung im Bezug auf die maximale Ausdehnung eine große Rolle.

Ein *Metropolitan Area Network* (MAN) verbindet LANs. Als Übertragungsmedium werden wegen der geringen Dämpfung und hohen Datentransferrate meist Lichtwellenleiter verwendet. Für Funknetze existiert in der Normenfamilie IEEE 802.16 der Standard Wireless Metropolitan Area Networks (WiMAX). MANs erstrecken sich über ein Ballungsgebiet oder eine Großstadt und haben eine Ausdehnung bis zu 100 km.

Netze, die große geografische Bereiche innerhalb einer Nation oder eines Kontinents abdecken und MANs verbinden, heißen *Wide Area Network* (WAN). Sie basierten in der Vergangenheit zumeist auf dem Standard Asynchronous Transfer Mode (ATM). Aktuelle Installationen verwenden häufig Ethernet mit einer Datendurchsatzrate von 10 Gbit/s. Die Ausdehnung von WANs ist bis zu 1.000 km.

Über unbegrenzte geographische Entfernungen erstreckt sich ein *Global Area Network* (GAN), das WANs verbindet. Das Internet ist

every GAN is the Internet. The connection of globally distributed offices of a company is another example of a GAN. Satellites or fiber-optic cables are used for data transmission.

ein Beispiel für ein GAN, aber nicht jedes GAN ist das Internet. Die Vernetzung weltweit verteilter Niederlassungen eines Unternehmens ist ein Beispiel für ein GAN. Zur Datenübertragung werden Satelliten oder Lichtwellenleiter eingesetzt.

3.4

Data Transmission

Communication between computers is possible via serial and parallel as well as via synchronous and asynchronous data transmission. Also, a distinction is made between simplex, duplex and half-duplex, depending on the directionality of the transmission.

Datenübertragung

Die Kommunikation zwischen Rechnern ist mit serieller und paralleler sowie synchroner und asynchroner Datenübertragung möglich. Zudem unterscheidet man abhängig von der Richtungsabhängigkeit der Übertragung Simplex, Duplex und Halbduplex.

3.4.1

Serial and Parallel Transmission

When *serial data transmission* is used, the bits are transmitted over a single line one after the other. The transfer of a single byte takes eight times as long compared to parallel data transmission. Application examples are local bus systems and connections in computer networks [11].

In addition to the control lines, multiple data lines exist for the transfer of the data bits, when *parallel data transmission* is used. One example is the parallel port for the connection of printers, which was replaced in recent years by the USB interface. Via a parallel interface, as many bits can be transmitted per time unit as there are data lines. If there are eight data lines, an entire byte of data can be transferred.

Another example is Parallel ATA (PATA). This parallel interface for connecting storage drives such as hard disk drives and CD/DVD drives can transfer two bytes of data per time unit. PATA has also been replaced in recent years by a serial successor, namely Serial ATA (SATA).

Parallel data transmission enables a higher data throughput but requires a larger number of lines. This is more cost-intensive for covering

Serielle und parallele Übertragung

Bei *serieller Datenübertragung* werden die Bits auf einer einzelnen Datenleitung nacheinander übertragen. Der Transfer eines Bytes dauert acht mal so lange wie bei paralleler Datenübertragung. Anwendungsbeispiele sind lokale Bus-Systeme und Verbindungen in Computernetzen [11].

Neben den Steuerleitungen sind bei *paralleler Datenübertragung* mehrere Datenleitungen zur Übertragung der Datenbits vorhanden. Ein Beispiel ist die parallele Schnittstelle zum Anschluss von Druckern, die in den vergangenen Jahren von der seriellen USB-Schnittstelle verdrängt wurde. Über eine parallele Schnittstelle können pro Zeiteinheit so viele Bits übertragen werden wie Datenleitungen vorhanden sind. Gibt es acht Datenleitungen, kann ein vollständiges Byte Daten übertragen werden.

Ein weiteres Beispiel ist Parallel ATA (PATA). Diese parallele Schnittstelle zum Anschluss von Speicherlaufwerken wie Festplatten und CD/DVD-Laufwerken kann pro Zeiteinheit zwei Bytes Daten übertragen. Auch PATA wurde in den vergangenen Jahren von einem seriellen Nachfolger, nämlich Serial ATA (SATA), abgelöst.

Parallele Datenübertragung ermöglicht einen höheren Datendurchsatz, benötigt dafür aber mehr Leitungen. Das macht die Überbrückung

long distances. Serial data transmission is also suitable for long distances.

großer Distanzen kostspieliger. Serielle Datenübertragung ist auch für große Distanzen geeignet.

3.4.2

Synchronous and Asynchronous Transmission

To read data from a bitstream, it must be sampled over a *time window*. This time window is produced by using a *clock source*. When using *synchronous data transmission*, the communication partners synchronize the transmission with a clock signal. This can be sent via a separate line, or it can be recovered from the data signal by the receiver. This process is called *clock recovery*. Due to the clock recovery, the data does not have to be resynchronized regularly. However, the implementation of clock recovery is complicated.

When *asynchronous data transmission* is used, the communication partners use independent clock sources. If data is transferred, a start bit is set before the data. This signals the receiver that it should start its clock source. At the end of the data, a stop bit follows, which terminates the data transmission. No synchronization of the clock sources is necessary. However, because the clock sources can differ from each other, the maximum size of the data that can be transmitted in one piece is limited. Also, the start bit and stop bit are overhead – in other words, an administrative effort or additional information that is not part of the payload.

In computer networks, the sender sets the clock and delivers it with the data stream. Therefore, synchronous data transmission always takes place. However, a distinction is made between synchronous and asynchronous computer networks. In synchronous computer networks, the connection between the communication partners is kept, and thus the connection is permanently synchronized. In asynchronous computer networks, there is no connection between the phases of data transfer. Therefore, the communication partners must synchronize themselves each time the connection is established.

Synchrone und asynchrone Übertragung

Um Daten aus einem Bitstrom auszulesen, muss dieser über ein *Zeitfenster* abgetastet werden. Dieses Zeitfenster wird über eine *Taktquelle* gewonnen. Bei *synchroner Datenübertragung* synchronisieren die Kommunikationspartner die Übertragung zeitlich mit einem Taktsignal. Dieses kann über eine eigene Leitung gesendet werden oder es wird vom Empfänger aus dem Datensignal zurückgewonnen. Dieser Vorgang heißt *Taktrückgewinnung*. Durch die Taktrückgewinnung müssen die Daten nicht regelmäßig neu synchronisiert werden. Die Realisierung der Taktrückgewinnung ist aber aufwändig.

Bei *asynchroner Datenübertragung* verwenden die Kommunikationspartner unabhängige Taktquellen. Werden Daten übertragen, wird vor die Daten ein Startbit gesetzt. Dieses signalisiert dem Empfänger, dass er seine Taktquelle starten soll. Am Ende der Daten folgt ein Stopbit, das die Datenübertragung beendet. Es ist keine Synchronisation der Taktquellen nötig. Weil aber die Taktquellen voneinander abweichen können, ist die maximale Größe der am Stück übertragbaren Daten eingeschränkt. Zudem stellen Startbit und Stopbit einen Overhead dar – also einen Verwaltungsaufwand oder Zusatzinformation, die nicht Teil der Nutzdaten sind.

Bei Computernetzen gibt der Sender den Takt vor und liefert ihn mit dem Datenstrom. Es findet also immer synchrone Datenübertragung statt. Man unterscheidet aber dennoch synchrone und asynchrone Computernetze. Bei synchronen Computernetzen bleibt die Verbindung zwischen den Kommunikationspartnern bestehen und damit ist die Verbindung dauerhaft synchronisiert. Bei asynchronen Computernetzen besteht zwischen den Phasen des Datentransfers keine Verbindung. Die Kommunikationspartner müssen sich beim Verbindungsaufbau somit immer wieder neu synchronisieren.

3.4.3

Directional Dependence of Data Transmission

The terms simplex, duplex and half-duplex indicate the directional dependence of a data transmission. With *simplex*, the information transfer only works in one direction (unidirectional). At the end of the transmission, the communication channel can be used by another sender. Application examples are radio, traditional television without return channel and radio message receiver (pager).

In contrast to Simplex, the information transfer via *duplex* (Full-duplex) works in both directions simultaneously (bidirectional). Application examples of systems with duplex transmission are networks with twisted pair cables (see section 5.2.2), because these provide separate lines for sending and receiving.

When implementing *half-duplex*, the information transfer works in both directions (bidirectional), but not simultaneously. Examples of such communication systems are networks with fiber-optic cables or coaxial cables, because there is just a single line for sending and receiving. Another example are wireless networks with only a single channel.

Richtungsabhängigkeit der Übertragung

Die Richtungsabhängigkeit einer Datenübertragung geben die Begriffe Simplex, Duplex und Halbduplex an. Bei *Simplex* funktioniert der Informationstransfer nur in einer Richtung (unidirektional). Nach dem Ende der Übertragung kann der Kommunikationskanal von einem anderen Sender verwendet werden. Anwendungsbeispiele sind Radio, klassisches Fernsehen ohne Rückkanal und Funkmeldeempfänger (Pager).

Im Gegensatz zu Simplex funktioniert die Übertragung bei *Duplex* (Vollduplex) in beide Richtungen gleichzeitig (bidirektional). Beispiele für Systeme mit Duplex-Übertragung sind Netzwerke mit Twisted-Pair-Kabeln (siehe Abschnitt 5.2.2), denn diese bieten separate Leitungen zum Senden und Empfangen.

Bei *Halbduplex* (Wechselbetrieb) funktioniert der Informationstransfer in beide Richtungen (bidirektional), aber nicht gleichzeitig. Beispiel für solche Kommunikationssysteme sind Netzwerke auf Basis von Glasfaser- oder Koaxialkabeln (siehe die Abschnitte 5.2.1 und 5.2.3), denn hier gibt es nur eine Leitung für Senden und Empfangen. Ein weiteres Beispiel sind Funknetze mit nur einem Kanal.

3.5

Devices in Computer Networks

The most simple devices in computer networks are *Repeaters*. These extend the range of local area networks by amplifying the received electrical or optical signals that are attenuated in the transmission media and by removing noise and jitter (fluctuations in the accuracy of the clock signal). A Repeater forwards signals (bits) but does not analyze their meaning or validate their correctness. A Repeater for wired computer networks has only two interfaces (*ports*).

Hubs are multiport Repeaters. They implement the same functionality as Repeaters, but have more than two ports. A Hub only forwards incoming signals to all its other ports. Since

Geräte in Computernetzen

Die einfachsten Geräte in Computernetzen sind *Repeater*. Diese vergrößern die Reichweite lokaler Netze, indem sie die elektrischen oder optischen Signale, die im Übertragungsmedien abgeschwächt werden, verstärken und vom Rauschen und Jitter (Genauigkeitsschwankungen im Übertragungstakt) reinigen. Ein Repeater leitet Signale (Bits) zwar weiter, analysiert aber nicht deren Bedeutung oder untersucht deren Korrektheit. Ein Repeater für kabelgebundene Computernetze verfügt lediglich über zwei Schnittstellen (*Ports*).

Hubs sind Multiport-Repeater, denn sie haben die gleiche Funktionalität wie Repeater, bieten aber mehr als zwei Schnittstellen. Ein Hub leitet eintreffende Signale einfach zu allen Ausgangs-

Repeaters and Hubs only forward the signals and do not analyze them, they have no network addresses and operate – from the perspective of other network devices – in a transparent way.

If signals must be transmitted over long distances, a *Modem* can be used to modulate them onto a carrier frequency in the high-frequency range, and another Modem can recover them via demodulation. Examples include (A)DSL or cable modems for broadband Internet access and telephone or fax modems for narrowband Internet access.

Bridges are used to connect different physical networks. A Bridge forwards frames from one physical network to another but has only two ports. Modems and WLAN access points also belong to the Bridge category.

(Layer-2-)Switches are multiport Bridges. They have the same functionality as Bridges but have more than two ports. Bridges and Switches filter Frames. This means that they only forward frames when it makes sense to do so. Also, they examine frames with checksums for correctness. Since they do not actively participate in the communication, they do not need network addresses for themselves for filtering and forwarding the frames.

Large network infrastructures usually consist of several smaller networks with separate logical address ranges. *Routers* do the forwarding of the packets between logical networks. Also, Routers allow connecting the local area network (LAN) to a WAN.

Layer-3-Switches feature similar functionality as Routers. However, these are only used within local area networks to implement different logical address ranges, and they do not allow to connect to a WAN. Another typical functionality of Layer-3-Switches is the provision of virtual (logical) networks (see section 10.2).

Gateways (Protocol Translators) enable communication between networks that use different protocols and/or different sorts of addressing.

Firewalls are based on firewall software and restrict communication between different networks or from/to individual participants based

ports weiter. Da Repeater und Hubs die Signale nur weiterleiten und nicht analysieren, haben sie selbst keine Adressen und arbeiten für die übrigen Netzwerkgeräte transparent.

Sollen Signale über weite Strecken transportiert werden, kann ein *Modem* diese auf eine Trägerfrequenz im Hochfrequenzbereich aufmodulieren und ein anderes Modem diese durch demodulieren wieder zurückgewinnen. Beispiele sind (A)DSL- oder Kabelmodems für Breitbandinternetzugänge und Telefon- oder Faxmodems für Schmalbandinternetzugänge.

Den Zusammenschluss verschiedener physischer Netze ermöglichen *Bridges*. Eine Bridge leitet Datenrahmen von einem physischen Netz zum anderen, verfügt aber lediglich über zwei Schnittstellen. Auch Modems und WLAN-Basisstationen (Access Points) gehören zur Gruppe der Bridges.

(Layer-2-)Switche sind Multiport-Bridges. Sie haben die gleiche Funktionalität wie Bridges, verfügen aber über mehr als zwei Schnittstellen. Bridges und Switche filtern die Datenrahmen. Das heißt, dass sie Datenrahmen nur weiterleiten, wenn es sinnvoll ist. Zudem untersuchen sie die Datenrahmen mit Prüfsummen auf Korrektheit. Da sie nicht aktiv an der Kommunikation teilnehmen, brauchen sie zum Filtern und Weiterleiten der Rahmen selbst keine Adressen.

Große Netzwerkinfrastrukturen bestehen meist aus mehreren kleineren Netzen mit eigenen logischen Adressbereichen. Die Weiterleitung der Datenpakete zwischen logischen Netzen erledigen *Router*. Zudem ermöglichen Router die Verbindung des lokalen Netzes (LAN) mit einem WAN.

Eine ähnliche Funktionalität wie Router haben *Layer-3-Switche*. Diese werden aber nur innerhalb lokaler Netze verwendet, um verschiedene logischen Adressbereiche zu realisieren und sie ermöglichen keine Verbindung mit einem WAN. Eine weitere typische Funktionalität von Layer-3-Switchen ist die Bereitstellung virtueller (logischer) Netzwerke (siehe Abschnitt 10.2).

Gateways (Protokollumsetzer) ermöglichen die Kommunikation zwischen Netzen, die auf unterschiedlichen Protokollen basieren und/oder unterschiedliche Adressierung verwenden.

Firewalls basieren auf einer Firewall-Software und schränken anhand festgelegter Firewall-Regeln die Kommunikation zwischen unter-

on specified firewall rules. Possible information with which the firewall rules are specified are – among others – the IP addresses of sender and receiver, port numbers and network protocols. Personal firewalls or desktop firewalls are locally installed software that protects a single computer from access from the network. When using an external firewall, the firewall software is not deployed on the systems to be protected, but on a dedicated network device which separates the network to be protected from the public network.

The already mentioned include all sorts of devices that use computer networks for communication. Examples are PCs, mainframes, mobile phones, refrigerators, etc. All devices in computer networks are also called *nodes* and for radio networks *stations*.

An overview of the different devices in computer networks and their assignment to the layers of the hybrid reference model (see section 4.2) presents Table 3.2. The layers that are indicated in the table refer to the necessary functionalities of the devices. If devices such as cable modems also implement further functions such as a web interface for administration or a DHCP, SMTP or Telnet server, they also require protocols of the Network Layer (layer 3), Transport Layer (layer 4), and Application Layer (layer 5).

schiedlichen Netzen oder von/zu einzelnen Teilnehmern ein. Mögliche Informationen mit denen die Firewall-Regeln definiert werden sind u. a. die IP-Adressen von Sender und Empfänger, Portnummern und Netzwerkprotokolle. Bei Personal- bzw. Desktop-Firewalls handelt es sich lediglich um eine lokal installierte Software, die einen einzelnen Computer vor unerwünschten Zugriffen aus dem Netzwerk schützt. Bei einer externen Firewall befindet sich die Firewall-Software nicht auf den zu schützenden Systemen selbst, sondern auf einem separaten Netzwerkgerät, welches das zu schützende vom öffentlichen Netz trennt.

Die bereits erwähnten *Endgeräte* können beliebige Geräte sein, die Computernetze zur Kommunikation nutzen. Beispiele sind PCs, Großrechner, Mobiltelefone, Kühlschränke usw. Alle Geräte in Computernetzen werden auch *Knoten* (*Nodes*) und bei Funknetzen *Stationen* genannt.

Eine Übersicht über die verschiedenen Geräte in Computernetzen und deren Zuordnung zu den Schichten des hybriden Referenzmodells (siehe Abschnitt 4.2) enthält Tabelle 3.2. Die in der Tabelle angegeben Schichten beziehen sich auf die zwingend nötigen Funktionalitäten der Geräte. Bieten Geräte wie z. B. Kabelmodems auch weitere Funktionalitäten wie zum Beispiel eine Weboberfläche zur Administration oder einen DHCP-, SMTP- oder Telnet-Server, benötigen Sie auch Protokolle der Vermittlungsschicht (Schicht 3), Transportschicht (Schicht 4) und Anwendungsschicht (Schicht 5).

Table 3.2: Devices in Computer Networks and their Assignment to the Layers of the Hybrid Reference Model

Device	Layer(s)
Repeater	1
Multiport Repeater (Hub)	1
Modem[a]	1–2
Bridge	1–2
Multiport Bridge (Layer-2-Switch)	1–2
WLAN access point	1–2
Router	1–3
Layer-3-Switch	1–3
Gateway (Protocol Translator)	1–5
Firewall	1–5
Terminal Device (Node, Station)	1–5

[a] e. g. Telephone Modem, ISDN device, (A)DSL Modem, Cable Modem, 3G/4G/5G Modem, etc.

3.6

Topologies of Computer Networks

The topology of a computer network determines how the communication partners are connected. The *physical topology* describes the wiring, and the *logical topology* describes the flow of data between the terminal devices.

The physical and logical topologies of a computer network may differ. The structure of large-scale networks is often composed of several different topologies. The selection of the topology is crucial for the network's reliability. The representation of topologies is done graphically with nodes and edges (see Figure 3.1).

Topologien von Computernetzen

Die Topologie eines Computernetzes legt fest, wie die Kommunikationspartner miteinander verbunden sind. Die *physische Topologie* beschreibt den Aufbau der Netzverkabelung und die *logische Topologie* den Datenfluss zwischen Endgeräten.

Die physische und logische Topologie eines Computernetzes können sich unterscheiden. Die Struktur großer Netze ist oftmals aus mehreren unterschiedlichen Topologien zusammensetzt. Die Auswahl der Topologie ist entscheidend für die Ausfallsicherheit. Die Darstellung von Topologien erfolgt grafisch mit Knoten und Kanten (siehe Abbildung 3.1).

3.6.1

Bus Network

All nodes are connected via a shared transmission medium. There are no active components between the nodes and the medium. Therefore, the failure of individual nodes does not cause the failure of the network. The bus topology was popular until the mid-1990s as a topology for LANs, because central components such as Hubs or Switches were expensive at that time. The benefit of this topology is the low cost for constructing a network.

A drawback is that an interruption of the bus causes the failure of the network. Another disadvantage is that only a single participant can send data at a given point in time because otherwise, collisions will occur. A media access control method as CSMA/CD is necessary to control the access to the medium.

Network technologies that implement this topology are Thin Ethernet (10BASE2) and Thick Ethernet (10BASE5).

A more modern network technology based on the bus topology is *PowerLAN*, which is also called *Powerline Communication*. It uses the electric power grid as a shared transmission medium. The network devices work according to the operational principle of Modems. Signals are modulated onto one or more different car-

Bus-Topologie

Alle Knoten sind über ein Übertragungsmedium, den Bus, verbunden. Zwischen den Knoten und dem Medium befinden sich keine aktiven Komponenten. Der Ausfall einzelner Knoten führt darum auch nicht zum Ausfall des Netzes. Die Bus-Topologie war bis in Mitte der 1990er Jahre als Topologie für LANs populär, da Hubs oder Switches zu dieser Zeit kostspielig waren. Der Vorteil dieser Topologie sind die geringen Kosten für den Aufbau eines Netzes.

Nachteilig ist, dass eine Unterbrechung des Bus zum Ausfall des Netzes führt. Es kann auch zu jedem Zeitpunkt nur ein Teilnehmer Daten senden, da es sonst zu Kollisionen kommt. Ein Medienzugriffsverfahren wie CSMA/CD ist nötig, um den Zugriff auf das Medium zu regeln.

Netzwerktechnologien, die auf dieser Topologie basieren, sind Thin Ethernet (10BASE2) und Thick Ethernet (10BASE5).

Eine modernere Netzwerktechnologie, die auf der Bus-Topologie basiert, ist *PowerLAN*, das auch *Powerline Communication* genannt wird. Dabei wird das Stromnetz als gemeinsames Übertragungsmedium verwendet. Die Netzwerkgeräte arbeiten nach dem Prinzip von Modems. Signale werden auf eine oder mehrere unter-

Figure 3.1: Topologies of Computer Networks

rier frequencies in the high-frequency range and demodulated during the recovery.

schiedliche Trägerfrequenzen im Hochfrequenzbereich aufmoduliert und bei der Zurückgewinnung wieder demoduliert.

3.6.2

Ring Network

Two nodes are directly connected. The information to be transmitted is forwarded from node to node to the destination. An interruption of the ring causes the failure of the network.

Since each participant is also a signal amplifier (Repeater), depending on the transmission medium, considerable ring lengths of several hundred meters are possible.

Network technologies that implement this topology are Token Ring and the Fiber Dis-

Ring-Topologie

Jeweils zwei Knoten sind direkt miteinander verbunden. Die zu übertragende Information wird von Knoten zu Knoten bis zum Bestimmungsort weitergeleitet. Eine Unterbrechung des Rings führt zum Ausfall des Netzes.

Da jeder Teilnehmer gleichzeitig auch ein Signalverstärker (Repeater) ist, sind abhängig vom Übertragungsmedium große Ringlängen von mehreren hundert Metern möglich.

Netzwerktechnologien, die auf dieser Topologie basieren, sind Token Ring und das Fiber

tributed Data Interface (FDDI). FDDI implements two rings in order to have a redundant line in the event of an error.

Distributed Data Interface (FDDI). Bei FDDI ist der Ring doppelt ausgeführt, um für den Fehlerfall eine redundante Leitung vorzuhalten.

3.6.3

Star Network

All nodes are directly connected to a central component (Hub or Switch).

The advantages of this topology include good expandability and stability. The failure of a node does not affect the other nodes and does not cause a failure of the network. Disadvantages are the considerable effort for the cabling, and the dependence on the central component since the failure of this component leads to failure of the network. In order to reduce this risk, the central component can be implemented in a redundant way.

Examples of network technologies that use the star topology are Fast Ethernet, Fibre Channel, InfiniBand, and Token Ring.

Token Ring is an example for the fact that the physical and the logical topology may differ, because it uses the logical ring topology, but for reasons of reliability, it is usually wired as a physical star topology by using a ring concentrator device, the so-called Media Access Unit (MAU). A MAU is a *ring in a box*. If a node is not connected or has failed, the MAU bypasses this node, and the ring is not interrupted.

Stern-Topologie

Alle Knoten sind direkt an einer zentralen Komponente (Hub oder Switch) angeschlossen.

Vorteile dieser Topologie sind gute Erweiterbarkeit und Stabilität. Der Ausfall eines Knotens beeinflusst die anderen Knoten nicht und führt nicht zum Ausfall des Netzes. Nachteilig ist der hohe Aufwand für die Verkabelung und die Abhängigkeit von der zentralen Komponente, da der Ausfall dieser Komponente zum Ausfall des Netzes führt. Um dieses Risiko abzumildern, kann die zentrale Komponente redundant ausgelegt werden.

Beispiele für Netzwerktechnologien, die auf der Stern-Topologie basieren, sind Fast-Ethernet, Fibre Channel, InfiniBand und Token Ring.

Token Ring ist ein Beispiel dafür, dass sich die physische und die logische Topologie unterscheiden können, denn es basiert auf einer logischen Ring-Topologie, wird aber aus Gründen der Ausfallsicherheit mit Hilfe eines Ringleitungsverteilers (RLV), der sogenannten Media Access Unit (MAU), meist als physische Stern-Topologie verkabelt. Eine MAU ist ein *Ring in der Box*. Ist ein Knoten nicht angeschlossen oder ausgefallen, überbrückt die MAU den Knoten und der Ring ist nicht unterbrochen.

3.6.4

Mesh Network

In mesh networks, each node is connected to one or more other nodes. In a *fully connected mesh network*, each node is connected to each other.

If nodes or connections fail, communication is usually still possible via alternative routes. The advantage of this topology is the high level of reliability (depending on the cabling effort). Drawbacks are the high cabling effort and increased electric energy consumption. Also, in not fully connected mesh networks, it is compli-

Maschen-Topologie

In vermaschten Netzen ist jeder Teilnehmer mit einem oder mehreren anderen Teilnehmern verbunden. Ist das Netz *vollständig vermascht*, ist jeder Teilnehmer mit jedem anderen verbunden.

Fallen Teilnehmer oder Verbindungen aus, ist die Kommunikation im Regelfall über alternative Routen weiter möglich. Der Vorteil dieser Topologie ist die (abhängig vom Verkabelungsaufwand) hohe Ausfallsicherheit. Nachteilig sind der hohe Verkabelungsaufwand und erhöhte Energieverbrauch. Zudem ist in nicht vollständig vermaschten Netzen komplex im Zu-

cated to identify the best way to forward packets from sender to receiver.

Computer networks, which use the mesh topology are often ad-hoc (radio) networks. The logical topology between Routers also implements the mesh topology.

ge der Weiterleitung der Pakte vom Sender zum Empfänger den besten Weg zu ermitteln.

Computernetze, welche die Maschen-Topologie verwenden, sind häufig Ad-hoc-(Funk-)Netze. Auch die logische Topologie zwischen Routern entspricht der Maschen-Topologie.

3.6.5

Tree Network

In a tree topology, one or more edges originate from a root. Every edge leads to a leaf node or the roots of other trees. In a tree topology, several star topology networks are hierarchically connected.

The failure of a terminal device (leaf node) has no consequences on the functionality of the network. Also, the expandability is excellent, and long distances can be realized. Networks with this topology are well suited for search and sort algorithms. Drawbacks are that if a node or even the root fails, the entire (sub-)tree behind can no longer be reached. In a large tree, the root may become a bottleneck, because the communication from one half of the tree to the other always takes place via the root.

Examples of computer networks that implement the tree topology are connections between Hubs or Switches via an uplink (see Figure 5.15).

Baum-Topologie

In einer Baum-Topologie gehen von einer Wurzel eine oder mehrere Kanten aus. Jede Kante führt zu einem Blattknoten oder zu Wurzeln weiterer Bäume. Mehrere Netze der Sterntopologie sind in einer Baum-Topologie hierarchisch verbunden.

Der Ausfall eines Endgeräts (Blattknotens) hat keine Konsequenzen für die Funktionalität des Netzes. Zudem ist die Erweiterbarkeit gut und große Entfernungen sind realisierbar. Netze mit dieser Topologie eignen sich gut für Such- und Sortieralgorithmen. Nachteilig ist, dass beim Ausfall eines Knotens oder sogar der Wurzel der davon ausgehende (Unter-)Baum nicht mehr erreichbar ist. Bei einem großen Baum kann die Wurzel zum Engpass werden, da die Kommunikation von der einen Baumhälfte in die andere Hälfte immer über die Wurzel geht.

Beispiel für Computernetze mit der Baum-Topologie sind Verbindungen zwischen Hubs oder Switches via Uplink (siehe Abbildung 5.15).

3.6.6

Cellular Network

Wireless networks use this topology. A cell is the area in which nodes can communicate with the base station. The failure of a node does not effect the functionality of the network. A drawback is the limited range of the base stations, depending on their number and positions.

Examples of computer networks with the cellular topology are WLAN and Bluetooth.

Zellen-Topologie

Diese Topologie kommt bei drahtlosen Netzen zum Einsatz. Eine Zelle ist der Bereich, in dem Endgeräte mit der Basisstation kommunizieren können. Der Ausfall eines Endgeräts hat hat keinen Einfluss auf die Funktionalität des Netzes. Nachteilig ist die begrenzte Reichweite der Basisstationen, abhängig von deren Anzahl und Positionen.

Beispiele für Computernetze mit der Zellen-Topologie sind WLAN und Bluetooth.

3.7

Frequency and Data Signal

Electrical engineering distinguishes between the voltage types *direct current*, where the level and polarity of the voltage always remains the same, and *alternating current*, where the level and polarity change periodically.

A *period* is the duration of the periodic voltage curve. The *frequency* is the number of oscillations (periods) per second. The lower the period duration is, the higher is the frequency. Frequencies are given in the unit Hertz (Hz). One Hertz means one oscillation per second.

Frequenz und Datensignal

Die Elektrotechnik unterscheidet die Spannungsarten *Gleichspannung*, bei der die Höhe und Polarität der Spannung immer gleich bleibt, und *Wechselspannung*, bei der die sich die Höhe und Polarität periodisch ändern.

Die *Periodendauer* ist die Zeit des periodischen Spannungsverlaufs. Die *Frequenz* ist die Anzahl der Schwingungen pro Sekunde. Je niedriger die Periodendauer ist, desto höher ist die Frequenz. Frequenzen gibt man in der Einheit Hertz (Hz) an. Ein Hertz bedeutet eine Schwingung pro Sekunde.

$$\text{Frequency } [\text{Hz}] = \frac{1}{\text{Period } [\text{s}]}$$

Communication in computer networks takes place through the exchange of binary data in the form of analog signals that are subject to physical laws. This includes *attenuation* (signal weakening), which reduces the amplitude of a signal with increasing distance on all transmission media. If the amplitude of a data signal has dropped below a particular value, it can no longer be interpreted. Thus, the attenuation limits the maximum distance that can be bridged for all transmission media. The higher the frequency is, the higher is the attenuation.

Die Kommunikation in Computernetzen erfolgt durch den Austausch binärer Daten in Form analoger Signale, die physikalischen Gesetzmäßigkeiten unterliegen. Dazu gehört die *Dämpfung* (Signalabschwächung), welche die Amplitude eines Signals mit zunehmender Wegstrecke auf allen Übertragungsmedien abschwächt. Hat die Amplitude eines Datensignals einen bestimmten Wert unterschritten, kann es nicht mehr eindeutig erkannt werden. Die Dämpfung begrenzt also die maximal überbrückbare Distanz bei allen Übertragungsmedien. Je höher die Frequenz ist, desto höher ist auch die Dämpfung.

3.8

Fourier Series and Bandwidth

A square-wave signal – also a binary signal – can be represented by a *Fourier series* as the sum of a set of oscillating functions. A square wave signal consists of a *fundamental frequency* and *harmonics* (integer multiples of the fundamental frequency). The more harmonics are considered, the more similar becomes the result with a square wave signal. The Fourier series provides information about the deformation of the data signal and the required bandwidth of the transmission medium.

Fourierreihe und Bandbreite

Ein Rechtecksignal – also auch ein Binärsignal – lässt sich mit der *Fourierreihe* als Überlagerung von harmonischen Schwingungen darstellen. Ein Rechtecksignal besteht aus einer *Grundfrequenz* und aus *Harmonischen* (ganzzahligen vielfachen der Grundfrequenz). Je mehr Harmonische berücksichtigt werden, umso näher kommt man einem Rechtecksignal. Die Fourierreihe gibt Auskunft über die Verformung des Datensignals und die nötige Bandbreite des Übertragungsmediums.

To transmit a square-wave signal clearly, the transmission medium needs to be able to transmit in a bug-free way not only the fundamental frequency but also the harmonics of the 3rd and 5th order (which means three and five times the frequency). The harmonics of the 3rd and 5th order are necessary for keeping the square wave's rectangular shape and for preventing rounded-off signal edges. In practice, the harmonics are more attenuated than the fundamental frequency. The *bandwidth*, from the viewpoint of the transmission medium, is the range of frequencies which can be transmitted via the medium without interference [11].

Um ein Rechtecksignal eindeutig erkennbar zu übertragen, muss das Übertragungsmedium nicht nur die Grundfrequenz, sondern auch die 3. und 5. Harmonische (also 3- und 5-fache Frequenz) fehlerfrei übertragen. Die 3. und 5. Harmonische sind nötig, damit das Rechtecksignal noch seine rechteckige Form behält und nicht abgerundet aussieht In der Praxis werden die Harmonischen immer stärker gedämpft als die Grundfrequenz. Die *Bandbreite* aus Sicht des Übertragungsmediums ist der Bereich von Frequenzen, der über das Medium ohne Beeinflussung übertragen werden kann [11].

3.9

Bit Rate and Baud Rate

The *bit rate* is the number of transmitted payload in bits per time. Typically it is measured in bits per second (b/s).

The *baud rate* is the number of transferred symbols per time unit. The baud rate is also called symbol rate. Initially, the baud rate indicated the signaling rate of a telegraph, thus the number of Morse code characters per second. One baud indicates that one symbol is transmitted per second.

The ratio between bit rate and baud rate depends on the line encoding scheme used (see section 5.5).

Bitrate und Baudrate

Die *Bitrate* ist die Anzahl der übertragenen Nutzdaten in Bits pro Zeiteinheit. Typischerweise wird in Bit pro Sekunde (b/s) gemessen.

Die *Baudrate* ist die Anzahl der übertragenen Symbole pro Zeiteinheit. Baud nennt man auch Symbolrate. Ursprünglich gab die Baudrate die Signalisierungsgeschwindigkeit beim Telegrafen an, also die Anzahl der Morsezeichen pro Sekunde. Ein Baud ist die Geschwindigkeit, wenn ein Symbol pro Sekunde übertragen wird.

Das Verhältnis zwischen Bitrate und Baudrate hängt von der verwendeten Kodierung (siehe Abschnitt 5.5) ab.

3.10

Bandwidth and Latency

The performance of a computer network depends on its *bandwidth* (*throughput*) and *latency* (*delay*).

The bandwidth indicates how many bits can be transferred via the network over a period of time.

A network with a bandwidth (throughput) of 1 Mbit/s can transfer one million bits per second. Thus, a bit has a *width* of 1 μs. If the bandwidth doubles, the number of bits that can be transmitted per second doubles.

Bandbreite und Latenz

Für die Leistungsfähigkeit eines Computernetzes sind dessen *Bandbreite* (*Durchsatz*) und *Latenz* (*Verzögerung*) entscheidend.

Die Bandbreite gibt an, wie viele Bits innerhalb eines Zeitraums über das Netzwerk übertragen werden können.

Bei einem Netzwerk mit einer Bandbreite bzw. Durchsatzrate von 1 Mbit/s können eine Millionen Bits pro Sekunde übertragen werden. Ein Bit ist somit eine millionstel Sekunde, also 1 μs *breit*. Verdoppelt sich die Bandbreite, verdoppelt sich die Bits, die pro Sekunde übertragen werden können.

The latency of a network is the time it takes for a message (of a given size) to travel from one end of the network to the other. The latency is the sum of the *propagation delay*, *transmission delay* and *waiting time* [9].

Die Latenz eines Netzwerks ist die Zeit, die nötig ist, bis eine Nachricht (mit einer bestimmten Größe) von einem Ende des Netzwerks zum anderen Ende gelangt ist. Die Latenz ist die Summe aus der *Ausbreitungsverzögerung*, *Übertragungsverzögerung* und *Wartezeit* [9].

$$\text{Latency} = \text{Propagation delay} + \text{Transmission delay} + \text{Waiting time}$$

$$\text{Propagation delay} = \frac{\text{Distance}}{\text{Speed of light} * \text{Velocity factor}}$$

$$\text{Transmission delay} = \frac{\text{Message size}}{\text{Bandwidth}}$$

The speed of light is $299,792,458\,\mathrm{m/s}$, and the velocity factor depends on the transmission medium used. The velocity factor is 1 for a vacuum, 0.6 for twisted pair cables, 0.67 for optical fiber cables, and 0.77 for coaxial cables.

The distance is the length of the network connection via which the data is sent. Waiting times are caused by network devices (e.g., Switches) on the way to the destination because they need to buffer the received data before forwarding it. If the message only consists of a single bit and if the network connection between sender and receiver is only a single line or channel, the transmission delay and the waiting time are irrelevant. In this case, the latency is equal to the propagation delay [9].

Die Lichtgeschwindigkeit ist $299.792.458\,\mathrm{m/s}$ und der Ausbreitungsfaktor (Verkürzungsfaktor), hängt vom Übertragungsmedium ab. Der Ausbreitungsfaktor ist 1 für Vakuum, 0,6 für Twisted-Pair-Kabel, 0,67 für Glasfaser und 0,77 für Koaxialkabel.

Die Entfernung entspricht der Länge der Netzwerkverbindung, über die die Daten gesendet werden. Wartezeiten ergeben sich durch Netzwerkgeräte (z. B. Switche) auf dem Weg zum Ziel, die empfangene Daten erst zwischenspeichern müssen, bevor sie diese weiterleiten. Besteht die Nachricht nur aus einem einzigen Bit und handelt es sich bei der Netzwerkverbindung zwischen Sender und Empfänger nur um eine einzige Leitung oder einen Kanal, sind die Übertragungsverzögerung und die Wartezeit irrelevant. In diesem Fall entspricht die Latenz der Ausbreitungsverzögerung [9].

3.10.1

Bandwidth-Delay Product

With the *bandwidth-delay product*, it is possible to calculate the volume of a network connection. It calculates the product of the bandwidth specified in bits per second and the delay (latency), which is specified in seconds.

Signals cannot travel infinitely fast on transmission media. The propagation speed is in any event limited by the speed of light and depends on the transmission medium and its velocity

Bandbreite-Verzögerung-Produkt

Mit dem *Bandbreite-Verzögerung-Produkt* ist es möglich, das Volumen einer Netzwerkverbindung zu berechnen. Es berechnet das Produkt aus der Bandbreite die in Bits pro Sekunde angegeben ist und der Verzögerung, die in Sekunden angeben ist.

Signale können sich auf Übertragungsmedien nicht unendlich schnell bewegen. Die Ausbreitungsgeschwindigkeit wird in jedem Fall von der Lichtgeschwindigkeit begrenzt und hängt vom

factor. For this reason, there may be data on the transmission medium which the sender has already sent but did not yet arrive at the receiver.

Übertragungsmedium und dem sich daraus ergebenden Ausbreitungsfaktor (Verkürzungsfaktor) ab. Aus diesem Grund können sich auf dem Übertragungsmedium Daten befinden, die der Sender bereits losgeschickt hat, die aber noch nicht beim Empfänger angekommen sind.

The product of bandwidth and delay corresponds to the maximum number of bits that can reside in the line between the sender and receiver.

Das Produkt aus Bandbreite und Verzögerung entspricht der maximalen Anzahl Bits, die sich zwischen Sender und Empfänger in der Leitung befinden können.

The following example calculates the bandwidth-delay product for a network connection with 100 Mbit/s bandwidth at a delay of 10 ms.

Das folgende Rechenbeispiel berechnet das Bandbreite-Verzögerungs-Produkt für eine Netzwerkverbindung mit 100 Mbit/s Bandbreite bei einer Verzögerung von 10 ms.

$$100,000,000\,\text{Bits/s} \times 0.01\,\text{s} = 1,000,000\,\text{Bits}$$

Up to 1,000,000 Bits reside inside the network connection, which is equivalent to 125,000 bytes (approx. 123 kB).

Es befinden sich also maximal 1.000.000 Bits auf der Netzwerkverbindung, was 125.000 Bytes (ca. 123 kB) entspricht.

In the context of the bandwidth delay product, there are different points of view as to what exactly is meant by the delay.

Im Zusammenhang mit dem Bandbreite-Verzögerung-Produkt gibt es unterschiedliche Sichtweisen, was genau unter der Verzögerung zu verstehen ist.

Some sources [9][17] define the bandwidth-delay product as the product of bandwidth and half *Round-Trip-Time* (RTT), the so-called *one-way latency*. This point of view is appropriate especially for the Data Link Layer and Ethernet (see chapter 6) because there, no acknowledgments for correctly received frames are returned to the sender.

Einige Quellen [9][17] definieren das Bandbreite-Verzögerung-Produkt als Produkt aus Bandbreite und halber *Round-Trip-Time* (RTT), die sogenannte *Einweglatenz*. Diese Sichtweise ist besonders im Hinblick auf die Sicherungsschicht und Ethernet passend (siehe Kapitel 6), weil dort keine Bestätigungen (*Acknowledgments*) für korrekt empfangene Rahmen zum Sender zurückgeschickt werden.

Other sources [7][10][17][24] define the bandwidth-delay product as a product of bandwidth and full RTT (see also RFC 1072). This point of view is appropriate especially for the Transport Layer protocol TCP, because there, for each transmission, the acknowledgment (`ACK`) must also to be taken into account. The RTT can be measured with the command line tools `ping` or `traceroute` (see section 7.8 and chapter 11). Typically, inside an Ethernet LAN with twisted pair cables, the RTT of an IP packet that has a size of 1.5 kb is less than 1 ms. In a WLAN, the typical RTT is less than 10 ms. For communication between different networks, the RTT depends on the network segments which need to be passed, as well as their network technologies and distances.

Andere Quellen [7][10][17][24] definieren das Bandbreite-Verzögerung-Produkt als Produkt aus Bandbreite und vollständiger RTT (siehe auch RFC 1072). Diese Sichtweise ist besonders im Hinblick auf die Transportschicht und dort für das Transportprotokoll TCP sinnvoll, da für jede Übertragung auch zusätzlich die Bestätigung (`ACK`) berücksichtigt werden muss. Die RTT kann mit den Kommandozeilenwerkzeugen `ping` oder `traceroute` (siehe Abschnitt 7.8 und Kapitel 11) gemessen werden. Typischerweise ist die RTT eines 1,5 kB großen IP-Pakets in einem Ethernet-LAN mit Twisted-Pair-Kabeln kleiner als 1 ms. Innerhalb eines WLAN ist die typische RTT kleiner als 10 ms. Bei netzübergreifender Kommunikation hängt die RTT von den zu überwindenden Netzwerkabschnitten, deren Netzwerktechnologien und Distanzen ab.

3.11

Media Access Control

When using wired networks, such as Thin Ethernet (10BASE2 or Token Ring, as well as with wireless networks, all nodes access a shared transmission medium.

A *media access control method* must ensure that only one participant sends data at a time because only if this condition is satisfied, the data can be transmitted without errors and collisions be avoided. In shared media, it is distinguished between *deterministic media access control methods* and *non-deterministic media access control methods*.

Zugriffsverfahren

Bei kabelgebundenen Computernetzen wie Thin Ethernet (10BASE2) oder Token Ring sowie bei Funknetzen greifen alle Teilnehmer auf ein gemeinsames Übertragungsmedium zu.

Über ein *Zugriffsverfahren* muss sichergestellt sein, dass innerhalb eines Zeitraums immer nur ein Teilnehmer Daten sendet, denn nur dann können die Daten fehlerfrei übertragen und Kollisionen vermieden werden. Bei Shared Media unterscheidet man *Deterministische* und *Nicht-deterministische Zugriffsverfahren*.

3.11.1

Deterministic Media Access Control

The access to the transmission medium takes place at a particular time in agreement with the other nodes.

One example is the *token passing scheme* of Token Ring and FDDI. The permission to send is implemented via a token. The node that has the token is permitted to send data for a particular period of time. Once a station has finished transmitting, it passes the token to the next node. If a node does not want to send data, it immediately passes the token to the next node. The token passing scheme is fair because no node is skipped during the passing of the token. Each node is allowed to send data after a certain waiting period of a predictable maximum duration.

Deterministisches Zugriffsverfahren

Der Zugriff auf das Übertragungsmedium erfolgt zu einem bestimmten Zeitpunkt in Übereinstimmung mit den anderen Teilnehmern.

Ein Beispiel ist das *Token-Passing-Verfahren* bei Token Ring und FDDI. Das Senderecht wird über ein Token realisiert. Der Teilnehmer, der das Token hat, ist berechtigt, für einen bestimmten Zeitraum Daten zu versenden. Ist ein Teilnehmer mit dem Senden fertig, gibt er das Senderecht weiter. Macht der Teilnehmer von seinem Senderecht keinen Gebrauch, gibt er das Senderecht weiter. Das Token-Passing-Verfahren ist fair, denn kein Teilnehmer wird bei der Weitergabe des Tokens übergangen. Jeder Teilnehmer darf nach einer bestimmten Wartezeit, deren maximale Dauer vorhersehbar ist, Daten senden.

3.11.2

Non-deterministic Media Access Control

Regarding access to the transmission medium, all nodes are in direct competition with each other. The waiting period before the transmission medium can be used, and the amount of data that can be transferred after a particular time, cannot be predicted and depends on the number of nodes and the amount of data that is sent by each node.

Nicht-deterministisches Zugriffsverfahren

Alle Teilnehmer stehen in Bezug auf den Zugriff auf das Übertragungsmedium in direktem Wettbewerb zueinander. Die Wartezeit des Zugriffs und die Datenmenge, die nach einem bestimmten Zeitpunkt übertragen werden kann, sind nicht vorhersagbar und hängen von der Anzahl der Teilnehmer und der Datenmenge ab, die von den einzelnen Teilnehmern versendet wird.

One example is Carrier Sense Multiple Access Collision Detection (CSMA/CD) of Thin Ethernet (10BASE2). If a node wants to send data, it first checks whether the transmission medium is in the idle state. If it is in the idle state, the node starts to send out data. If two or more nodes want to send data at the same time, both assume the transmission medium is idle, and a collision occurs. By using CSMA/CD, collisions are detected by all nodes, and consequently, the senders abort their sending procedures. After a waiting period that is determined by each node randomly, the nodes again try to send their data. This approach aims to reduce the likelihood that further collisions occur.

Ein Beispiel ist Carrier Sense Multiple Access Collision Detection (CSMA/CD) bei Thin Ethernet (10BASE2). Will ein Teilnehmer Daten senden, prüft er vorher, ob das Übertragungsmedium frei ist. Ist es frei, kann der Teilnehmer senden. Wollen zwei oder mehr Teilnehmer zur selben Zeit senden, gehen beide von einem freien Medium aus und es kommt zur Kollision. Durch CSMA/CD werden Kollisionen durch alle Teilnehmer erkannt, worauf die sendenden Teilnehmer das Senden abbrechen. Die Teilnehmer versuchen nach einer Wartezeit, die jeder selbst nach dem Zufallsprinzip ermittelt, erneut zu senden. Das soll die Wahrscheinlichkeit einer erneuten Kollision verringern.

3.12

Collision Domain

If multiple network devices that use a shared transmission medium attempt to send data at the same time, unwanted collisions occur. The area in which collisions can expand is called *collision domain*. A collision domain is a network or a part of a network in which multiple network devices use a shared transmission medium.

For collision detection, the wired network standard Ethernet uses the media access control method CSMA/CD. For wireless networks, guaranteed collision detection is impossible. For this reason, WLAN uses the media access control method CSMA/CA for collision avoidance.

Kollisionsgemeinschaft

Versuchen mehrere Netzwerkgeräte, die ein gemeinsames Übertragungsmedium nutzen, gleichzeitig zu senden, kommt es zu unerwünschten Kollisionen. Der Bereich, in dem sich Kollisionen ausdehnen können, heißt *Kollisionsdomäne* (Kollisionsgemeinschaft). Eine Kollisionsdomäne ist ein Netzwerk oder ein Teil eines Netzwerks, in dem mehrere Netzwerkgeräte ein gemeinsames Übertragungsmedium nutzen.

Zur Kollisionserkennung verwendet der kabelgebundene Netzwerkstandard Ethernet das Medienzugriffsverfahren CSMA/CD. Bei Funknetzen ist eine garantierte Kollisionserkennung unmöglich. Aus diesem Grund verwendet WLAN zur Kollisionsvermeidung das Medienzugriffsverfahren CSMA/CA.

4

Protocols and Reference Models

Communication in computer networks is based on the exchange of messages. If the participants want to communicate with each other, they must be able to understand each other. That is why agreements are made that clarify how communication works. These agreements are called protocols, and they can be found on different levels (layers). This starts with the details of the bit transmission on the bottom layer and continues to the details at the highest layers that specify how the information is displayed.

Protocols define the *syntax*, which is the format of valid messages, and the *semantics*, which is the vocabulary and the meaning of valid messages.

Because of the many requirements that computer networks need to satisfy, communication into these is subdivided into layers within *reference models*. Each *layer* deals with a particular aspect of communication and provides *interfaces* to the layer above and the layer below. Each interface consists of a set of *operations*, which together define a *service*. In the layers, the data are encapsulated (see *data encapsulation* in section 4.3). Because each layer is self-contained, individual protocols can be modified or replaced without affecting all aspects of communication. This is possible as long as there is no modification to the interface and the behavior for the next upper layer. The three most popular reference models are the TCP/IP reference model, the OSI reference model and the hybrid reference model.

Protokolle und Schichtenmodelle

Die Kommunikation in Computernetzen basiert auf dem Austausch von Nachrichten. Sollen Teilnehmer miteinander kommunizieren, müssen sie einander verstehen können. Darum werden Absprachen getroffen, die klären, wie die Kommunikation funktioniert. Diese Absprachen heißen Protokolle und sind auf verschiedenen Ebenen (Schichten) anzutreffen. Das beginnt mit den Details der Bitübertragung auf der untersten Ebene bis zu den Details auf hoher Ebene, die festlegen, wie die Informationen dargestellt werden.

Protokolle definieren die *Syntax*, also das Format gültiger Nachrichten, und mit der *Semantik* das Vokabular gültiger Nachrichten und deren Bedeutung.

Wegen der zahlreichen Anforderungen an Computernetze wird die Kommunikation in diesen mit *Schichtenmodellen* unterteilt. Jede *Schicht* (*Layer*) behandelt einen bestimmten Aspekt der Kommunikation und bietet *Schnittstellen* zur darüberliegenden und darunterliegenden Schicht. Jede Schnittstelle besteht aus einer Menge von *Operationen*, die zusammen einen *Dienst* definieren. In den Schichten werden die Daten gekapselt (siehe *Datenkapselung* in Abschnitt 4.3). Weil jede Schicht in sich abgeschlossen ist, können einzelne Protokolle verändert oder ersetzt werden, ohne alle Aspekte der Kommunikation zu beeinflussen. Das ist möglich, solange es keine Änderung an der Schnittstelle und dem Verhalten für die nächst höhere Schicht gibt. Die drei bekanntesten Schichtenmodelle sind das TCP/IP-Referenzmodell, das OSI-Referenzmodell und das hybride Referenzmodell.

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_4

4.1

TCP/IP Reference Model

The *TCP/IP Reference Model* was developed from 1970 onwards by the Department of Defense (DoD) of the United States. It is also called the *DoD model*. It divides the tasks of communication into four consecutive layers (see Table 4.1).

TCP/IP-Referenzmodell

Das *TCP/IP-Referenzmodell* wurde ab 1970 vom Department of Defense (DoD) der USA entwickelt. Es ist auch unter dem Namen *DoD-Schichtenmodell* bekannt. Die Aufgaben der Kommunikation wurden in vier aufeinander aufbauende Schichten unterteilt (siehe Tabelle 4.1).

Table 4.1: Layers in the TCP/IP Reference Model

Layer	Name	Protocols (examples)
4	Application Layer	HTTP, FTP, SMTP, POP3, DNS, SSH, Telnet
3	Transport Layer	TCP, UDP
2	Internet Layer	IP (IPv4, IPv6), ICMP, IPsec, IPX
1	Link Layer	Ethernet, WLAN, ATM, FDDI, PPP, Token Ring

This reference model was developed in the context of the Arpanet and is considered as the basis of the Internet [19]. For each layer, the functionality is specified. Communication protocols must implement these requirements. The concrete implementation is not specified and can be different. Therefore, for each one of the four layers, multiple protocols do exist.

Dieses Referenzmodell wurde im Rahmen des Arpanet entwickelt und gilt als Basis des Internet [19]. Für jede Schicht ist festgelegt, was sie zu leisten hat. Diese Anforderungen müssen Kommunikationsprotokolle realisieren. Die konkrete Umsetzung ist nicht vorgegeben und kann unterschiedlich sein. Daher existieren für jede der vier Schichten zahlreiche Protokolle.

Figure 4.1: Each Layer adds additional Data to the Message

Each layer adds additional *header* information to the message (see Figure 4.1). Some protocols (e.g., Ethernet) add not only a header in the Link Layer but also a *trailer* at the end of the message. The receiver analyzes the header (and trailer) on the same layer (see Figure 4.2).

In the literature, for example by Andrew S. Tanenbaum [17], the TCP/IP reference model is often presented as a five-layer model, since it makes sense to split the *Link Layer* (*Network Access Layer*) into two layers because they have entirely different tasks. This model is an ex-

Jede Schicht fügt einer Nachricht zusätzliche Informationen als *Header* hinzu (siehe Abbildung 4.1). Einige Protokolle (zum Beispiel Ethernet) fügen in der Netzzugangsschicht nicht nur einen Header, sondern auch einen *Trailer* am Ende der Nachricht an. Header (und Trailer) werden beim Empfänger auf gleicher Schicht ausgewertet (siehe Abbildung 4.2).

Das TCP/IP-Referenzmodell wird in der Literatur häufig, so unter anderem bei Andrew S. Tanenbaum [17], als fünfschichtiges Modell dargestellt, da es sinnvoll ist, die *Netzzugangsschicht* (*Network Access Layer* oder *Link Layer*) in zwei Schichten aufzuteilen, weil diese völlig

Figure 4.2: Communication in the TCP/IP Reference Model

tension of the TCP/IP model and is called the *hybrid reference model*.

The tasks of the individual layers are discussed using the hybrid reference model in the next section.

unterschiedliche Aufgabenbereiche haben. Dieses Modell ist eine Erweiterung des TCP/IP-Modells und heißt *hybrides Referenzmodell*.

Die Aufgaben der einzelnen Schichten werden anhand des hybriden Referenzmodells im nächsten Abschnitt diskutiert.

4.2

Hybrid Reference Model

The names of the top two layers and the tasks of the top three layers are identical to the layers of the TCP/IP reference model (see Figure 4.3). The Internet Layer in the TCP/IP reference model and the Network Layer in the hybrid reference model differ only in the name.

Hybrides Referenzmodell

Die Namen der obersten beiden Schichten und die Aufgabenbereiche der obersten drei Schichten sind mit den Schichten des TCP/IP-Referenzmodells identisch (siehe Abbildung 4.3). Die Internetschicht (*Internet Layer*) beim TCP/IP-Referenzmodell und die Vermittlungsschicht beim hybriden Referenzmodell unterscheiden sich nur im Namen.

4.2.1

Physical Layer

The Physical Layer deals with the characteristics of the various transmission media. This layer is responsible for the transfer of ones and zeros. Here, the physical connection to the medium and the conversion of the data into signals takes place. Protocols of the physical layer define how many bits can be sent per second and whether the transmission can take place simultaneously in both directions.

Bitübertragungsschicht

Die Bitübertragungsschicht (*Physical Layer*) beschäftigt sich unter anderem mit den Merkmalen der verschiedenen Übertragungsmedien. Diese Schicht ist für das Übertragen der Einsen und Nullen zuständig. Hier findet der physische Anschluss an das Medium und die Umsetzung der Daten in Signale statt. Protokolle der Bitübertragungsschicht definieren unter anderem, wie viele Bits pro Sekunde gesendet werden können

und ob die Übertragung in beide Richtungen gleichzeitig stattfinden kann.

4.2.2

Data Link Layer

If errors occur during the transmission of the bit sequences – this often occurs in practice – a procedure is necessary to detect these errors. This is one of the tasks of the Data Link Layer. Also, the protocols of the Data Link Layer control the access to the transmission medium (for example via CSMA/CD or CSMA/CA).

Sicherungsschicht

Treten bei der Übertragung der Bitfolgen Fehler auf – das kommt in der Praxis häufig vor – ist ein Verfahren nötig, um diese Fehler zu erkennen. Das ist einer der Aufgaben der Sicherungsschicht (*Data Link Layer*). Zudem regeln die Protokolle der Sicherungsschicht den Zugriff auf das Übertragungsmedium (zum Beispiel via CSMA/CD oder CSMA/CA).

Figure 4.3: TCP/IP Reference Model and Hybrid Reference Model

At the sender, the Data Link Layer packs the Network Layer packets into *frames* and transmits them with the desired reliability via a physical network from one network device to another. At the receiver, the Data Link Layer identifies the frames in the bit stream of the Physical Layer. The delivery of frames on a physical network requires physical addresses (MAC addresses), whose format defines the Data Link Layer. For error detection, the Data Link Layer protocols attach a checksum to every frame. This way, frames with errors can be detected and thrown away by the receiver. The feature to request discarded frames again is not implemented in the Data Link Layer.

In the Data Link Layer, frames can only be exchanged between network devices that are connected to the same physical network. The connection of different physical networks is done by using Bridges and Switches (Multiport Bridges).

Beim Sender verpackt die Sicherungsschicht die Pakete der Vermittlungsschicht in *Rahmen* (*Frames*) und überträgt sie mit der gewünschten Zuverlässigkeit innerhalb eines physischen Netzes von einem Netzwerkgerät zum anderen. Beim Empfänger erkennt die Sicherungsschicht die Rahmen im Bitstrom der Bitübertragungsschicht. Für die Zustellung der Rahmen auf einem physischen Netzwerk sind physische Adressen (MAC-Adressen) nötig, deren Format die Sicherungsschicht definiert. Zur Fehlererkennung fügen die Protokolle der Sicherungsschicht jedem Rahmen eine Prüfsumme an. So können fehlerhafte Rahmen vom Empfänger erkannt und verworfen werden. Ein erneutes Anfordern verworfener Rahmen sieht die Sicherungsschicht nicht vor.

Auf der Sicherungsschicht können ausschließlich Rahmen zwischen Netzwerkgeräten innerhalb eines physischen Netzes versendet werden. Die Verbindung unterschiedlicher physi-

sche Netze geschieht mit Bridges und Switches (Multiport-Bridges).

4.2.3

Network Layer

The task of the *Network Layer*, which is called*Internet Layer* in the TCP/IP reference model, is to forward *packets* between logical networks over physical network segments. For this *internetworking*, the network layer defines logical addresses (IP addresses).

At the sender, the Network Layer packs the segments of the Transport Layer into packets. At the receiver, the Network Layer detects and unpacks the Packets in the frames of the Data Link Layer.

Routers limit logical subnets. Forwarding packets on their way from the sender to the destination, which is called routing, is one task of this layer. Usually, the connectionless Internet Protocol (IP) is used. Each IP packet is independently *routed* to its destination, and the path is not recorded. Other Network Layer protocols such as IPX have been replaced mainly by IP.

Vermittlungsschicht

Die Aufgabe der *Vermittlungsschicht* (*Network Layer*), die im TCP/IP-Referenzmodell *Internetschicht* (*Internet Layer*) heißt, ist die Weitervermittlung von *Paketen* zwischen logischen Netzen über physische Übertragungsabschnitte hinweg. Für dieses *Internetworking* definiert die Vermittlungsschicht logische Adressen (IP-Adressen).

Beim Sender verpackt die Vermittlungsschicht die Segmente der Transportschicht in Pakete. Beim Empfänger erkennt und entpackt die Vermittlungsschicht die Pakete in den Rahmen der Sicherungsschicht.

Router begrenzen logische Subnetze. Das Routing, also das Weiterleiten von Paketen auf dem Weg vom Sender zum Empfänger, ist eine Aufgabe dieser Schicht. Meist wird das verbindungslose Internetprotokoll IP (Internet Protocol) verwendet. Jedes IP-Paket wird unabhängig an sein Ziel vermittelt (*geroutet*) und der Pfad dabei nicht aufgezeichnet. Andere Internetprotokolle wie IPX wurden von IP weitgehend verdrängt.

4.2.4

Transport Layer

The *Transport Layer* enables the transport of *Segments* between processes on different devices via so-called end-to-end protocols.

At the sender, the Transport Layer packs the data of the Application Layer into segments. At the receiver, the Transport Layer detects and unpacks the segments inside the packets from the Network Layer.

While the Data Link Layer and the Network Layer address network devices in a physical way, in this layer the running processes are addressed with port numbers. The Transport Layer, therefore, ensures that the data in the Network Layer is delivered correctly to the correct applications. It splits the data at the sender, using Transport Layer protocols, into smaller parts (segments), which the protocol of the Network Layer can

Transportschicht

Die *Transportschicht* (*Transport Layer*) ermöglicht den Transport von *Segmenten* zwischen Prozessen auf unterschiedlichen Geräten über sog. Ende-zu-Ende-Protokolle.

Beim Sender verpackt die Transportschicht die Daten der Anwendungsschicht in Segmente. Beim Empfänger erkennt und entpackt die Transportschicht die Segmente in den Paketen der Vermittlungsschicht.

Während die Sicherungsschicht und die Vermittlungsschicht Netzwerkgeräte physisch und logisch adressieren, werden in dieser Schicht die laufenden Prozesse mit Portnummern adressiert. Die Transportschicht sorgt also dafür, dass die Daten der Vermittlungsschicht korrekt an die richtigen Anwendungen ausgeliefert werden. Sie teilt die Daten beim Sender mit Transportprotokollen in kleinere Teile (Segmente) auf, die das

forward. The size of the segments corresponds to the maximum size of the payload of a packet in the Network Layer.

Transport Layer protocols implement different forms of communication. *Connectionless communication* works analogously to a mailbox. The sender sends messages without prior connection establishment. With connectionless communication, the Transport Layer does not provide a way to validate that a segment arrives at the destination. If validation is required, it must be carried out in the Application Layer. The lack of a delivery guarantee is one disadvantage of this form of communication. One benefit is the better data rate because less overhead arises.

One alternative is using *connection-oriented communication*. This works in the same way as the telephone. A connection is established between sender and receiver before data is exchanged. The connection remains active even if no data is transferred. As soon as all data has been exchanged, one of the communication partners terminates the connection.

Connection-oriented communication enables flow control and congestion control, in which the receiver controls the transmission speed of the sender. Depending on the Transport Layer protocol used, the Transport Layer also ensures loss-free delivery of the segments. This is equivalent to a delivery guarantee. The correct order of the segments at the receiver is also guaranteed by this form of communication.

Examples of Transport Layer protocols are the connectionless *User Datagram Protocol* (UDP) and the connection-oriented *Transport Control Protocol* (TCP).

Protokoll der Vermittlungsschicht weiterleiten kann. Die Größe der Segmente entspricht der maximalen Größe der Nutzdaten eines Pakets in der Vermittlungsschicht.

Transportprotokolle bieten unterschiedliche Kommunikationsformen. Die *verbindungslose Kommunikation* funktioniert analog zum Briefkasten. Nachrichten werden vom Sender verschickt, ohne zuvor eine Verbindung aufzubauen. Bei verbindungsloser Kommunikation gibt es in der Transportschicht keine Kontrolle, dass ein Segment ankommt. Ist eine Kontrolle gewünscht, muss diese in der Anwendungsschicht erfolgen. Die fehlende Zustellungsgarantie ist ein Nachteil dieser Kommunikationsform. Ein Vorteil ist der höhere Datendurchsatz, da weniger Overhead anfällt.

Alternativ ist *verbindungsorientierte Kommunikation* möglich. Diese funktioniert analog zum Telefon. Vor dem Datenaustausch wird zwischen Sender und Empfänger eine Verbindung aufgebaut. Die Verbindung bleibt auch dann bestehen, wenn keine Daten übertragen werden. Sobald alle Daten ausgetauscht sind, wird einer der Kommunikationspartner die Verbindung abbauen.

Verbindungsorientierte Kommunikation ermöglicht Datenflusskontrolle, bei welcher der Empfänger die Sendegeschwindigkeit des Senders steuert. Je nach verwendetem Transportprotokoll sichert die Transportschicht auch die verlustfreie Lieferung der Segmente. Es gibt somit eine Zustellungsgarantie. Auch die korrekte Reihenfolge der Segmente beim Empfänger ist bei dieser Kommunikationsform garantiert.

Beispiele für Transportprotokolle sind das verbindungslose *User Datagram Protocol* (UDP) und das verbindungsorientierte *Transport Control Protocol* (TCP).

4.2.5

Application Layer

The *Application Layer* contains all protocols that interact with the user application programs (e.g., browser or email client). Here, the messages (e.g., HTML pages or emails) are located according to the respective Application Layer protocol.

Anwendungsschicht

Die *Anwendungsschicht* (*Application Layer*) enthält alle Protokolle, die mit Anwendungsprogrammen (zum Beispiel Browser oder Email-Programm) zusammenarbeiten. Hier befinden sich die Nachrichten (zum Beispiel HTML-Seiten oder Emails) entsprechend dem jeweiligen Anwendungsprotokoll.

Examples of Application Layer protocols are HTTP, FTP, SMTP, POP3, DNS, SSH, and Telnet.

Beispiele für Anwendungsprotokolle sind HTTP, FTP, SMTP, POP3, DNS, SSH und Telnet.

4.3

How Communication works

The *communication flow* is demonstrated by using the hybrid reference model (see Figure 4.4).

Vertical communication describes the process in which the data passes through the layers of the reference model used. A message is packed layer by layer from the top layer to the bottom layer and extracted by the receiver in the reverse order from the bottom layer to the top layer. The sender adds a header to the data in each layer and a trailer in the Data Link Layer. These additional headers and trailers are analyzed and removed in the respective layers by the receiver. These operations are called *encapsulation* and *de-encapsulation*.

Ablauf der Kommunikation

Der *Ablauf der Kommunikation* wird am Beispiel des hybriden Referenzmodells demonstriert (siehe Abbildung 4.4).

Die *vertikale Kommunikation* beschreibt den Vorgang, bei dem die Daten die Schichten des verwendeten Referenzmodells durchlaufen. Eine Nachricht wird von der obersten Schicht beginnend nach unten Schicht für Schicht immer weiter verpackt und beim Empfänger in umgekehrter Reihenfolge von der untersten Schicht nach oben immer weiter entpackt. Beim Sender wird den Daten in jeder Schicht ein Header und und in der Sicherungsschicht ein Trailer angefügt. Beim Empfänger werden diese zusätzlichen Header und Trailer in den jeweiligen Schichten ausgewertet und entfernt. Diese Vorgänge heißen *Datenkapselung* (*Encapsulation*) und *Entkapselung* (*De-encapsulation*).

Figure 4.4: Data Encapsulation in vertical Communication

In *horizontal communication*, sender and receiver each use the same protocol functions on the same layers (see Figure 4.2).

Bei *horizontaler Kommunikation* verwenden Sender und Empfänger jeweils die gleichen Protokollfunktionen auf den gleichen Schichten (siehe Abbildung 4.2).

Figure 4.5: Comparison of the Reference Models

4.4

OSI Reference Model

Some years after the TCP/IP reference model that dates back to the early 1970s, the *OSI reference model* (Open Systems Interconnection) was developed from 1979 on. It was standardized in 1983 by the International Organization for Standardization (ISO). The structure is similar to the TCP/IP reference model, but the OSI model uses seven layers (see Figure 4.5).

In contrast to the hybrid reference model, some tasks of the Application Layer in the OSI reference model are divided between the two additional layers Session Layer and Presentation Layer.

OSI-Referenzmodell

Einige Jahre nach dem TCP/IP-Referenzmodell, das aus den frühen 1970er Jahren stammt, wurde das *OSI-Referenzmodell* (Open Systems Interconnection) entwickelt und 1983 von der Internationalen Organisation für Normung (ISO) standardisiert. Der Aufbau ist dem TCP/IP-Referenzmodell ähnlich, allerdings verwendet das OSI-Modell sieben Schichten (siehe Abbildung 4.5).

Im Gegensatz zum hybridem Referenzmodell sind einige Aufgaben der Anwendungsschicht beim OSI-Referenzmodell auf die beiden zusätzlichen Schichten Sitzungsschicht und Darstellungsschicht aufgeteilt.

4.4.1

Session Layer

The tasks of the Session Layer include the establishment, monitoring, and termination of sessions. A session is the basis for a virtual connection between two applications on physically independent computers. Also, the Session Layer is intended to control the dialogues, which means that it determines which communication partner is allowed to send data next. Another task is to provide checkpoints that can be used in longer data transfers for synchronization. If the connection fails, the return to a checkpoint avoids having to start the transmission over again from the beginning.

Sitzungsschicht

Aufgaben der Sitzungsschicht (*Session Layer*) sind Aufbau, Überwachung und Beenden von Sitzungen. Eine Sitzung ist die Grundlage für eine virtuelle Verbindung zwischen zwei Anwendungen auf physisch unabhängigen Rechnern. Zudem ist der Sitzungsschicht die Dialogkontrolle zugedacht, also die Kontrolle darüber, welcher Kommunikationspartner als nächstes Daten übertragen darf. Eine weitere Aufgabe ist die Bereitstellung von Kontrollpunkten, die in längeren Datenübertragungen zur Synchronisierung eingebaut werden können. Kommt es zum Verbindungsabbruch, kann zum letzten

Protocols that meet the required capabilities of the Session Layer, are among others, Telnet for the remote control of computers and FTP for file transfer. However, these protocols can also be assigned to the Application Layer. The Application Layer contains the protocols that are used by the application programs. Because FTP and Telnet are used directly by the application programs and not by abstract protocols of upper layers, it makes more sense to assign these protocols of the Session Layer to the Application Layer.

Kontrollpunkt zurückgekehrt werden und die Übertragung muss nicht von vorne beginnen.

Protokolle, die den geforderten Fähigkeiten der Sitzungsschicht entsprechen, sind unter anderem Telnet zur Fernsteuerung von Rechnern und FTP zur Übertragung von Dateien. Allerdings können diese Protokolle auch der Anwendungsschicht zugeordnet werden. Die Anwendungsschicht enthält die Protokolle, die die Anwendungsprogramme verwenden. Da FTP und Telnet direkt von den Anwendungsprogrammen verwendet werden und nicht von abstrakteren Protokollen in höheren Ebenen, ist es sinnvoller, die Protokolle der Sitzungsschicht der Anwendungsschicht zuzuordnen.

4.4.2

Presentation Layer

The Presentation Layer contains rules for the formatting (presentation) of messages. The sender can notify the receiver that a message has a specific format (for example ASCII) to enable the possibly necessary conversion at the receiver. Data records can be specified with fields (e.g., name, student ID number...). The definition of data types and their length, as well as compression and encryption, are also tasks that are assigned to the Presentation Layer.

Just like the Session Layer, the Presentation Layer is seldom used in practice, because all tasks that are assigned to this layer are fulfilled by Application Layer protocols today.

Darstellungsschicht

Die Darstellungsschicht (*Presentation Layer*) enthält Regeln zur Formatierung (Präsentation) der Nachrichten. Der Sender kann den Empfänger informieren, dass eine Nachricht in einem bestimmten Format (zum Beispiel ASCII) vorliegt, um die eventuell nötige Konvertierung beim Empfänger zu ermöglichen. Datensätze können hier mit Feldern (zum Beispiel Name, Matrikelnummer...) definiert werden. Auch die Art und Länge der Datentypen sowie Kompression und Verschlüsselung sind der Darstellungsschicht zugedachte Aufgabenbereiche.

Genau wie die Sitzungsschicht wird auch die Darstellungsschicht in der Praxis kaum benutzt, weil alle dieser Schicht zugedachten Aufgaben heute Anwendungsprotokolle erfüllen.

4.5

Conclusion on the Reference Models

The TCP/IP reference model is considered the basis of the Internet. It specifies for each layer which functionalities need to be implemented by it, but not how this is done.

The OSI reference model is very similar to the TCP/IP reference model. The models only differ in two aspects. In the OSI reference model, the tasks of the Network Layer are divided between

Fazit zu den Referenzmodellen

Das TCP/IP-Referenzmodell gilt als Basis des Internet. Es macht Vorgaben, welche Funktionalitäten in den jeweiligen Schichten zu erbringen sind, aber nicht, wie das zu geschehen hat.

Das OSI-Referenzmodell ist dem TCP/IP-Referenzmodell sehr ähnlich. Beide Modelle unterscheiden sich nur in zwei Punkten. Beim OSI-Referenzmodell sind die Aufgaben der Netz-

the Physical Layer and the Data Link Layer. The tasks of the Application Layer in the OSI reference model are divided between the Session Layer, Presentation Layer, and Application Layer.

The hybrid reference model is helpful because the TCP/IP reference model does not distinguish between the Physical Layer and the Data Link Layer, while their task areas are entirely different. However, subdividing the Application Layer into three layers has not proven to be meaningful and does not take place in practice, because the functionalities, which are intended for the Session Layer and the Presentation Layer, are implemented nowadays by the Application Layer protocols.

The hybrid reference model illustrates the functioning of computer networks in a realistic way because it distinguishes the Physical Layer and the Data Link Layer and at the same time does not subdivide the Application Layer. It combines the benefits of the TCP/IP reference model and the OSI reference model without taking over their drawbacks.

zugangsschicht auf die Bitübertragungsschicht und Sicherungsschicht aufgeteilt. Die Aufgaben der Anwendungsschicht sind beim OSI-Referenzmodell auf die Sitzungsschicht, Darstellungsschicht und Anwendungsschicht aufgeteilt.

Das hybride Referenzmodell ist sinnvoll, weil das TCP/IP-Referenzmodell nicht zwischen Bitübertragungsschicht und Sicherungsschicht unterscheidet, dabei sind deren Aufgabenbereiche vollkommen unterschiedlich. Die Trennung der Anwendungsschicht in drei Schichten hat sich jedoch nicht als sinnvoll herausgestellt und findet in der Praxis nicht statt, denn die Funktionalitäten, die für die Sitzungsschicht und Darstellungsschicht vorgesehen sind, sind heute Teil der Protokolle in der Anwendungsschicht.

Das hybride Referenzmodell bildet die Funktionsweise von Computernetzen realistisch ab, da es die Bitübertragungsschicht und Sicherungsschicht unterscheidet und gleichzeitig die Anwendungsschicht nicht unterteilt. Es kombiniert die Vorteile des TCP/IP-Referenzmodells und des OSI-Referenzmodells, ohne deren jeweilige Nachteile zu übernehmen.

5

Physiscal Layer

The *Physical Layer* is the first layer of the OSI reference model and the hybrid reference model. The transmission media and all devices that are directly connected to the transmission media, including antennas, connectors, and repeaters, are part of the Physical Layer. Another task of this layer is the definition of the line codes, which specify how the data (bit sequences) is sent via the transmission media.

Bitübertragungsschicht

Die *Bitübertragungsschicht* ist die erste Schicht des OSI-Referenzmodells und des hybriden Referenzmodells. Die Übertragungsmedien und alle Geräte, die direkt mit den Medien verbunden sind, das schließt auch Antennen, Stecker und Repeater ein, sind Teil der Bitübertragungsschicht. Die Art und Weise, wie die Daten (Bitfolgen) auf den Übertragungsmedien gesendet werden, legen Leitungscodes fest, deren Definition auch Teil dieser Schicht ist.

5.1

Networking Technologies

This section covers the wired networking technologies Ethernet and Token Ring and the wireless network technologies Bluetooth and WLAN, while the focus is on Ethernet and WLAN (WiFi). Each network technology specifies software (protocols of the Physical Layer and Data Link Layer), hardware (terminal devices, plugs, and jacks) as well as transmission media.

Vernetzungstechnologien

Dieser Abschnitt behandelt die kabelgebundenen Vernetzungstechnologien Ethernet und Token Ring und die Funknetztechnologien Bluetooth und WLAN, wobei der Fokus auf Ethernet und WLAN (WiFi) liegt. Die Vernetzungstechnologien spezifizieren jeweils Software (Protokolle der Bitübertragungsschicht und Sicherungsschicht), Hardware (Endgeräte, Stecker und Buchsen) sowie Übertragungsmedien.

5.1.1

Ethernet

Ethernet (IEEE 802.3) was developed in the 1970s, and it is the most widely used LAN technology since the 1990s. Because of Ethernet, other standards such as Token Ring became obsolete or turned into niche products for special applications like FDDI.

Ethernet

Ethernet (IEEE 802.3) wurde in den 1970er Jahren entwickelt und ist seit den 1990er Jahren die meistverwendete LAN-Technik. Durch Ethernet wurden andere Standards wie Token Ring vollständig verdrängt oder wie FDDI zu Nischenprodukten für Spezialanwendungen gemacht.

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_5

Numerous Ethernet standards differ in the data rate and the transmission medium. There are versions for coaxial cables, twisted pair cables, and fiber-optic cables, with transmission speeds of up to 10 Gbit/s. Because all Ethernet variants are extensions of Thick Ethernet (10BASE5), the *connection type* to the transmission medium is *passive*. This means that network devices are only in an active state when they send data. Table 5.1 contains a selection of common Ethernet standards.

Es existieren zahlreiche Ethernet-Standards, die sich unter anderem in der Übertragungsrate und dem Übertragungsmedium unterscheiden. So existieren Versionen für Koaxialkabel, Twisted-Pair-Kabel und Glasfaser-Kabel bis maximal 10 Gbit/s. Weil alle Ethernet-Varianten Erweiterungen von Thick Ethernet (10BASE5) sind, ist die *Anschlussart* an das Medium *passiv*. Das heißt, dass Netzwerkgeräte nur dann aktiv sind, wenn sie selbst senden. Tabelle 5.1 enthält eine Auswahl bekannter Ethernet-Standards.

Table 5.1: Data Rates of the Ethernet Standards

Standard	Mbps	Transmission medium
10BASE2/5	10	Coaxial cables (50 Ohm impedance)
10BROAD36	10	Coaxial cables (75 Ohm impedance)
10BASE-F	10	Fiber-optic cables
10BASE-T	10	Twisted pair cables (Cat-3)
100BASE-FX	100	Fiber-optic cables
100BASE-T4	100	Twisted pair cables (Cat-3)
100BASE-TX	100	Twisted pair cables (Cat-5)
1000BASE-LX	1,000	Fiber-optic cables
1000BASE-SX	1,000	Fiber-optic cables (Multi-mode fiber)
1000BASE-ZX	1,000	Fiber-optic cables (Single-mode fiber)
1000BASE-T	1,000	Twisted pair cables (Cat-5)
1000BASE-TX	1,000	Twisted pair cables (Cat-6)
2.5GBASE-T	2,500	Twisted pair cables (Cat-5e)
5GBASE-T	5,000	Twisted pair cables (Cat-6)
10GBASE-SR	10,000	Fiber-optic cables (Multi-mode fiber)
10GBASE-LR	10,000	Fiber-optic cables (Single-mode fiber)
10GBASE-T	10,000	Twisted pair cables (Cat-6A)
40GBASE-T	40,000	Twisted pair cables (Cat-8.1)

Looking at Table 5.1, the naming scheme of the Ethernet standards becomes obvious. Each name consists of three parts. Part 1 indicates the data rate, part 2 the transmission method (baseband or broadband) and part 3 the hundredfold factor of the maximum segment length or the transmission medium. For example, the name 10BASE5 indicates:

- Data rate: 10 Mbps
- Transmission method: Baseband
- Maximum segment length: $5 \times 100\,\text{m} = 500\,\text{m}$

In Tabelle 5.1 ist auch das einheitliche Namensschema der Ethernet-Standards erkennbar. Jeder Name besteht aus drei Teilen. Teil 1 definiert die Übertragungsrate, Teil 2 das Übertragungsverfahren (Basisband oder Breitband) und Teil 3 den hundertfachen Faktor der maximalen Segmentlänge oder das Übertragungsmedium. Der Name 10BASE5 zum Beispiel sagt folgendes aus:

- Übertragungsrate: 10 Mbit/s
- Übertragungsverfahren: Basisband
- Maximale Segmentlänge: $5 \times 100\,\text{m} = 500\,\text{m}$

Transmission Methods of Ethernet

With one exception, all Ethernet standards operate according to the *baseband* transmission method (BASE). Baseband systems have no carrier frequencies. This means that data is transmitted directly (in the baseband) on the transmission medium. Digital signals are injected directly as impulses into the copper cable or fiber-optic and occupy the entire bandwidth of the cable or a part of it. The unused part of the bandwidth cannot be used for other services. Therefore, baseband systems offer just a single channel.

In the *broadband* transmission method (BROAD), the data is modulated onto a carrier frequency. This allows several signals to be transmitted simultaneously in different frequency ranges (bands). Only 10BROAD36 uses the broadband method, but due to the high cost for the modulation hardware, the system was no economic success.

Although the broadband concept was not successful in Ethernet, it is now used in many areas of communication and telecommunication. Examples include the cable television network, in which different television channels, and with different carrier frequencies also radio channels, telephone, and Internet are available as well as the electrical power grid that can be used to establish network connections by using *PowerLAN*, which is also called *Powerline Communication*).

Übertragungsverfahren bei Ethernet

Mit einer Ausnahme arbeiten alle Ethernet-Standards mit dem *Basisband*-Übertragungsverfahren (BASE). Basisbandsysteme haben keine Trägerfrequenzen. Das heißt, die Daten werden direkt (im Basisband) auf dem Übertragungsmedium übertragen. Digitale Signale werden direkt als Impulse in das Kupferkabel oder den Lichtwellenleiter eingespeist und belegen die vollständige Bandbreite des Kabels oder einen Teil davon. Dabei kann der ungenutzte Teil der Bandbreite nicht mehr für andere Dienste genutzt werden. Basisbandsysteme bieten also nur einen einzigen Kanal.

Beim *Breitband*-Übertragungsverfahren (BROAD) werden die Daten auf eine Trägerfrequenz aufmoduliert. Dadurch können mehrere Signale gleichzeitig in unterschiedlichen Frequenzbereichen (Trägern) übertragen werden. Ausschließlich 10BROAD36 verwendet das Breitbandverfahren, aber wegen der hohen Hardwarekosten für die Modulation war das System wirtschaftlich nicht erfolgreich.

Zwar konnte sich das Breitbandkonzept bei Ethernet nicht durchsetzen, doch wird es heute in viele Bereichen der Nachrichtenübermittlung und Telekommunikation verwendet. Beispiele sind das Kabelfernsehnetz, in dem verschiedene Fernsehkanäle, und mit unterschiedlichen Trägerfrequenzen auch Radiokanäle, Telefon und Internet zur Verfügung stehen sowie das Elektrizitätsnetz, über das via *PowerLAN* bzw. *Powerline Communication* auch Netzwerkverbindungen möglich sind.

5.1.2

Token Ring

Token Ring (IEEE 802.5) is a standard for LANs, in which the terminal devices are logically connected as a ring. A *token* travels in the ring and is passed from one node to the next.

The *connection type* to the transmission medium is *active*. This means that the network devices continuously participate actively in the token passing. Token Ring with a data rate of 4 Mbps was introduced in 1985 for the IBM PC. The standard with a data rate of 16 Mbps was

Token Ring

Token Ring (IEEE 802.5) ist ein Standard für LANs, in dem die Endgeräte logisch zu einem Ring verbunden sind. Über den Ring kreist ein *Token*, das von einem Teilnehmer zum nächsten weitergereicht wird.

Die *Anschlussart* an das Übertragungsmedium ist *aktiv*. Das heißt, die Netzwerkgeräte beteiligen sich fortwährend aktiv an der Weitergabe des Tokens. Token Ring mit 4 Mbit/s Übertragungsrate wurde 1985 für den IBM PC vorgestellt. Der Standard für 16 Mbit/s Über-

introduced in 1989. Since 1998, there is also a standard for 100 Mbit/s. Token Ring has been IBM's preferred networking technology until the mid-1990s but is rarely used today since IBM abandoned this technology in 2004.

The token frame that travels over the ring implements a *deterministic access method* (see section 3.11.1). The token is passed from one terminal device to the next. Even in idle mode, which means that no node wants to send data, the terminal devices continuously pass on the token frame. If a terminal device wants to send data, it waits until the token frame has reached it. Then, the terminal device appends its payload to the token, adds the necessary control signals and sets the token bit from value `0` (*free token*) to `1` (*data frame*). The terminal device places the data frame on the ring, where it is passed from one terminal device to the next. If a data frame token reaches its destination, the receiver copies the payload data and acknowledges the receive. The sender receives the acknowledgment and sends the token with the next payload data, or it places a free token on the ring.

tragungsrate erschien 1989. Seit 1998 gibt auch einen Standard für 100 Mbit/s. Token Ring war bis Mitte der 1990er Jahre die von IBM bevorzugte Vernetzungstechnologie, wird heute aber kaum noch verwendet, da IBM Token Ring im Jahr 2004 aufgegeben hat.

Das über den Ring kreisende Token realisiert ein *deterministisches Zugriffsverfahren* (siehe Abschnitt 3.11.1). Das Token wird von einem Endgerät zum nächsten weitergereicht. Auch im Leerlauf, also wenn kein Teilnehmer senden möchte, geben die Endgeräte den Token-Rahmen fortwährend weiter. Möchte ein Endgerät Daten senden, wartet es, bis das Token ihn erreicht hat. Dann hängt das Endgerät seine Nutzdaten an das Token an, ergänzt es um die nötigen Steuersignale und setzt das Token-Bit von `0` (*Freies Token*) auf `1` (*Datenrahmen*). Das Endgerät gibt das Datenrahmen-Token in den Ring, wo es von einem Endgerät zum nächsten weitergereicht wird. Empfängt der Empfänger das an ihn adressierten Datenrahmen-Token, kopiert er die Nutzdaten und quittiert den Datenempfang. Der Sender empfängt die Quittung und sendet das Token mit den nächsten Nutzdaten oder setzt ein Frei-Token auf den Ring.

5.1.3

Wireless Local Area Network (WLAN)

WLAN (WiFi) is the most popular technology for setting up wireless computer networks. Some unique characteristics of the transmission medium cause the following challenges, which need to be considered when setting up and working with wireless networks [23]:

- *Interference with other sources.* Examples include WLAN and Bluetooth. Both network technologies operate on the same frequency band and can interfere with each other. Electromagnetic noise from motors or microwave ovens can also cause interferences.

- *Multipath propagation.* This effect occurs when parts of the electromagnetic waves are reflected and therefore travel paths of different lengths from the sender to the re-

Wireless Local Area Network (WLAN)

Die bekannteste Technologie zum Aufbau von Funknetzen im Bereich der Computernetze ist WLAN (WiFi). Einige spezielle Eigenschaften des Übertragungsmediums sind der Grund für die folgenden Herausforderungen, die beim Aufbau und der Arbeit mit Funknetzen zu beachten sind [23]:

- *Interferenzen mit anderen Quellen.* Beispiele hierfür sind WLAN und Bluetooth. Beide Netzwerktechnologien arbeiten auf dem gleichen Frequenzband und können interferieren. Auch elektromagnetisches Rauschen durch Motoren oder Mikrowellengeräte können zu Interferenzen führen.

- *Mehrwegeausbreitung.* Dieser Effekt tritt auf, wenn Teile der elektromagnetischen Wellen reflektiert werden und darum unterschiedlich lange Wege vom Absender

ceiver. As a result, a difficult to interpret signal arrives at the receiver, because the reflections influence subsequent transmissions. The principle is shown in Figure 5.1. A similar situation occurs when objects move around between sender and receiver because this may change the propagation paths.

zum Empfänger zurücklegen. Das Resultat ist ein unscharfes Signal beim Empfänger, weil die Reflektionen beeinflussen nachfolgende Übertragungen. Das Prinzip zeigt Abbildung 5.1. Auch wenn sich Objekte zwischen Sender und Empfänger bewegen, können sich die Ausbreitungswege im Laufe der Zeit ändern.

Figure 5.1: Multipath propagation

- *Hidden Terminal* (invisible or hidden terminal devices). Obstacles cause terminal devices that communicate with the same device (e.g., an access point) to not recognize each other and thus to interfere with each other. The principle is shown in Figure 6.21.

- *Hidden-Terminal* (unsichtbare bzw. versteckte Endgeräte). Hindernisse sind dafür verantwortlich, dass Endgeräte, die mit dem gleichen Gerät (zum Beispiel einer Basisstation) kommunizieren, einander nicht erkennen und sich dadurch gegenseitig stören. Das Prinzip zeigt Abbildung 6.21.

- *Fading* (decreasing signal strength). Electromagnetic waves are gradually attenuated by obstacles (e.g., walls) and in free space. The principle is shown in Figure 6.22.

- *Fading* (abnehmende Signalstärke). Elektromagnetische Wellen werden durch Hindernisse (zum Beispiel Wände) und im freien Raum allmählich abgeschwächt. Das Prinzip zeigt Abbildung 6.22.

WLAN is a standard for wireless local area network, that is based on the IEEE 802.11 family of standards. Communication between wireless terminal devices can take place directly in *Ad-hoc mode* or *Infrastructure mode* via an access point.

WLANs sind lokale Funknetze, die meist auf Standards der Normenfamilie IEEE 802.11 basieren. Die Kommunikation zwischen Endgeräten kann direkt im *Ad-hoc-Modus* oder im *Infrastruktur-Modus* via Basisstation (Access Point) erfolgen.

Table 5.2: Data Rates of the IEEE Standards for WLAN

IEEE standard	Maximum (gross) data rate	Realistic (net) data rate
802.11	2 Mbps	1 Mbps
802.11a	54 Mbps[a]	20–22 Mbps
802.11b	11 Mbps[b]	5–6 Mbps
802.11g	54 Mbps	20–22 Mbps
802.11h	54 Mbps[a]	20–22 Mbps
802.11n	600 Mbps[c]	200-250 Mbps
802.11ac	1733 Mbps[d]	800-850 Mbps

[a] Some manufacturers added proprietary extensions to their products, enabling them to support 108 Mbps at 40 MHz channel width
[b] Some manufacturers added proprietary extensions to their products, enabling them to support 22 Mbps at 40 MHz channel width
[c] When using 4x4 MIMO and 40 MHz channel width
[d] When using 4x4 MIMO and 80 MHz channel width

In Ad-hoc mode, the terminal devices form a meshed network. So they communicate directly with each other. Each terminal device can have multiple connections to other devices. To set up an Ad-hoc mode network, the same network name – Service Set Identifier (SSID) and the same encryption parameters must be set for all terminal devices.

In Infrastructure mode, each terminal device registers with its MAC address at the access point. The access point sends small *Beacon Frames* to all terminal devices in range at adjustable intervals (e.g., ten times per second). The Beacons contain the network name (SSID), the list of supported data rates and the encryption type.

Im Ad-hoc-Modus bilden die Endgeräte ein vermaschtes Netz. Sie kommunizieren also direkt miteinander. Jedes Endgerät kann mehrere Verbindungen zu anderen Geräten unterhalten. Zum Aufbau eines Ad-hoc-Netzes müssen bei allen Endgeräten der gleiche Netzwerkname – *Service Set Identifier* (SSID) und dieselben Verschlüsselungsparameter eingestellt sein.

Im Infrastruktur-Modus melden sich die Endgeräte mit ihren MAC-Adressen an der Basisstation an. Diese sendet in einstellbaren Intervallen (zum Beispiel 10 × pro Sekunde) kleine *Leuchtfeuer-Rahmen* (*Beacons*) an alle Endgeräte im Empfangsbereich. Die Beacons enthalten unter anderem den Netzwerknamen (SSID), die Liste der unterstützten Übertragungsraten und die Art der Verschlüsselung.

Data Rate of WLAN

The various WLAN standards offer different data rates. All stations share the bandwidth for upload and download, which is further reduced by the media access method used (see section 6.8.2). Also, the gross transfer rate contains the data of the WLAN frames (see Figure 6.17), which is not part of the payload. Therefore, the available net transmission rate, which only takes into account the payload, is only slightly more than half of the gross rate even under optimal conditions (see Table 5.2).

Datenübertragungsraten bei WLAN

Die verschiedenen WLAN-Standards bieten unterschiedliche Datenübertragungsraten. Alle Stationen teilen sich die Bandbreite für Up- und Download, die durch das verwendete Medienzugriffsverfahren (siehe Abschnitt 6.8.2) weiter reduziert wird. Zudem enthält die Bruttoübertragungsrate noch diejenigen Daten in den WLAN-Rahmen (siehe Abbildung 6.17), die nicht den Nutzdaten zugerechnet werden. Darum liegt die erreichbare Nettoübertragungsrate, die nur die reinen Nutzdaten berücksichtigt, selbst unter optimalen Bedingungen nur wenig über der Hälfte der Bruttowerte (siehe Tabelle 5.2).

Since WLAN was designed for use inside buildings, it uses only a low transmission power (up to 100 mW at 2.4 GHz and 1 W at 5 GHz), which is considered safe for human health. For comparison, the maximum permitted transmission power of GSM telephones that operate in the frequency range of 880–960 MHz is 2 W. There are also WLAN devices for 2.4 GHz that offer a transmission power of up to 1 W. However, using such devices is illegal in Germany and many other countries.

Da WLAN für den Einsatz innerhalb von Gebäuden entwickelt wurde, sendet es auch nur mit einer niedrigen Leistung (maximal 100 mW bei 2,4 GHz und 1 W bei 5 GHz), was als gesundheitlich unbedenklich gilt. Die maximal erlaubte Sendeleistung von GSM-Telefonen, die im Frequenzbereich 880–960 MHz senden, zum Vergleich, ist 2 W. Es existieren auch WLAN-Geräte für 2,4 GHz mit einer Sendeleistung von bis zu 1 W. Deren Betrieb ist aber in Deutschland und vielen anderen Ländern illegal.

Frequencies of WLAN

Most WLAN standards use the frequency blocks 2.4000–2.4835 GHz and 5.150–5.725 GHz in the microwave range. The standards differ – among other things – in the data rates (see Table 5.2), frequency blocks used (see Table 5.3), and modulation methods, as well as the resulting channel width (see Table 5.5).

Frequenzen von WLAN

Die meisten WLAN-Standards verwenden die Frequenzblöcke 2,4000–2,4835 GHz und 5,150–5,725 GHz im Mikrowellenbereich. Die Standards unterscheiden sich unter anderem in den Datenübertragungsraten (siehe Tabelle 5.2), verwendeten Frequenzblöcken (siehe Tabelle 5.3) und Modulationsverfahren, mit den daraus resultierenden Kanalbandbreiten (siehe Tabelle 5.5).

Table 5.3: IEEE Standards for WLAN

IEEE standard	Standard since	Frequencies 2.4 GHz	5 GHz
802.11	1997	X	
802.11a	1999		X
802.11b	1999	X	
802.11g	2003	X	
802.11h	2003		X
802.11n	2009	X	X
802.11ac	2013		X

IEEE 802.11h is an adaptation of IEEE 802.11a to avoid disturbing military radar systems and satellite radio in Europe. The only differences to IEEE 802.11a are the additional capabilities Dynamic Frequency Selection and Transmission Power Control.

IEEE 802.11h ist eine Anpassung von IEEE 802.11a, damit in Europa unter anderem militärische Radarsysteme und Satellitenfunk nicht gestört werden. Die einzigen Unterschiede zu IEEE 802.11a sind die zusätzlichen Fähigkeiten Dynamic Frequency Selection (*Dynamisches Frequenzwahlverfahren*) und Transmission Power Control (*Übertragungssendeleistungs-Steuerung*).

Although WLAN is used worldwide, there are legal differences. In Germany, for example, the 5.15–5.35 GHz range may only be used within enclosed spaces with a maximum transmission power of 200 mW.

Obwohl WLAN weltweit verwendet wird, gibt es rechtliche Unterschiede. So darf zum Beispiel in Deutschland der Bereich 5,15–5,35 GHz nur innerhalb geschlossener Räume und mit einer maximalen Strahlungsleistung von 200 mW genutzt werden.

The frequency blocks are split into channels, similar to television or radio broadcast-

Die Frequenzblöcke sind in Kanäle unterteilt, vergleichbar mit Fernseh- oder Radioübertra-

ing. The frequency block 2,4000–2,4835 GHz, for example, is divided into 13 channels with a bandwidth of 5 MHz each. In Japan, an additional channel 14 is available, which is restricted for use only with the modulation method DSSS, and it is located 12 MHz above channel 13 (see Table 5.4).

gungen. Der Frequenzblock 2,4000–2,4835 GHz zum Beispiel ist in 13 Kanäle zu je 5 MHz Bandbreite unterteilt. In Japan existiert noch ein zusätzlicher 14. Kanal, der nur für das Modulationsverfahren DSSS freigegeben ist und 12 MHz über dem 13. Kanal liegt (siehe Tabelle 5.4).

Table 5.4: Permitted use of the WLAN Frequencies in the 2.4 GHz Range

Channel	Frequency [GHz]	EU	USA	Japan
1	2.412	X	X	X
2	2.417	X	X	X
3	2.422	X	X	X
4	2.427	X	X	X
5	2.432	X	X	X
6	2.437	X	X	X
7	2.442	X	X	X
8	2.447	X	X	X
9	2.452	X	X	X
10	2.457	X	X	X
11	2.462	X	X	X
12	2.467	X	X	X
13	2.472	X	X	X
14	2.484			X

The various WLAN standards use different modulation methods (see Table 5.5).

Die unterschiedlichen WLAN-Standards verwenden unterschiedliche Modulationsverfahren (siehe Tabelle 5.5).

Table 5.5: Modulation Methods and Channel Widths of the IEEE Standards for WLAN

IEEE standard	Modulation method	Channel width
802.11	FHSS[a] or DSSS[b]	22 MHz
802.11a	OFDM[c]	20 MHz
802.11b	DSSS[b]	22 MHz
802.11g	OFDM[c]	20 MHz
802.11h	OFDM[c]	20 MHz
802.11n	OFDM[c]	20 or 40 MHz
802.11ac	OFDM[c]	20, 40, 80 or 160 MHz

[a] Frequency Hopping Spread Spectrum
[b] Direct Sequence Spread Spectrum
[c] Orthogonal Frequency-Division Multiplexing

Due to the modulation method DSSS of 802.11b with 22 MHz wide channels and a channel spacing of 5 MHz, there are only three (EU and USA) or four (Japan) channels whose signals do not overlap, namely channels 1, 6 and 11, and additionally only in Japan channel 14

Wegen dem Modulationsverfahren DSSS bei 802.11b mit 22 MHz breiten Kanälen und einem Kanalrasterabstand von 5 MHz existieren nur drei (EU und USA) bzw. vier (Japan) Kanäle, deren Signale sich nicht überlappen, nämlich die Kanäle 1, 6 und 11, sowie zusätzlich nur in

(see Figure 5.2). DSSS distributes the payload over a wide frequency range and is therefore almost insensitive to narrowband interferences such as Bluetooth.

The standards 802.11g and 802.11n use the modulation method OFDM. This is a multi-carrier method. Each channel is 20 MHz wide and consists of 64 sub-carriers, which are 0.3125 MHz wide each. Only 52 of the 64 sub-carriers are used. The effective bandwidth per channel is therefore just 16.25 MHz (see Figure 5.3). Therefore, there are only four overlap-free channels 1, 5, 9 and 13.

Terminal devices, which operate according to 802.11a also use the modulation method OFDM with 20 MHz wide channels, but they operate exclusively in the frequency block 5.150–5.725 GHz.

802.11n also supports 40 MHz wide channels. If 40 MHz wide channels are used, only two channels exist in the frequency block 2,4000–2,4835 GHz, namely channel 3 and 11, whose signals do not overlap. Each channel consists of 128 sub-carriers, which are each 0.3125 MHz wide, and of which only 108 are used. The useful bandwidth per channel is therefore just 33.75 MHz (see Figure 5.4).

Many terminal devices, which support 802.11n, can additionally use the frequency block 5.150–5.725 GHz (see Table 5.6).

Signals with higher frequencies have shorter wavelengths. This is why obstacles cause more problems for a wireless network in the 5 GHz range than in the 2.4 GHz range. The reduced wavelength at 5 GHz also affects the transmission power. The shorter the wavelength, the higher is the effect of attenuation caused by the transmission medium. This is why WLAN at 2.4 GHz achieves a greater range with the same transmission power than WLAN at 5 GHz.

Multiple Input Multiple Output (MIMO)

The maximum gross data rate at IEEE 802.11n is 150, 300, 450 or 600 Mbit/s, depending on the number of antennas in the stations. The fact that 802.11n uses *MIMO* causes this performance increase in comparison to IEEE 802.11a/b/g/h. In addition to expanding the channels to 40 MHz, 802.11n uses up

Japan Kanal 14 (siehe Abbildung 5.2). DSSS ist ein Frequenzspreizverfahren, das die Nutzdaten über einen breiten Frequenzbereich verteilt und darum weitgehend unempfindlich ist gegenüber schmalbandigen Störungen wie zum Beispiel Bluetooth.

Die Standards 802.11g und 802.11n verwenden das Modulationsverfahren OFDM. Bei diesem handelt es sich um ein Mehrträgerverfahren. Jeder Kanal ist 20 MHz breit und besteht aus 64 Zwischenträgern zu je 0,3125 MHz, von denen aber nur 52 verwendet werden. Die Nutzbandbreite pro Kanal ist also nur 16,25 MHz (siehe Abbildung 5.3). Es gibt also nur die vier überlappungsfreien Kanäle 1, 5, 9 und 13.

Endgeräte nach 802.11a verwenden auch das Modulationsverfahren OFDM mit 20 MHz breiten Kanälen, arbeiten aber ausschließlich im Frequenzblock 5,150–5,725 GHz.

Bei 802.11n ist wahlweise ein Betrieb mit 40 MHz Kanalbreite möglich. Bei 40 MHz breiten Kanälen existieren im Frequenzblock 2,4000–2,4835 GHz nur zwei Kanäle, nämlich Kanal 3 und 11, deren Signale sich nicht überlappen. Jeder Kanal besteht aus 128 Zwischenträgern zu je 0,3125 MHz, von denen aber nur 108 verwendet werden. Die Nutzbandbreite pro Kanal ist also nur 33,75 MHz (siehe Abbildung 5.4).

Viele Endgeräte, die 802.11n unterstützten, können zusätzlich den Frequenzblock 5,150–5,725 GHz nutzen (siehe Tabelle 5.6).

Signale mit höheren Frequenzen haben kürzere Wellenlängen. Darum bereiten Hindernisse einem Funknetz im Bereich 5 GHz mehr Schwierigkeiten als im Bereich 2,4 GHz. Die reduzierte Wellenlänge bei 5 GHz wirkt sich auch auf die Sendeleistung aus. Je kürzer die Wellenlänge ist, desto stärker wirkt sich die Dämpfung durch das Übertragungsmedium aus. Darum erreicht WLAN bei 2,4 GHz bei gleicher Sendeleistung eine größere Reichweite als WLAN bei 5 GHz.

Multiple Input Multiple Output (MIMO)

Die maximale Bruttodatenrate bei IEEE 802.11n liegt je nach Anzahl der Antennen in den Stationen bei 150, 300, 450 oder 600 Mbit/s. Diese Leistungssteigerung gegenüber IEEE 802.11a/b/g/h ist darauf zurückzuführen, dass 802.11n *MIMO* verwendet. Zusätzlich zur Verbreiterung der Kanäle auf 40 MHz

Figure 5.2: Channels of WLAN 802.11b with 22 MHz Channel Width and the Modulation Method DSSS

Figure 5.3: Channels of WLAN 802.11g/n with 20 MHz Channel Width and the Modulation Method OFDM

Figure 5.4: Channels of WLAN 802.11n with 40 MHz Channel Width and the Modulation Method OFDM

Table 5.6: Permitted use of the WLAN Frequencies in the 5 GHz Range

Channel	Frequency [GHz]	EU	USA	Japan
36	5.180	X[1]	X	X[1]
40	5.200	X[1]	X	X[1]
44	5.220	X[1]	X	X[1]
48	5.240	X[1]	X	X[1]
52	5.260	X[1,2,3]	X[2]	X[1,2,3]
56	5.280	X[1,2,3]	X[2]	X[1,2,3]
60	5.300	X[1,2,3]	X[2]	X[1,2,3]
64	5.320	X[1,2,3]	X[2]	X[1,2,3]
100	5.500	X[2,3]	X[2]	X[2,3]
104	5.520	X[2,3]	X[2]	X[2,3]
108	5.540	X[2,3]	X[2]	X[2,3]
112	5.560	X[2,3]	X[2]	X[2,3]
116	5.580	X[2,3]	X[2]	X[2,3]
120	5.600	X[2,3]	X[2]	X[2,3]
124	5.620	X[2,3]	X[2]	X[2,3]
128	5.640	X[2,3]	X[2]	X[2,3]
132	5.660	X[2,3]	X[2]	X[2,3]
136	5.680	X[2,3]	X[2]	X[2,3]
140	5.700	X[2,3]	X[2]	X[2,3]
144	5.720	X[2,4]	X[2]	—
149	5.745	X[2,4]	X	—
153	5.765	X[2,4]	X	—
157	5.785	X[2,4]	X	—
161	5.805	X[2,4]	X	—
165	5.825	X[2,4]	X	—

[1] Indoor only
[2] Dynamic Frequency Selection (DFS)
[3] Transmit Power Control (TPC)
[4] Short Range Devices (SRD) = 25 mW max.

to four antennas for simultaneous working in the frequency ranges 2.4 GHz and 5 GHz. With each parallel data stream (antenna), a maximum data rate (gross) of 150 Mbps can be achieved. Up to four data streams can be bundled. The corresponding number of antennas (up to four) is required on both sides.

Also, the standard IEEE 802.11ac uses MIMO. This standard even allows a maximum of eight antennas to be used in parallel. In addition, the channels can be extended to 40 MHz, 80 MHz and 160 MHz (see Figure 5.5). In contrast to 802.11n, the 802.11ac standard only operates in the 5 GHz frequency band. The possible maximum (gross) data rate for MIMO with eight antennas on both sides (8x8 MIMO) and

kommen bei 802.11n bis zu vier Antennen zum Einsatz. Diese ermöglichen gleichzeitiges Arbeiten in den Frequenzbereichen 2,4 und 5 GHz. Pro parallelem Datenstrom (Antenne) sind maximal 150 Mbit/s (Brutto-)Datendurchsatz möglich. Bis zu vier Datenströme können gebündelt werden. Die entsprechende Anzahl Antennen (bis zu vier) ist jeweils auf beiden Seiten nötig.

Auch der Standard IEEE 802.11ac verwendet MIMO. Bei diesem sind Standard sind sogar maximal acht parallel nutzbare Antennen möglich. Zudem ist eine Verbreiterung der Kanäle auf 40 MHz, 80 MHz und 160 MHz möglich (siehe Abbildung 5.5). Im Gegensatz zu 802.11n arbeitet 802.11ac ausschließlich im 5-GHz-Band. Die mögliche maximale (brutto) Übertragungsrate bei MIMO mit jeweils acht Antennen (8x8

the 160 MHz channel width is almost 7 Gbps. In practice, access points usually do not have more than three or four antennas. For 3x3 MIMO with 80 MHz channel width, the maximum (gross) data rate is 1.3 Gbps. Doubling the channel width also doubles the maximum gross data rate. Mobile devices usually have only one or two antennas due to their energy consumption.

MIMO) bei den Kommunikationspartner und 160 MHz Kanalbreite ist fast 7 GBit/s. In der Praxis haben die Basisstation meist nicht mehr als drei oder vier Antennen. Bei 3x3 MIMO mit 80 MHz Kanalbreite ist die maximale (brutto) Übertragungsrate 1,3 Gbit/s. Eine Verdopplung der Kanalbreite verdoppelt auch die maximale brutto Übertragungsrate. Mobile Geräte haben wegen des Energieverbrauchs üblicherweise nur eine oder zwei Antennen.

Additional Extensions of the WLAN Standard

Additional WLAN standard extensions exist. Some of them may only be used in specific regions. One example is the standard IEEE 802.11j, which was released in 2004 for Japan and uses the frequency block 4.9–5 GHz. The maximum (gross) data rate is 54 Mbps.

Another example is IEEE 802.11y. This standard was released in 2008 and may only be used in the United States because it uses the frequency block 3.65–3.7 GHz. Due to increased transmission power, ranges of up to 5000 m (outdoor) are possible. The standard is suitable for public WLAN networks, for *last mile* and corporate or university networks to cover a campus. This standard also offers a maximum (gross) data rate of 54 Mbps.

Zusätzliche Erweiterungen des WLAN-Standards

Es existieren zusätzliche Erweiterungen des WLAN-Standards, die zum Teil nur regional verwendet werden (dürfen). Ein Beispiel ist der Standard IEEE 802.11j, der 2004 für Japan verabschiedet wurde und im Frequenzblock 4,9–5 GHz sendet. Die maximale (Brutto-)Datenübertragungsrate ist 54 Mbit/s.

Ein weiteres Beispiel ist IEEE 802.11y. Dieser Standard wurde 2008 verabschiedet und darf nur in den USA genutzt werden, weil er im Frequenzblock 3,65–3,7 GHz sendet. Durch eine höhere Sendeleistung sind Reichweiten im Freien bis zu 5000 m möglich. Der Standard eignet sich unter anderem für städtische WLAN-Netze, die *letzte Meile* und Unternehmens- oder Hochschulnetze zur Abdeckung eines Campus. Auch bei diesem Standard ist die maximale (Brutto-)Datenübertragungsrate 54 Mbit/s.

WLAN Security

WLAN according to IEEE 802.11 implements the security standard *Wired Equivalent Privacy* (WEP), which is based on the RC4 algorithm. The algorithm calculates the XOR of the payload bit stream and the pseudo-random bit stream, which is generated from the RC4 algorithm. The algorithm works with 40 Bit or 104 Bit long static keys. Known-plaintext attacks can crack the algorithm because the headers of the IEEE 802.11 protocol are predictable. The calculation of the WEP key using a few minutes of recorded data takes only a few seconds with tools like Aircrack.

Better security is provided by the standard *Wi-Fi Protected Access* (WPA). This is also

Sicherheit bei WLAN

WLAN gemäß IEEE 802.11 enthält den Sicherheitsstandard *Wired Equivalent Privacy* (WEP), der auf dem RC4-Algorithmus basiert. Bei diesem findet eine XOR-Verknüpfung des Bitstroms der Nutzdaten mit einem aus dem RC4-Algorithmus generierten, pseudozufälligen Bitstrom statt. Der Algorithmus arbeitet mit 40 Bit bzw. 104 Bit langen statischen Schlüsseln. Das Verfahren kann durch Known-Plaintext-Angriffe geknackt werden, weil die Header des IEEE 802.11-Protokolls vorhersagbar sind. Die Berechnung des WEP-Schlüssels mit Hilfe von einigen Minuten aufgezeichneter Daten dauert mit Werkzeugen wie Aircrack nur wenige Sekunden.

Mehr Sicherheit bietet der Sicherheitsstandard *Wi-Fi Protected Access* (WPA). Auch die-

Figure 5.5: Channels of WLAN 802.11ac with different Channel Width Settings and the Modulation Method OFDM

based on the RC4 algorithm but contains additional protection through dynamic keys. These are based on the Temporal Key Integrity Protocol (TKIP), which encrypts each data packet with a different key. WPA can also be cracked with brute force methods or with dictionary attacks on the password used.

The best current security standard is *Wi-Fi Protected Access 2* (WPA2). It is based on the Advanced Encryption Standard (AES) and in addition to TKIP contains the Counter-Mode/CBC-Mac Protocol (CCMP) encryption protocol, which offers higher security than TKIP. A WLAN protected with a sufficiently long password and WPA2 encryption is currently considered secure.

WPA and WPA2 both are defined in the IEEE 802.11i standard, which extends IEEE 802.11.

ser basiert auf dem RC4-Algorithmus, enthält jedoch zusätzlichen Schutz durch dynamische Schlüssel. Diese basieren auf dem Temporal Key Integrity Protocol (TKIP), das jedes Datenpaket mit einem anderen Schlüssel verschlüsselt. WPA kann mit der Brute-Force-Methode oder mit Wörterbuchangriffen auf das benutzte Passwort geknackt werden.

Der gegenwärtig beste Sicherheitsstandard ist *Wi-Fi Protected Access 2* (WPA2). Er basiert auf dem Advanced Encryption Standard (AES) und enthält zusätzlich zu TKIP das Verschlüsselungsprotokoll Counter-Mode/CBC-Mac Protocol (CCMP), das eine höhere Sicherheit als TKIP bietet. Ein mit einem ausreichend langen Passwort geschütztes WLAN mit WPA2-Verschlüsselung gilt gegenwärtig als sicher.

WPA und WPA2 sind beide im Standard IEEE 802.11i definiert, der IEEE 802.11 erweitert.

5.1.4 Bluetooth

Bluetooth is a wireless network system for data transmission over short distances. The Swedish company Ericsson initiated its development in 1994. Further development is carried out by the Bluetooth Special Interest Group (BSIG). Bluetooth was developed to replace short cable connections between different devices.

Bluetooth devices may be operated worldwide without approval and use the frequency block 2.402–2.480 GHz. WLANs, cordless telephones and microwave ovens can cause interferences if they operate in the same frequency band. To avoid interferences, Bluetooth uses a frequency hopping method in which the frequency band is divided into 79 frequency stages at intervals of 1 MHz. The frequency levels are changed up to 1600 times per second [14][26].

Bluetooth defines three performance classes. Class 1 devices operate with a maximum transmission power of 100 mW. Class 2 devices use a maximum transmission power of 2,5 mW. Class 3 devices use a maximum transmission power of 1 mW.

Bluetooth

Bluetooth ist ein Funksystem zur Datenübertragung auf kurzen Distanzen. Die Entwicklung wurde von der schwedischen Firma Ericsson 1994 begonnen. Die Weiterentwicklung erfolgt durch die Interessengemeinschaft Bluetooth Special Interest Group (BSIG). Bluetooth wurde entwickelt, um kurze Kabelverbindungen zwischen verschiedenen Geräten zu ersetzen.

Bluetooth-Geräte dürfen weltweit zulassungsfrei betrieben werden und senden im Frequenzblock 2,402–2,480 GHz. WLANs, schnurlose Telefone oder Mikrowellenherde können Störungen verursachen, wenn Sie im gleichen Frequenzband arbeiten. Um Störungen zu vermeiden, verwendet Bluetooth ein Frequenzsprungverfahren, bei dem das Frequenzband in 79 verschiedene Frequenzstufen im Abstand von je 1 MHz eingeteilt wird. Die Frequenzstufen werden bis zu 1600 mal pro Sekunde gewechselt [14][26].

Bluetooth definiert drei Leistungsklassen. Endgeräte der Klasse 1 haben eine maximale Sendeleistung von 100 mW. Endgeräte der Klasse 2 haben eine maximale Sendeleistung von 2,5 mW und Endgeräte der Klasse 3 eine maximale Sendeleistung von 1 mW.

Network Topologies of Bluetooth

Via Bluetooth, connectionless and connection-oriented transmissions from point-to-point networks or as ad-hoc networks (so-called piconets) are possible.

Bluetooth devices are organized in so-called *Piconets* (see Figure 5.6). A piconet consists of a maximum of 255 participants, of which a maximum of eight may be active. One active node is the *master*, and the remaining seven active nodes are *slaves*. The remaining 247 participants are passive and can be activated by the master at any time. The master assigns the transmission medium to the nodes by providing transmission slots to the slaves. This procedure is called Time Division Multiplexing. Thus, the master coordinates the media access. [14]

Netzwerk-Topologien bei Bluetooth

Via Bluetooth sind verbindungslose sowie verbindungsorientierte Übertragungen von Punkt zu Punkt oder als Ad-hoc-Netze (sogenannte Piconetze) möglich.

Bluetooth-Geräte organisieren sich in sogenannten *Piconetzen* (siehe Abbildung 5.6). Ein Piconetz besteht aus maximal 255 Teilnehmern, wovon maximal acht aktiv sein dürfen. Ein aktiver Teilnehmer ist der *Master* und die restlichen sieben sind *Slaves*. Die übrigen 247 Teilnehmer sind passiv und können jederzeit vom Master aktiviert werden. Der Master teilt das Übertragungsmedium unter den Teilnehmern auf, indem er Sendeslots an die Slaves vergibt. Dieses Vorgehen heißt Zeitmultiplexverfahren (Time Division Multiplexing). Der Master koordiniert also den Medienzugriff. [14]

Figure 5.6: Bluetooth Topologies

If the number of participants increases, the bandwidth available to each participant decreases. A Bluetooth device can be registered in several piconets. However, it can only be the master of a single network. If a station is in range of two piconets, it can connect them to a *scatternet*. Up to ten piconets form a scatternet. Each piconet is identified by the different sequences in the frequency hopping method. The data rates of scatternets are usually low.

Steigt die Anzahl der Teilnehmer, sinkt die Bandbreite, die jeder Teilnehmer zur Verfügung hat. Ein Bluetooth-Gerät kann in mehreren Piconetzen angemeldet sein. Es kann aber nur in einem Netz der Master sein. Ist ein Teilnehmer im Empfangsbereich zweier Piconetze, kann er diese zu einem *Scatternetz* zusammenschließen. Bis zu zehn Piconetze bilden ein Scatternet. Jedes Piconet wird durch die unterschiedlichen Wechselfolgen im Frequenzsprungverfahren identifiziert. Die Datentransferraten in Scatternets sind meist gering.

Versions of the Bluetooth Standards

There are several versions of the Bluetooth standard.

Versionen des Bluetooth-Standards

Es gibt verschiedene Versionen des Bluetooth-Standards.

While Bluetooth up to revision 1.2 offered a maximum data rate of 1 Mbps (of which 721 kbit is payload), the data rate was increased to 3 Mbps (of which 2,1 Mbps is payload) with Bluetooth revision 2.0.

Bluetooth 3.0 + HS (High Speed) increases the maximum data transfer rate by using WLAN. The standard uses a 3 Mbps connection to transmit control data and session keys. If two devices want to exchange large amounts of data, they switch into the high-speed mode and establish an ad-hoc connection via WLAN 802.11g with a 54 Mbps data rate. Then the achievable (net) data rate is about 24 Mbps. Hence, Bluetooth 3.0 + HS is a combination of Bluetooth and WLAN.

Bluetooth 4 offers – among other things – a reduced power consumption. Bluetooth 5 raises the maximum range and improves the data rate.

Pairing of Bluetooth Devices

Before two Bluetooth devices can communicate with each other, they must know each other. The process of *getting to know each other* is called *pairing*. Up to Bluetooth 2.0, pairing is time-consuming because the users of both terminal devices must enter the same PIN. This PIN is the shared key for encryption and authentication. This ensures that no other device can monitor the connection or launch a *man-in-the-middle attack*. However, the pairing must only be carried out once for two devices.

Bluetooth 2.1 introduced *Secure Simple Pairing*. This method uses the Diffie-Hellmann algorithm for key distribution instead of a PIN. The security of this pairing method depends on whether the terminal devices have a display. If both devices have a display, users must confirm a common code by pressing a key. For devices that do not have a display to show the code, the confirmation is skipped. A man-in-the-middle attack is still possible with this method when no confirmation is used.

Bot Bluetooth bis Version 1.2 eine maximale Datentransferrate von 1 Mbit/s (davon 721 kbit Nutzdaten), stieg die Datentransferrate bei Bluetooth 2.0 auf 3 Mbit/s (davon 2,1 Mbit/s Nutzdaten).

Bluetooth 3.0 + HS (High Speed) steigert die maximale Datentransferrate durch einen Rückgriff auf WLAN. Der Standard verwendet eine Verbindung mit 3 Mbit/s zur Übertragung der Steuerdaten und Sitzungsschlüssel. Wollen zwei Endgeräte große Datenmengen austauschen, schalten diese in den Highspeed-Modus und bauen eine Ad-hoc-Verbindung via WLAN 802.11g mit 54 Mbit/s auf. Die erreichbare (Netto-) Datentransferrate ist dann ca. 24 Mbit/s. Bluetooth 3.0 + HS ist also eine Kombination aus Bluetooth und WLAN.

Bluetooth 4 bietet unter anderem einen reduzierten Stromverbrauch. Bluetooth 5 erhöht die maximale Reichweite und verbessert die Datenrate.

Pairing von Bluetooth-Geräten

Bevor zwei Endgeräte via Bluetooth miteinander kommunizieren können, müssen sie sich kennen. Der Vorgang des *Kennenlernens* heißt *Pairing*. Bis einschließlich Bluetooth 2.0 ist das Pairing aufwändig, denn die Benutzer beider Endgeräte müssen eine identische PIN eingeben. Diese PIN ist der gemeinsame Schlüssel für die Verschlüsselung und Authentifizierung. Damit wird sichergestellt, dass kein drittes Gerät die Verbindung mithören bzw. einen *Man-in-the-Middle-Angriff* ausüben kann. Das Pairing muss aber nur einmalig für zwei Geräte ausgeführt werden.

Mit Bluetooth 2.1 wurde das *Secure Simple Pairing* eingeführt. Dieses Verfahren verwendet den Diffie-Hellmann-Algorithmus zur Schlüsselverteilung anstatt einer PIN. Die Sicherheit dieses Pairing-Verfahrens hängt davon ab, ob die Endgeräte ein Display haben. Haben beide Endgeräte ein Display, müssen die Benutzer jeweils einen gemeinsamen Code durch Tastendruck bestätigen. Für Geräte, die kein Display zum Anzeigen des Codes haben, entfällt die Bestätigung. Ein Man-in-the-Middle-Angriff ist bei diesem Verfahren ohne Bestätigung nach wie vor möglich.

5.2

Transmission Media

Various cable-based transmission media for computer networks exist. Copper cables are *electrical conductors* on which data is transmitted via twisted pair cables or coaxial cables in the form of electrical impulses, and there are *fiber-optic cables* on which data is transmitted as light impulses.

Moreover, *wireless transmission* can be used in the *directional* or *undirected* form. Directional wireless transmission can base on the following technologies:

- *Radio technology*: Data is transmitted as electromagnetic waves (radio waves) in the radio frequency range. Examples are WLAN and transmissions from satellites.
- *Infrared*: Data is transmitted as electromagnetic waves in the invisible spectrum. One example is IrDA.
- *Laser*: Data is transmitted as light impulses via a Laser-Bridge.

Undirected wireless transmission is always based on radio technology. Examples of applications include mobile telephony, LTE, terrestrial broadcasting and satellite broadcasting.

There are three types of wired transmission media in Ethernet: coaxial cables, twisted-pair cables, and fiber-optic cables.

Übertragungsmedien

Es existieren verschiedene *leitungsgebundene Übertragungsmedien* für Computernetze. Es gibt *elektrische Leiter* aus Kupfer, auf denen Daten über Twisted-Pair-Kabel (verdrillte Kabel) oder Koaxialkabel in Form elektrischer Impulse übertragen werden, und es gibt *Lichtwellenleiter*, auf denen Daten als Lichtimpulse übertragen werden.

Zudem gibt es die Möglichkeit *nichtleitungsgebundener Übertragung* (drahtlose Übertragung), die *gerichtet* und *ungerichtet* möglich ist. Gerichtete drahtlose Übertragung kann auf folgenden Technologien basieren:

- *Funktechnik*: Daten werden als elektromagnetische Wellen (Radiowellen) im Radiofrequenzbereich übertragen. Beispiele sind WLAN und Übertragungen von Satelliten.
- *Infrarot*: Daten werden als elektromagnetische Wellen im Bereich des unsichtbaren Spektrums übertragen. Ein Beispiel ist IrDA.
- *Laser*: Daten werden via Laser-Bridge als Lichtimpulse übertragen.

Ungerichtete Übertragung basiert immer auf Funktechnik. Anwendungsbeispiele sind Mobilfunk, LTE, terrestrischer Rundfunk und Satelliten-Rundfunk.

Ethernet kennt drei Arten leitungsgebundener Übertragungsmedien: Koaxialkabel, Twisted-Pair-Kabel und Lichtwellenleiter.

5.2.1

Coaxial Cables

Coaxial cables are bipolar cables with a concentric (coaxial) structure. The inner conductor carries the signals and the outer conductor is on ground potential, surrounding the inner conductor (see Figure 5.7). The shielding of the signal-carrying conductor by the outer conductor that is kept at ground potential reduces electromagnetic interference.

Koaxialkabel

Koaxialkabel sind zweipolige Kabel mit konzentrischem (koaxialem) Aufbau. Der innere Leiter (*Seele*) führt das Signal und der äußere Leiter liegt auf Masse (Grundpotential) und umhüllt den inneren vollständig (siehe Abbildung 5.7). Wegen der Abschirmung des signalführenden Leiters durch die Umhüllung mit der Masse werden elektromagnetische Störungen reduziert.

Figure 5.7: Structure of a Coaxial Cable

The Ethernet standards Thick Ethernet and Thin Ethernet both use coaxial cables as the transmission medium.

Thick Ethernet (10BASE5), which is also called *Yellow Cable*, uses 10 mm thick coaxial cables (RG-8) with 50 Ohms impedance. To connect a terminal device, a hole must be drilled in the cable through the outer shielding to contact the inner conductor. Through the hole, the *transceiver* (see Figure 5.8) is connected via a *vampire tap* with the inner conductor, and the terminal device is connected to the transceiver via a transceiver cable (DB15) – also called AUI (Attachment Unit Interface)

Die beiden Ethernet-Standards Thick Ethernet und Thin Ethernet verwenden Koaxialkabel als Übertragungsmedium.

Bei *Thick Ethernet* (10BASE5), das auch *Yellow Cable* genannt wird, werden 10 mm dicke Koaxialkabel (RG-8) mit 50 Ohm Wellenwiderstand verwendet. Zum Anschluss eines Endgeräts muss ein Loch in das Kabel gebohrt werden. Durch das Loch wird über eine *Vampirklemme* der *Transceiver* (siehe Abbildung 5.8) mit der Seele verbunden und das Endgerät wird über ein Transceiver Kabel (DB15) – auch AUI (Attachment Unit Interface) genannt – mit dem Transceiver verbunden.

Figure 5.8: Transceiver of Thick Ethernet (10BASE5)

The hardware for Thick Ethernet is cost-intensive. A cheaper solution is Thin Ethernet (10BASE2), which is also called *Cheapernet*

Der Hardwareaufwand bei Thick Ethernet ist kostenintensiv. Eine preisgünstigere Lösung ist Thin Ethernet (10BASE2), das auch *Chea-*

or *ThinWire*. This standard uses 6 mm thick coaxial cable (RG-58) with 50 Ohms impedance. The cables are thinner and more flexible, and therefore more comfortable to lay. The cables and network devices have BNC connectors (Bayonet Neill Concelman). With connectors, which are called *T-Connectors* because of their appearance, the terminal devices are connected to the transmission medium (see Figure 5.9). Terminators prevent signal reflections on the medium (see Figure 5.10).

pernet oder *ThinWire* genannt wird. Dieser Standard verwendet 6 mm dicke Koaxialkabel (RG-58) mit 50 Ohm Wellenwiderstand. Die Kabel sind dünner und flexibler und dadurch einfacher zu verlegen. Die Kabel und Netzwerkgeräte haben BNC-Anschlüsse (Bayonet Neill Concelman). Mit Verbindungssteckern, die wegen ihres Aussehens *T-Stücke* heißen, verbindet man die Endgeräte mit dem Übertragungsmedium (siehe Abbildung 5.9). Abschlusswiderstände verhindern Reflexionen auf dem Medium (siehe Abbildung 5.10).

Figure 5.9: Cabling of Thin Ethernet (10BASE2)

Figure 5.10: T-Connectors and Terminators of Thin Ethernet (10BASE2)

5.2.2

Twisted Pair Cables

The wires of *twisted pair cables* are twisted pairwise with each other. Twisted wire pairs offer better protection against alternating magnetic fields and electrostatic influences from the outside than parallel signal wires. A *complementary signal* is sent via each pair of wires (on one wire 0 V to +2,5 V and on the other wire 0 V to −2,5 V). This allows the receiver to filter out interfering signals. Furthermore, it reduces

Twisted-Pair-Kabel

Die Adern von *Twisted-Pair-Kabeln* sind paarweise miteinander verdrillt. Verdrillte Adernpaare bieten besseren Schutz gegen magnetischen Wechselfelder und elektrostatische Beeinflussungen von außen als Adern, die nur parallel geführt sind. Über das Adernpaar wird jeweils ein *Komplementärsignal* gesendet (auf einer Ader 0 V bis +2,5 V und auf der anderen Ader 0 V bis −2,5 V). So kann der Empfänger Leitungsstörungen her-

electromagnetic emissions [11]. The operation principle is demonstrated in Figure 5.11.

The signal level of line A is equal to the sum of the signal levels of the payload signal and the noise. The signal level of line B is equal to the sum of the signal levels of the negative payload signal and the noise.

$$A = \text{PayloadSignal} + \text{Noise}$$
$$B = -\text{PayloadSignal} + \text{Noise}$$

Regardless of the level of the noise signal, the difference between the payload signal and the complementary signal remains the same. The difference of the signal levels of line A and line B at the receiver side is [16]:

$$[+\text{Payload Signal} + \text{Noise}]$$
$$-[-\text{Payload Signal} + \text{Noise}]$$
$$= 2 * \text{Payload Signal}$$

ausfiltern. Zudem wird die elektromagnetische Abstrahlung reduziert [11]. Das Funktionsprinzip zeigt Abbildung 5.11.

Die Signalamplitude von Leitung A entspricht der Summe aus Signalamplituden von Nutzsignal und Störsignal. Die Signalamplitude von Leitung B entspricht der Summe aus den Signalamplituden vom negierten Nutzsignal und dem Störsignal.

$$A = \text{Nutzsignal} + \text{Störsignal}$$
$$B = -\text{Nutzsignal} + \text{Störsignal}$$

Unabhängig von der Höhe des Störsignals bleibt die Differenz zwischen Nutzsignal und Komplementärsignal gleich. Die Differenz der Signalamplituden von Leitung A und von Leitung B beim Empfänger ist [16]:

$$[+\text{Nutzsignal} + \text{Störsignal}]$$
$$-[-\text{Nutzsignal} + \text{Störsignal}]$$
$$= 2 * \text{Nutzsignal}$$

Figure 5.11: When transmitting Data via Twisted Pair Cables, each Pair of Wires transmits Complementary Signals

All variants of the Ethernet standard, where twisted pair cables are the transmission medium, use connectors and sockets according to the standard 8P8C, which are usually called RJ45

Alle Varianten des Ethernet-Standards, bei denen Twisted-Pair-Kabel das Übertragungsmedium sind, verwenden Stecker und Buchsen nach dem Standard 8P8C, die meist RJ45

(see Figure 5.12). In the 1990s, twisted pair cables and RJ45 connectors and sockets have been established as the standard for copper-based IT networking.

genannt werden (siehe Abbildung 5.12). Seit den 1990er Jahren sind Twisted-Pair-Kabel und RJ45-Stecker und -Buchsen als Standard für kupferbasierte IT-Vernetzung etabliert.

Figure 5.12: 8P8C Connector (RJ45)

Ethernet 10BASE-T and Fast Ethernet 100BASE-TX use only one of the four wire pairs for sending and one for receiving (see Figure 5.13). `TD+` and `TD-` (Transceive Data) is the wire pair for the data output and `RD+` and `RD-` (Receive data) is the wire pair for data input. The remaining wire pairs are not used. Fast Ethernet 100BASE-T4 and Gigabit Ethernet 1000BASE-T, as well as all more modern standards, use all four wire pairs for sending and receiving.

Ethernet 10BASE-T und Fast-Ethernet 100BASE-TX verwenden von den vier Adernpaaren nur eins zum Senden und eins zum Empfangen (siehe Abbildung 5.13). `TD+` und `TD-` (Transceive Data) sind das Signalpaar für den Datenausgang und `RD+` und `RD-` (Receive Data) das Signalpaar für den Dateneingang. Die übrigen Adernpaare werden nicht verwendet. Fast-Ethernet 100BASE-T4 und Gigabit-Ethernet 1000BASE-T sowie alle nachfolgenden Standards verwenden jeweils alle vier Adernpaare zum Senden und zum Empfangen.

Figure 5.13: 8P8C Pinout (RJ45) of Ethernet

T568A and T568B are standards for the pin assignment of the RJ45 connectors and sockets and are used for Ethernet 10BASE-T, Fast-Ethernet 100BASE-TX, Gigabit-Ethernet 1000BASE-T, 2.5GBASE-T, 5GBASE-T, 10GBASE-T as well as 40GBASE-T.

The only difference between the standards is that the wire pairs 2 and 3 (green and orange) are interchanged. In Europe, T568A is usually used for wiring.

T568A und T568B sind Standards für die Pinbelegung der RJ45-Stecker und -Buchsen und werden bei Ethernet 10BASE-T, Fast-Ethernet 100BASE-TX, Gigabit-Ethernet 1000BASE-T, Gigabit-Ethernet 1000BASE-T, 2.5GBASE-T, 5GBASE-T, 10GBASE-T sowie 40GBASE-T verwendet.

Der einzige Unterschied zwischen den Standards ist die Vertauschung der beiden Adernpaare 2 und 3 (grün und orange). In Europa wird meistens nach T568A verkabelt.

Crossover Cables and Patch Cables

Two terminal devices can be connected directly with a *crossover cable* (see Figure 5.14). It connects the send and receive lines of both devices.

Crossover-Kabel und Patch-Kabel

Zwei Endgeräte verbindet man direkt via ein *Crossover-Kabel* (siehe Abbildung 5.14). Es verbindet die Dateneingänge und -ausgänge von Geräten miteinander.

Figure 5.14: Pinout of Patch Cables and Crossover Cables

More than two network devices are connected with *patch cables* (1 : 1 wiring). In this case, a Hub or a Switch is required. Some Hubs and Switches have an *Uplink port* (see Figure 5.15) to connect them to another Hub or Switch. In contrast to the other ports, the Uplink port is internally cross-connected.

Mehr als zwei Netzwerkgeräte vernetzt man mit *Patch-Kabeln* (1 : 1-Verdrahtung). In diesem Fall benötigt man einen Hub oder Switch. Einige Hubs und Switches haben einen *Uplink-Port* (siehe Abbildung 5.15) zur Verbindung mit einem weiteren Hub oder Switch. Der Uplink-Port

ist im Gegensatz zu den übrigen Ports intern gekreuzt.

Figure 5.15: If a Hub or Switch is equipped with an Uplink Port, it is internally crossed

Modern network devices automatically detect the transmit and receive lines of connected network devices. This feature is called *Auto-MDIX* and it made crossover cables obsolete. Nowadays, only patch cables are used. All network devices, which support Gigabit Ethernet 1000BASE-T or faster, implement Auto-MDIX.

Moderne Netzwerkgeräte erkennen selbstständig die Sende- und Empfangsleitungen verbundener Netzwerkgeräte. Diese Fähigkeit heißt *Auto-MDIX* und es machte Crossover-Kabel überflüssig. Üblicherweise werden nur noch Patch-Kabel verwendet. Alle Netzwerkgeräte, die Gigabit-Ethernet 1000BASE-T oder schneller beherrschen, unterstützen Auto-MDIX.

Table 5.7: Variants of Twisted Pair Cables

Label	Meaning (TP = Twisted Pair)	Cable shielding	Pair shielding
`UUTP`	Unshielded TP	none	none
`UFTP`	Foiled TP	none	foil
`USTP`	Shielded TP	none	braiding
`SUTP`	Screened Unshielded TP	braiding	none
`SFTP`	Screened Foiled TP	braiding	foil
`SSTP`	Screened Shielded TP	braiding	braiding
`FUTP`	Foiled Unshielded TP	foil	none
`FFTP`	Foiled Foiled TP	foil	foil
`FSTP`	Foiled Shielded TP	foil	braiding
`SFUTP`	Screened Foiled Unshielded TP	braiding and foil	none
`SFFTP`	Screened Foiled Foiled TP	braiding and foil	foil

Shielding of different Twisted Pair Cables

An electrically conductive shield offers additional protection from external electromagnetic fields (see Table 5.7). The naming scheme is `XXYZZ`.

Schirmung bei Twisted-Pair-Kabeln

Ein elektrisch leitender Schirm bietet zusätzlich Schutz gegen äußere elektromagnetische Felder (siehe Tabelle 5.7). Das Bezeichnungsschema hat die Form `XXYZZ`.

- `XX` indicates the cable shielding (`U` = unshielded, `F` = foil shielding, `S` = braided shielding, `SF` = braided and foil shielding)
- `Y` indicates the wire pair shielding (`U` = unshielded, `F` = foil shielding, `S` = braided shielding)
- `ZZ` stands for twisted pair (TP)

The shields must be electrically grounded on both sides of the cable. One-sided grounding causes an antenna effect (see Figure 5.16).

- `XX` steht für die Gesamtschirmung (`U` = ungeschirmt, `F` = Folie, `S` = Drahtgeflecht, `SF` = Drahtgeflecht und Folie)
- `Y` steht für die Aderpaarschirmung (`U` = ungeschirmt, `F` = Folie, `S` = Drahtgeflecht)
- `ZZ` steht für Twisted Pair (`TP`)

Die Schirme müssen auf beiden Seiten des Kabels geerdet sein. Einseitige Erdung führt zu Antennenwirkung (siehe Abbildung 5.16).

Figure 5.16: Shielding between Buildings causes an Antenna Effect

This results in a compensation current between the systems ($I = \frac{V}{R}$). The existence of this compensating current leads to malfunctions during operation or even to the destruction of network devices. Shielding therefore only makes sense if both sides share the same ground potential. For this reason, shielded cables should never be laid between buildings. Possible solutions include the installation of fiber-optic cables between buildings, the use of Laser-Bridges or wireless networks.

Es kommt zum Ausgleichsstrom zwischen den Systemen ($I = \frac{U}{R}$). Die Existenz dieses Ausgleichsstroms führt zu Störungen im Betrieb oder gar zur Zerstörung von Netzwerkgeräten. Schirmung ist also nur dann sinnvoll, wenn beide Seiten auf demselben Erdungspotenzial liegen. Darum sollten Kabel mit Schirmung niemals zwischen Gebäuden verlegt werden. Lösungsmöglichkeiten sind das Verlegen von Lichtwellenleitern zwischen Gebäuden, die Verwendung von Laser-Bridges oder von Funknetzen.

Categories of Twisted Pair Cables

Different performance levels (categories) of twisted pair cables exist (see Table 5.8). The component of the lowest category determines the performance of a network connection. For example, if Cat 6-capable network devices are connected via a Cat 5 cable, the performance of the connection is reduced to Cat 5.

Cat 1 cables are unshielded telephone cables that can be used for telephone modems. Cables for ISDN infrastructures belong to Cat 2.

Kategorien von Twisted-Pair-Kabeln

Es existieren Twisted-Pair-Kabel (siehe Tabelle 5.8) unterschiedlicher Leistungsfähigkeit (Kategorie). Die Leistungsfähigkeit einer Netzwerkverbindung wird von der Komponente mit der geringsten Kategorie (Cat) bestimmt. Sind zum Beispiel Cat-6-fähige Netzwerkgeräte über ein Cat-5-Kabel verbunden, reduziert sich die Leistungsfähigkeit der Verbindung auf Kategorie 5.

Kabel der Kategorie 1 sind für Modems geeignete Telefonkabel ohne Abschirmung. Einfache ISDN-Kabel entsprechen der Kategorie 2. Ka-

Table 5.8: Categories of Twisted Pair Cables

Category	Max. frequency	Compatible with...
Cat-1	100 kHz	Modem
Cat-2	1 oder 1.5 MHz	ISDN
Cat-3	16 MHz	10BASE-T (10 Mbps, 2 wire pairs, 100 m)
		100BASE-T4 (100 Mbps, 4 wire pairs, 100 m)
Cat-4	20 MHz	Token Ring (16 Mbps, 2 wire pairs)
Cat-5	100 MHz	100BASE-TX (100 Mbps, 2 wire pairs, 100 m)
		1000BASE-T (1 Gbps, 4 wire pairs, 100 m)
Cat-5e	100 MHz	2.5GBASE-T (2.5 Gbps, 4 wire pairs, 100 m)
Cat-6	250 MHz	5GBASE-T (5 Gbps, 4 wire pairs, 100 m)
		10GBASE-T (10 Gbps, 4 wire pairs, 55 m)
Cat-6A	500 MHz	10GBASE-T (10 Gbps, 4 wire pairs, 100 m)
Cat-7	600 MHz	10GBASE-T (10 Gbps, 4 wire pairs, 100 m)
Cat-7A	1000 MHz	10GBASE-T (10 Gbps, 4 wire pairs, 100 m)
Cat-8.1	2000 MHz	40GBASE-T (40 Gbps, 4 wire pairs, 30 m)

Cat 3 cables allow data transfers over 10 Mbit Ethernet (10BASE-T) and over 100 Mbit Ethernet (100BASE-T4), which uses four wire pairs. These cables are seldom used today, except for telephone/ISDN. Cat 4 cables are not widely used, because their performance gain is small compared to Cat 3. Cat 5 cables allow Fast Ethernet (100BASE-TX) and Gigabit Ethernet (1000BASE-T) data transfers by using four wire pairs. Cables of Cat 5e are guaranteed Gigabit Ethernet-compatible and meet stricter test standards than Cat 5 cables. Also, these cables support 2.5 Gbps Ethernet (2.5GBASE-T). Cables of this category are the most common cabling for Ethernet computer networks. Cat 6 cables can also be used for Gigabit Ethernet (1000BASE-T) and support 10 Gigabit Ethernet (10GBASE-T) with cable segments of up to 55 m. Cat 6A cables can be used for 10 Gigabit Ethernet (10GBASE-T) with cable segments of up to 100m in length.

For Cat 7 and Cat 7A cables, other connectors (e.g., TERA or GG45) and sockets than RJ45 were initially intended. However, these connectors were not successful in the market. Cat 7 and 7A cabling with RJ45 connectors offers no benefits over category 6A cables.

Cat 8.1 cables can be used for 40 Gigabit Ethernet (40GBASE-T) with RJ45 connectors.

bel der Kategorie 3 ermöglichen Datenübertragungen via 10 Mbit-Ethernet (10BASE-T) und via 100 Mbit-Ethernet (100BASE-T4), das vier Aderpaare verwendet. Diese Kabel sind außer für Telefon/ISDN kaum noch verbreitet. Kabel der Kategorie 4 sind wenig verbreitet, da der Gewinn gegenüber Kategorie 3 gering ist. Kabel der Kategorie 5 ermöglichen Datenübertragungen via Fast-Ethernet (100BASE-TX) und via Gigabit-Ethernet (1000BASE-T), das vier Aderpaare verwendet. Kabel der Kategorie 5e sind garantiert Gigabit-Ethernet-tauglich und erfüllen strengere Prüfstandards als Kabel der Kategorie 5. Zudem unterstützten diese Kabel auch 2,5 Gbit-Ethernet (2.5GBASE-T). Kabel der Kategorie 5e sind die häufigste Verkabelung für Ethernet-Computernetze. Kabel der Kategorie 6 sind ebenfalls für Gigabit-Ethernet (1000BASE-T) geeignet und unterstützen 10-Gigabit-Ethernet (10GBASE-T) mit Kabelsegmenten bis 55 m. Kabel der Kategorie 6A eignen sich für 10-Gigabit-Ethernet (10GBASE-T) mit bis zu 100 m langen Kabelsegmenten.

Für Kabel der Kategorien 7 und 7A waren ursprünglich andere Stecker (z.B. TERA oder alternativ GG45) und Buchsen als RJ45 vorgesehen. Diese Stecker konnten sich am Markt aber nicht durchsetzen. Eine Verkabelung der Kategorien 7 und 7A bietet mit RJ45-Steckern keine Vorteile gegenüber Kabeln der Kategorien 6A.

Kabel der Kategorie 8.1 eignen sich für 40-Gigabit-Ethernet (40GBASE-T) mit RJ45-

This standard supports cables of up to 30 m in length.

Steckern. Dieser Standard unterstützt Kabel bis zu einer Länge von 30 m.

Power over Ethernet

PoE stands for a group of protocols that describes how network devices can be supplied with electric energy via twisted pair cables. Especially for devices such as IP cameras, IP telephones, WLAN access points and single board computers that are sometimes operated in places without a power connection in range, it is a significant benefit if only the network cable has to be laid.

The standard that is most often used in practice is IEEE 802.3af. When using Ethernet 10BASE-T and Fast Ethernet 100BASE-TX, this can use the unused wire pairs of Ethernet 10BASE-T and Fast Ethernet 100BASE-TX for power supply. As an alternative, a direct current is transmitted in addition to the data signal on the signal-carrying wire pairs [11].

With IEEE 802.3af, the devices to be powered, which are called *Powered Device* (PD) in the PoE-context, are allowed to consume 12.95 Watts at a supply voltage of 48 V and a current of 350 mA. Newer PoE standards like IEEE 802.3at (PoE+) and IEEE 802.3bt (4PPoE) provide more PoE power but are less common in practice.

The injection of the electrical energy to a twisted pair cable can be done by using a *Power Sourcing Equipment* (PSE), also called *Endspan*, or via a *Midspan*. A PSE or Endspan is a Switch that supports PoE. A Midspan is a so-called PoE Injector, which is located between the Switch and the device to be supplied (PE).

Power over Ethernet

PoE steht für eine Gruppe von Protokollen, die beschreibt wie Netzwerkgeräte über Twisted-Pair-Kabel mit Strom versorgt werden können. Besonders bei Geräten wie IP-Kameras, IP-Telefonen, WLAN-Basisstationen (Access Points) und Einplatinencomputern, die gelegentlich an Stellen ohne direkten Stromanschluss betrieben werden sollen, ist es von großem Vorteil, wenn nur das Netzwerkkabel verlegt werden muss.

Der in der Praxis am häufigsten verwendete Standard ist IEEE 802.3af. Dieser kann die bei Ethernet 10BASE-T und Fast-Ethernet 100BASE-TX ungenutzten Adernpaare zur Stromversorgung nutzen. Alternativ wird auf den signalführenden Adernpaaren zusätzlich zum Datensignal ein Gleichstromanteil übertragen [11].

Bei IEEE 802.3af dürfen die zu versorgenden Geräte, die bei PoE *Powered Device* (PD) heißen, bei einer Versorgungsspannung von 48 V und 350 mA Stromstärke eine Leistung von 12,95 Watt verbrauchen. Neuere PoE-Standards wie IEEE 802.3at (PoE+) und IEEE 802.3bt (4PPoE) bieten mehr PoE-Leistung, sind in der Praxis aber seltener vorzufinden.

Die Einspeisung der elektrischen Leistung auf ein Twisted-Pair-Kabel kann über ein *Power Sourcing Equipment* (PSE), das auch *Endspan* heißt, oder alternativ über ein *Midspan* erfolgen. Bei einem PSE bzw. Endspan handelt es sich um einen Switch, der PoE unterstützt. Ein Midspan ist ein sogenannter PoE-Injektor, der sich zwischen Switch und zu versorgendem Gerät (PE) befindet.

5.2.3

Fiber-optic Cables

An optical fiber consists (from inside to outside) of a light-transmitting *core*, which is made of quartz glass. The core is surrounded by a *cladding*, which has a refractive index lower than that of the core and causes the beams to be guided by total reflection at the boundary layer with the core. The cladding is enclosed by a

Lichtwellenleiter

Ein Lichtwellenleiter besteht (von innen nach außen) aus einem lichtübertragenden Kern (*Core*) aus Quarzglas. Um den Kern befindet sich ein Mantel (*Cladding*), der einen niedrigeren Brechungsindex als der Kern hat und durch Totalreflexion an der Grenzschicht zum Kern die Führung der Strahlen bewirkt. Der Mantel ist von

protective *coating* that is also called *buffer* (see Figure 5.17). The final layer is the outer protective cover (*jacket*).

einer Schutzbeschichtung (*Coating* oder *Buffer*) umschlossen (siehe Abbildung 5.17). Die letzte Schicht ist die äußere Schutzhülle (*Jacket*).

Figure 5.17: Structure of a Fiber-Optic Cable

The structure, dimensions and refractive index of the core and cladding determine the number of *propagation modes*, by which light can propagate along the optical fiber. Each mode corresponds to a path in the optical fiber. *Multi-mode fibers* have up to several thousand propagation modes. *Mono-mode (single-mode) fibers* have only a single propagation mode. For shorter distances of up to approximately 500 m, multi-mode fibers are used, and mono-mode fibers are used for longer distances of up to approximately 70 km.

Aufbau, Abmessungen und Brechungsindex von Kern und Mantel bestimmten die Anzahl der *Moden*, die sich in den Fasern des Lichtwellenleiters ausbreiten können. Jeder Mode entspricht einem Weg im Lichtwellenleiter. *Multimodefasern* besitzen bis zu mehreren tausend Moden und *Monomodefasern* nur einen Grundmode. Für kürzere Strecken bis ca. 500 m verwendet man Multimodefasern und Monomodefasern für längere Strecken bis ca. 70 km.

5.3

Structured Cabling

Structured cabling is a blueprint for a cross-building network infrastructure for the transmission of various voice and data services. As few transmission media as possible should allow the transmission of as many applications as possible. All present-day and future communication systems shall be supported, expensive faulty installations and extensions avoided and the installation of new network components facilitated.

Structured cabling is a part of a campus with several buildings, and it consists of a primary, secondary, and tertiary area.

The *primary area* is the cross-building cabling. This area consists of redundant cable routes with optical fibers. The primary area starts and ends at building distributors. Reasons for the use of

Strukturierte Verkabelung

Die *strukturierte Verkabelung*, die auch *universelle Gebäudeverkabelung* (UGV) heißt, ist ein Aufbauplan für eine gebäudeübergreifende Netzwerkinfrastruktur zur Übertragung verschiedener Sprach- und Datendienste. Möglichst wenige Übertragungsmedien sollen die Übertragung möglichst vieler Anwendungen erlauben. Alle heutigen und zukünftigen Kommunikationssysteme sollen unterstützt, teure Fehlinstallationen und Erweiterungen vermieden und die Installation neuer Netzwerkkomponenten erleichtert werden.

Eine strukturierte Verkabelung ist Teil eines gebäudeübergreifenden Campus und besteht aus Primär-, Sekundär- und Tertiärbereich.

Der *Primärbereich* ist die gebäudeübergreifende Verkabelung. Dieser Bereich heißt auch auch „Campusverkabelung“ oder „Geländeverkabelung“ und besteht aus redundanten Kabel-

optical fibers are the relatively large distances of several hundred meters, which must be bridged, the requirement that shielded cables can only be used if both sides of the cable have the same ground potential and therefore shielded cables cannot be used to connect different buildings and the required bandwidth.

The *secondary area* is inside a building. It is an internal backbone and connects the building distributor with the floor distributors. Here, the wiring of individual floors within a building takes place. For the cabling, copper cables or fiber optic cables are used.

The *tertiary area* is the star-topology shaped wiring that connects the junction boxes with the floor distributors. For cost reasons, copper cables (twisted pair cables) are usually used as transmission media. Short cables (1 to 10 m) are used to connect the junction boxes with the terminal devices.

trassen mit Lichtwellenleitern. Der Primärbereich beginnt und endet an Gebäudeverteilern. Gründe für den Einsatz von Lichtwellenleitern sind die relativ großen Entfernungen von mehreren hundert Metern, die es zu überbrücken gilt, die Erdungsproblematik und die benötigte Bandbreite.

Der *Sekundärbereich* befindet sich innerhalb eines Gebäudes. Er ist ein gebäudeinternes Backbone und verbindet den Gebäudeverteiler mit den Etagenverteilern. Hier findet die Verkabelung einzelner Stockwerke innerhalb eines Gebäudes statt. Die Verkabelung erfolgt via Kupferkabel oder Lichtwellenleiter.

Der *Tertiärbereich* ist die sternförmige Verkabelung auf Etagenebene. Er verbindet die Anschlussdosen mit Etagenverteilern. Als Übertragungsmedien werden aus Kostengründen meist Kupferkabel (Twisted-Pair-Kabel) verwendet. Als Verbindung zwischen den Anschlussdosen mit den Endgeräten dienen kurze Anschlusskabel (1 bis 10 m).

5.4

Devices of the Physical Layer

Since the problem of *attenuation* (signal weakening) exists for all transmission media, the maximum range of a network is limited. *Repeaters* extend the range of a LAN by cleaning received electrical or optical signals from noise and jitter (deviation of the transmission timing) and by amplifying them.

A Repeater interprets the signal levels it receives, reencodes and retransmits the data. It only forwards signals but does not analyze their meaning or examines their correctness. A Repeater has only two interfaces (*ports*).

Repeaters with more than two interfaces are called *multiport Repeater* or *Hub* (see Figure 5.18). Repeaters and Hubs have neither physical nor logical network addresses because they only forward received signals. Thus they operate transparently and do not communicate

Geräte der Bitübertragungsschicht

Weil bei allen Übertragungsmedien das Problem der *Dämpfung* (Signalabschwächung) besteht, ist die maximale Reichweite begrenzt. *Repeater* vergrößern die Reichweite eines LAN, indem sie empfangene elektrische oder optische Signale vom Rauschen und von Jitter (Genauigkeitsschwankungen im Übertragungstakt) reinigen und verstärken.

Ein Repeater interpretiert die von ihm empfangenen Leitungspegel und sendet die Daten frisch kodiert weiter. Er leitet Signale zwar weiter, analysiert aber nicht deren Bedeutung und untersucht auch nicht deren Korrektheit. Ein Repeater verfügt lediglich über zwei Schnittstellen (*Ports*).

Repeater mit mehr als zwei Schnittstellen heißen *Multiport-Repeater* oder *Hub* (siehe Abbildung 5.18). Repeater und Hubs haben weder physische noch logische Netzadressen, da sie empfangene Signale nur weiterleiten. Sie arbeiten somit transparent und kommunizieren nicht

at a higher protocol layer than the Physical Layer.

auf einer höheren Protokollschicht als der Bitübertragungsschicht.

Figure 5.18: Multiport Repeater (Hub) with eight Network Interfaces (Ports)

Hubs implement the physical star network topology and logical bus network topology. Equal to a long cable, that connects all network devices, a Hub forwards incoming signals to all other ports. Therefore, each terminal device, which is connected to a Hub, can receive and analyze the entire traffic that passes through the Hub. Hubs, too, cannot analyze the signals they transmit, but can only clean and amplify them.

One benefit of Hubs over the physical bus network topology is that the failure of individual cable segments does not cause the whole network to fail. The network is not interrupted even when network devices are added or removed.

Hubs can be cascaded to achieve a greater network expansion. However, this is not possible arbitrarily often, because the *Round-Trip-Time* (RTT) must not be exceeded. This is the period which a frame requires to be transmitted from the sender to the receiver and back (as confirmation for a successful reception). In this case, the maximum RTT is crucial, which is the time to go from one end to the most distant point of the network and back again. The RTT depends on the speed (data rate) of the network. If the network becomes too large, the RTT becomes too high. This causes collisions to occur

Hubs realisieren die physische Stern-Topologie und logische Bus-Topologie. Genau wie bei einem langen Kabel, an dem alle Netzwerkgeräte hängen, leitet ein Hub ankommende Signale zu allen anderen Schnittstellen weiter. Darum kann jedes Endgerät, das mit einem Hub verbunden ist, den vollständigen Datenverkehr, der über den Hub geht, empfangen und analysieren. Auch Hubs können die Signale, die sie weiterleiten, nicht analysieren, sondern nur reinigen und verstärken.

Ein Vorteil von Hubs gegenüber der physischen Bus-Topologie ist, dass beim Hub der Ausfall einzelner Kabelsegmente nicht das vollständige Netz lahmlegt. Auch beim Hinzufügen oder Entfernen von Netzwerkgeräten wird das Netz nicht unterbrochen.

Um eine größere Netzausdehnung zu erreichen, kann man Hubs kaskadieren. Das ist aber nicht beliebig viel möglich, denn die *Round-Trip-Time* (RTT) darf nicht überschritten werden. Das ist die Rundlaufzeit, die ein Netzwerkrahmen benötigt, um vom Sender zum Empfänger und (als Bestätigung für den erfolgreichen Empfang) zurück übertragen zu werden. In diesem Fall ist die maximale RTT entscheidend, also die Zeit, um vom einen Ende zum weitest entfernten Ende des Netzes zu gelangen und wieder zurück. Die RTT hängt von der Geschwindigkeit (Datenübertragungsrate) des Netzwerks ab. Wird das Netz zu groß, wird die RTT zu hoch.

more frequently. Also, undetected collisions are possible.

To avoid interference, the *5-4-3 Repeater Rule* exists. It specifies that no more than five cable segments may be connected. For this purpose, a maximum of four Repeaters are used, and terminal devices may only be connected to three segments. This rule only applies to Repeaters and Hubs. Gigabit Ethernet and even faster network standards no longer specify Repeaters and Hubs.

Just like a physical bus topology, all network devices connected to a Hub are inside a *Collision Domain*.

Also, *Modems* operate on the Physical Layer. These devices make it possible to transport signals over long distances by modulating them onto a carrier frequency in the high-frequency range. A Modem at the receiver side can recover the original signals by demodulating. Examples include (A)DSL or Cable Modems for broadband Internet access and telephone or fax modems for narrowband Internet access.

Dadurch werden Kollisionen häufiger und unerkannte Kollisionen möglich.

Um Störungen zu vermeiden, existiert die *5-4-3-Repeater-Regel*. Diese besagt, das nicht mehr als fünf Kabelsegmente verbunden sein dürfen. Dafür werden maximal vier Repeater eingesetzt und an nur drei Segmenten dürfen Endgeräte angeschlossen sein. Diese Regel gilt nur für Repeater und Hubs. Bei Gigabit-Ethernet und noch leistungsfähigeren Netzwerkstandards sind keine Repeater und Hubs mehr spezifiziert.

Genau wie bei einer physischen Bus-Topologie befinden sich alle Netzwerkgeräte, die an einen Hub angeschlossen sind, in einer *Kollisionsdomäne*.

Auch *Modems* arbeiten auf der Bitübertragungsschicht. Diese Geräte ermöglichen den Transport von Signalen über weite Strecken, indem sie diese auf eine Trägerfrequenz im Hochfrequenzbereich aufmodulieren. Ein Modem auf der Gegenseite kann die ursprünglichen Signale durch demodulieren wieder zurückgewinnen. Beispiele sind (A)DSL- oder Kabelmodems für Breitbandinternetzugänge und Telefon- oder Faxmodems für Schmalbandinternetzugänge.

5.4.1

Impact of Repeaters and Hubs on the Collision Domain

Extending a network with the media access method CSMA/CD (see section 6.8.1) by Repeaters or Hubs increases the collision domain, because these devices cannot analyze signals, but can only forward them (see Figure 5.19).

To make CSMA/CD work, collisions inside a collision domain must reach any network device within a particular time. If the collision domain is too large, there is a risk that the sending network devices do not detect collisions. Therefore, a maximum number of 1023 devices per collision domain is allowed. As the number of network devices increases, the number of collisions increases. Thin Ethernet (10BASE2) and Thick Ethernet (10BASE5) allow for a maximum of two pairs of Repeaters between any two network devices.

Auswirkungen von Repeatern und Hubs auf die Kollisionsdomäne

Erweitert man ein Netzwerk mit dem Medienzugriffsverfahren CSMA/CD (siehe Abschn 6.8.1) durch Repeater oder Hubs, vergrößert sich die Kollisionsdomäne, denn diese Geräte können keine Signale analysieren, sondern nur weiterleiten (siehe Abbildung 5.19).

Damit CSMA/CD funktioniert, müssen Kollisionen in einer Kollisionsdomäne innerhalb einer bestimmten Zeit jedes Netzwerkgerät erreichen. Ist die Kollisionsdomäne zu groß, besteht die Gefahr, dass sendende Netzwerkgeräte Kollisionen nicht erkennen. Darum darf es maximal 1023 Geräte pro Kollisionsdomäne geben. Mit der Anzahl der Netzwerkgeräte steigt die Anzahl der Kollisionen. Bei Thin Ethernet (10BASE2) und Thick Ethernet (10BASE5) sind maximal zwei Repeater-Paare zwischen zwei beliebigen Netzwerkgeräten erlaubt.

Figure 5.19: Collision Domains of Repeaters and Hubs

5.5

Encoding Data with Line Codes

The efficient encoding of data has not only been important since the advent of computer networks. One example (see Table 5.9) for an efficient way to encode data is the *Morse Code* (Morse Alphabet).

Kodierung von Daten in Netzwerken

Die effiziente Kodierung von Daten ist nicht erst seit dem Aufkommen von Computernetzen wichtig. Ein Beispiel (siehe Tabelle 5.9) für eine effiziente Kodierung ist der *Morsekode* (Morsealphabet).

Table 5.9: Morse Code of Samuel Morse from 1838

A	· —	J	· — — —	S	· · ·	1	· — — — —
B	— · · ·	K	— · —	T	—	2	· · — — —
C	— · — ·	L	· — · ·	U	· · —	3	· · · — —
D	— · ·	M	— —	V	· · · —	4	· · · · —
E	·	N	— ·	W	· — —	5	· · · · ·
F	· · — ·	O	— — —	X	— · · —	6	— · · · ·
G	— — ·	P	· — — ·	Y	— · — —	7	— — · · ·
H	· · · ·	Q	— — · —	Z	— — · ·	8	— — — · ·
I	· ·	R	· — ·	0	— — — — —	9	— — — — ·

The encoding is called *line code* in this context and specifies how signals are transmitted on the used transmission medium of a computer network. Specific signal level sequences are assigned to bit sequences in the data stream. Computer networks must implement these operations:

1. Conversion (*encoding*) of binary data to signals.

Die Kodierung, die in diesem Kontext *Leitungscode* heißt, legt in Computernetzen fest, wie Signale auf dem verwendeten Medium übertragen werden. Bestimmte Signalpegelfolgen werden Bitfolgen im Datenstrom zugeordnet. In Computernetzen sind folgende Aktionen nötig:

1. Umwandlung (*Kodierung*) von Binärdaten in Signale.

2. Transmission of the signals via the transmission to the receiver.

3. Reconversion (*decoding*) of the signals to binary data.

The encoding of binary data to signals is possible in various ways. The following pages present different line codes (see Table 5.10).

The most straightforward way of representing the logical 0 and the logical 1 is by using different voltage levels. For example, a zero-bit can be encoded by one signal level (e.g., 0 V) and a one-bit by another signal level (e.g., 5 V). This line code is NRZ.

2. Übertragung der Signale über das Medium vom Sender zum Empfänger.

3. Rückwandlung (*Dekodierung*) der Signale in Binärdaten.

Die Kodierung der Binärdaten in Signale ist auf verschiedene Arten möglich. Auf den nachfolgenden Seiten werden verschiedene Leitungscodes vorgestellt (siehe Tabelle 5.10).

Die einfachste Form der Darstellung von logischer Null und Eins ist mit verschiedenen Spannungsniveaus möglich. Es kann zum Beispiel eine Null durch einen Signalpegel (zum Beispiel 0 Volt) und eine Eins durch einen anderen Signalpegel (zum Beispiel 5 Volt) kodiert werden. Dieser Leitungscode ist NRZ.

5.5.1

Non-Return to Zero (NRZ)

This line code encodes a logical zero-bit with signal level 1 and a logical one-bit with signal level 2 (see Figure 5.20).

The serial bus system CAN (Controller Area Network) that was developed by Bosch in the 1980s for connecting control units in cars implements NRZ.

When transmitting long series of zero-bits or one-bits, the physical signal level does not change. This causes *baseline wander* and problems with *clock recovery*.

Shift of the average Signal Level

The receiver differentiates the signal levels by using the average signal level of a certain number of recently received signals. Signals below the average are interpreted by the receiver as zero-bit and signals clearly above the average as one-bit. When transmitting long sequences of zero- or one-bits, the average may shift so much that it becomes difficult to detect a significant change of the signal.

To prevent a shift of the average (*Baseline Wander*) when using a line code with two signal levels, the usage of both signal levels must be *evenly distributed*. The data to be transmitted must, therefore, be encoded in such a way that

Non-Return to Zero (NRZ)

Dieser Leitungscode kodiert den Datenwert Null mit Signalpegel 1 und den Datenwert Eins mit Signalpegel 2 (siehe Abbildung 5.20).

Das serielle Bussystem CAN-Bus (Controller Area Network), das die Firma Bosch in den 1980er Jahren zur Vernetzung von Steuergeräten in Automobilen entwickelt hat, verwendet NRZ.

Beim Übertragen längerer Serien von Nullen oder Einsen gibt es keine Änderung des Signalpegels. Das führt zur *Verschiebung des Durchschnitts* und Problemen bei der *Taktrückgewinnung*.

Verschiebung des Durchschnitts

Der Empfänger unterscheidet die Signalpegel anhand des Durchschnitts einer bestimmten Anzahl zuletzt empfangener Signale. Signale unter dem Durchschnitt interpretiert der Empfänger als Null und deutlich über dem Durchschnitt interpretiert er als Eins. Beim Übertragen langer Serien von Nullen oder Einsen kann sich der Durchschnitt soweit verschieben, dass es schwierig wird, eine signifikante Änderung im Signal zu erkennen.

Zur Verhinderung der Durchschnittsverschiebung (*Baseline Wander*) muss im Leitungscode die Belegung der Signalpegel bei zwei Signalpegeln *gleichverteilt* sein. Die zu übertragenden Daten müssen also so kodiert werden, dass die Si-

Table 5.10: Line Codes

Line code	Signal levels	Baseline wander possible	Signal level change	Selfsynchronizing[a]	Efficiency[b]	Directly transferable	Additional encoding
NRZ	2	yes	at changes	no	100 %	no	—
NRZI	2	yes	for 1-bits	no	75 %	no	—
MLT-3	3	yes	for 1-bits	no	100 %	no	—
RZ	3	yes	always	yes	50 %	no	—
Unip. RZ	2	yes	for 1-bits	no	75 %	no	—
AMI	3	no	for 1-bits	no	100 %	no	Scrambler
B8ZS	3	no	for 1-bits	yes	100 %	yes	—
Manchester	2	no	always	yes	50 %	yes	—
Manchester II	2	no	always	yes	50 %	yes	—
Diff. Manch.	2	yes	always	yes	50 %	yes	—
4B5B	2	yes	—	yes	80 %	no	NRZI or MLT-3
5B6B	2	no	—	yes	$83.\overline{3}$ %	no	NRZ
8B10B	2	no	—	yes	80 %	no	NRZ
8B6T	3	no	—	yes	100 %	yes	—

[a] Specifies if the clock recovery is possible with this line code.
[b] Ratio of bit rate (payload in bits per time) and baud rate (signal changes per second).

Figure 5.20: NRZ

the signal levels occur equally frequently. If a network technology uses three or five signal levels, the average over the time must be equal to the average signal level.

Clock Recovery (Synchronization)

Another problem with NRZ and other line codes is the *clock recovery* (synchronization). Even if the processes for encoding and decoding run on different computers, they must be controlled by the same clock. In each clock cycle, the sender transmits one bit, and the receiver receives one bit. If the clocks of the sender and receiver drift apart, the receiver may become confused during a sequence of logic zero-bits or one-bits. One way to avoid this problem is to have a separate line that transmits the clock.

One network technology that has a separate line only for the clock is the serial bus system I²C (Inter-Integrated Circuit). However, equal to comparable bus systems, this is only suitable for local applications and not for connecting systems over large distances.

In computer networks, separate signal lines only for the clock are not feasible because of the cabling effort. Instead, it is recommended to increase the number of signal level changes to enable the clock recovery from the data stream.

gnalpegel gleich häufig vorkommen. Verwendet eine Netzwerktechnologie drei oder fünf Signalpegel, muss der Durchschnitt über die Zeit dem mittleren Signalpegel entsprechen.

Taktrückgewinnung (Synchronisierung)

Ein weiteres Problem bei NRZ und anderen Leitungscodes ist die Taktrückgewinnung (*Clock Recovery*) oder Synchronisierung. Auch wenn die Prozesse für Kodierung und Dekodierung auf unterschiedlichen Rechnern laufen, müssen sie vom gleichen Takt gesteuert werden. In jedem Taktzyklus überträgt der Sender ein Bit und der Empfänger empfängt eins. Driften die Taktgeber von Sender und Empfänger auseinander, könnte sich der Empfänger bei einer Folge von Nullen oder Einsen verzählen. Eine Möglichkeit das Problem zu vermeiden, ist eine getrennte Leitung, die den Takt überträgt.

Eine Vernetzungstechnologie, die eine separate Signalleitung nur für den Takt vorsieht, ist das serielle Bussystem I²C (Inter-Integrated Circuit). Dieses ist aber wie auch vergleichbare Bussysteme nur für lokale Anwendungen und nicht zur Überbrückung großer Distanzen geeignet.

In Computernetzen sind separate Signalleitungen nur für den Takt wegen des Verkabelungsaufwands nicht praktikabel. Stattdessen empfiehlt es sich, die Anzahl der Signalpegelwechsel zu erhöhen, um die Taktrückgewinnung aus dem Datenstrom zu ermöglichen.

5.5.2

Non-Return to Zero Invert (NRZI)

NRZI is a variant of NRZ. To send a one-bit, a signal level change occurs at the beginning of the clock. To send a one-bit, the signal level remains unchanged for a whole clock (see Figure 5.21).

Non-Return to Zero Invert (NRZI)

NRZI ist eine Variante von NRZ. Um den Datenwert Eins zu senden, findet zu Beginn des Takts ein Wechsel des Signalpegels statt. Um den Datenwert Null zu senden, bleibt der Signalpegel einen ganzen Takt lang unverändert (siehe Abbildung 5.21).

Figure 5.21: NRZI

Clock recovery is impossible for sequences of zero-bits. Because the usage of the signal levels is not evenly distributed, baseline wander can occur.

This line code is used, among others, by FDDI and Ethernet 100BASE-FX.

Die Taktrückgewinnung ist bei Reihen aufeinanderfolgender Nullen nicht möglich. Weil die Belegung der Signalpegel nicht gleichverteilt ist, kann es zu Durchschnittsverschiebungen kommen.

Diesen Leitungscode verwenden unter anderem FDDI und Ethernet 100BASE-FX.

5.5.3

Multilevel Transmission Encoding – 3 Levels (MLT-3)

This line code uses three signal levels (+, 0 and −). If a zero-bit is transmitted, no signal level change takes place. A one-bit value is encoded alternately according to the sequence [+, 0, −, 0] (see Figure 5.22).

Multilevel Transmission Encoding – 3 Levels (MLT-3)

Dieser Leitungscode verwendet drei Signalpegel (+, 0 und −). Beim Senden des Datenwerts Null findet kein Pegelwechsel statt. Der Datenwert Eins wird abwechselnd entsprechend der Folge [+, 0, −, 0] kodiert (siehe Abbildung 5.22).

Figure 5.22: MLT-3

As with NRZI, the problem of clock recovery exists with consecutive zero-bits and baseline wander can occur.

This line code is used, among others, by Ethernet 100BASE-TX.

Genau wie bei NRZI besteht das Problem der Taktrückgewinnung bei Reihen aufeinanderfolgender Nullen und Durchschnittsverschiebungen sind möglich.

Diesen Leitungscode verwendet unter anderem Ethernet 100BASE-TX.

5.5.4

Return-to-Zero (RZ)

RZ also uses three signal levels. To send a one-bit, the positive signal level is transmitted for half a clock cycle, and then the signal level returns to the average signal level. To send a zero-bit, the negative signal level is transmit-

Return-to-Zero (RZ)

Auch RZ verwendet drei Signalpegel. Um den Datenwert Eins zu senden, wird der positive Signalpegel für einen halben Takt übertragen und danach zum mittleren Signalpegel zurückgekehrt. Um den Datenwert Null zu senden, wird

ted for half a clock cycle, and then the signal level returns to the average signal level (see Figure 5.23).

The guaranteed change of the signal level for each transmitted bit enables the receiver to do the clock recovery. However, the required bandwidth compared to NRZ is twice as high. Also, successive one-bits or zero-bits may result in baseline wander.

der negative Signalpegel für einen halben Takt übertragen und danach zum mittleren Signalpegel zurückgekehrt (siehe Abbildung 5.23).

Die garantierte Änderung des Signalpegels bei jedem übertragenen Datenbit ermöglicht dem Empfänger die Taktrückgewinnung. Die benötigte Bandbreite gegenüber NRZ ist aber doppelt so hoch. Zudem kann es bei Reihen aufeinanderfolgender Einsen oder Nullen zu Durchschnittsverschiebungen kommen.

Figure 5.23: Return-to-Zero

5.5.5

Unipolar RZ

This line code, which is among others used for wireless optical data transmission via IrDA in the SIR transmission mode, is a particular form of the RZ line code because it uses only two signal levels. For sending a one-bit, the signal level returns to the low signal level after half the clock cycle. Sending a zero-bit does not cause a signal level change (see Figure 5.24).

Unipolares RZ

Dieser Leitungscode, der unter anderem bei drahtlosen optischen Datenübertragungen via IrDA im Übertragungsmodus SIR verwendet wird, ist eine Sonderform der RZ-Kodierung, denn er verwendet nur zwei Signalpegel. Beim Senden des Datenwerts Eins kehrt man nach dem halben Takt zum Signalpegel 1 zurück. Beim Senden des Datenwerts Null findet kein Pegelwechsel statt (siehe Abbildung 5.24).

Figure 5.24: Unipolar Return-to-Zero

With sequences of zero-bits, the signal level does not change. This makes clock recovery impossible for the receiver. Even with this line code, the usage of the signal levels is not equally distributed. Thus, baseline wander can occur.

Bei Serien von Nullen findet kein Wechsel des Signalpegels statt. Das macht die Taktrückgewinnung für den Empfänger unmöglich. Auch bei diesem Leitungscode ist die Belegung der Signalpegel nicht gleichverteilt. Somit sind Durchschnittsverschiebungen möglich.

5.5.6

Alternate Mark Inversion (AMI)

AMI, which is also called *bipolar encoding*, uses three signal levels. A zero-bit is transmitted as average signal level (0). One-bits are transmitted alternately as a positive (+) or negative signal level (-). Therefore, baseline wander cannot occur (see Figure 5.25).

Alternate Mark Inversion (AMI)

AMI, der auch *Bipolar Encoding* heißt, arbeitet mit drei Signalpegeln. Der Datenwert Null wird als mittlerer Signalpegel (0) übertragen. Der Datenwert Eins wird abwechselnd als positiver (+) oder negativer Signalpegel (-) übertragen. Durchschnittsverschiebungen sind daher ausgeschlossen (siehe Abbildung 5.25).

Figure 5.25: Alternate Mark Inversion

The ISDN S_0 bus uses a modified version of this line code. A one-bit is transmitted as average signal level, and zero-bits are transmitted alternately as a positive or negative signal level.

The problem of consecutive one-bits is solved with AMI. However, the problem of consecutive zero-bits still exists because they do not cause a signal level change. This makes clock recovery impossible for the receiver. For this reason, a *scrambler* is usually used after the AMI encoding. Scramblers convert a bitstream according to a simple algorithm in a reversible way. In this case, scramblers are used to disrupt long sequences of zero-bits to enable clock recovery for the receiver.

The detection of transmission errors is partially possible with AMI, since the signal sequences ++, --, +0+ and -0- are not allowed.

Der ISDN S_0-Bus verwendet eine modifizierte Version der AMI-Kodierung. Dabei wird der Datenwert Eins als mittlerer Signalpegel und der Datenwert Null abwechselnd als positiver oder negativer Signalpegel übertragen.

Das Problem aufeinanderfolgender Einsen ist bei AMI gelöst. Das Problem aufeinanderfolgender Nullen besteht aber immer noch, weil es hier nicht zum Wechsel des Signalpegels kommt. Das macht die Taktrückgewinnung für den Empfänger unmöglich. Aus diesem Grund wird nach der AMI-Kodierung meistens ein *Scrambler* (Verwürfler) verwendet. Scrambler stellen einen Datenstrom nach einem einfachen Algorithmus umkehrbar um. In diesem Fall kommen Scrambler zum Einsatz, die lange Ketten von Nullen unterbrechen, um so die Taktrückgewinnung für den Empfänger zu ermöglichen.

Die Erkennung von Übertragungsfehlern ist bei AMI teilweise möglich, da die Signalfolgen ++, --, +0+ und -0- nicht erlaubt sind.

5.5.7

Bipolar with 8 Zeros Substitution (B8ZS)

This line code is a slightly modified version of AMI. B8ZS prevents a loss of synchronization for series of zero-bits by two rules for modifying sequences of eight zero-bits:

Bipolar with 8 Zeros Substitution (B8ZS)

Dieser Leitungscode ist eine geringfügig modifizierte Version von AMI. B8ZS verhindert den Synchronisationsverlust bei Serien von Nullen

- `+00000000` is encoded as `+000+-0-+`.
- `-00000000` is encoded as `-000-+0+-`.

Both replacement rules are code violations, because, in both substitution rules, two positive and two negative signal levels occur one after another. This makes the substitutions recognizable for the receiver.

With B8ZS, in contrast to AMI, no scramblers are necessary because long sequences of zero-bits are no problem. As with AMI, baseline wander cannot occur.

durch zwei Regeln zur Modifikation von Folgen von 8 Nullbits:

- `+00000000` wird `+000+-0-+` kodiert.
- `-00000000` wird `-000-+0+-` kodiert.

Eigentlich sind beide Ersetzungsregeln Coderegelverletzungen, denn in beiden Ersetzungen kommen zwei positive und zwei negative Signalpegel nacheinander vor. Das macht die Ersetzungen für den Empfänger erkennbar.

Bei B8ZS sind im Gegensatz zu AMI keine Scrambler nötig, weil längere Serien von Nullen kein Problem sind. Durchschnittsverschiebungen sind wie bei AMI ausgeschlossen.

Figure 5.26: Manchester

5.5.8

Manchester

The Manchester line code uses two signal levels and is self-synchronizing because of the signal level changes in each bit cell. A logical one-bit is encoded with a change from signal level 1 to signal level 2 (rising edge), and a zero-bit is encoded with a change from signal level 2 to signal level 1 (falling edge). If two identical bits follow each other, at the end of the bit cell, the signal level changes to the initial level (see Figure 5.26).

Ethernet uses this line code with 10 Mbps (for example 10BASE2 and 10BASE-T).

The beginning of a transmission (which means the first bit cell) is highlighted by a particular bit sequence (*preamble*). Because the signal level always changes with this line code, clock recovery is possible for the receiver. Another benefit is that the baseline wander cannot occur because the usage of the signal levels is equally distributed.

A drawback of the Manchester encoding is that the transmission of a single bit requires an

Manchester

Die Manchesterkodierung arbeitet mit zwei Signalpegeln und ist selbstsynchronisierend, weil in jeder Bitzelle der Signalpegel wechselt. Der Datenwert Eins wird mit einem Wechsel von Signalpegel 1 zu Signalpegel 2 (steigende Flanke) kodiert und der Datenwert Null mit einem Wechsel von Signalpegel 2 zu Signalpegel 1 (fallende Flanke). Folgen zwei identische Bits aufeinander, wird am Ende der Bitzelle auf das Anfangsniveau zurückgesprungen (siehe Abbildung 5.26).

Diesen Leitungscode verwendet Ethernet mit 10 Mbit/s (zum Beispiel 10BASE2 und 10BASE-T).

Der Beginn einer Übertragung (also die erste Bitzelle) wird durch eine spezielle Bitfolge (*Präambel*) gekennzeichnet. Weil es bei diesem Leitungscode stets Wechsel des Signalpegels gibt, ist die Taktrückgewinnung für den Empfänger möglich. Ein weitere Vorteil ist, das der Durchschnitt sich nicht verschieben kann, weil die Belegung der Signalpegel gleichverteilt ist.

Nachteilig bei der Manchesterkodierung ist, dass die Übertragung eines Bits im Schnitt 1,5

average of 1.5 signal level changes. Since the number of signal level changes is a limiting factor of the transmission medium, modern network technologies don't use this line code.

The Manchester encoding, the bit rate, which is the number of transferred payload data (in bits) per time, corresponds to half the baud rate (the rate at which signals change per time unit). So the efficiency of the line code is only 50 % compared to NRZ.

Wechsel des Signalpegels erfordert. Da die Anzahl der Pegelwechsel ein limitierender Faktor des Übertragungsmediums ist, verwenden moderne Netzwerktechnologien andere Leitungscodes als die Manchesterkodierung.

Bei der Manchesterkodierung entspricht die Bitrate, also die Anzahl der Nutzdaten (in Bits) pro Zeit der halben Baudrate (Rate, in der sich Signale pro Zeiteinheit ändern). Also ist die Effizienz des Leitungscodes nur 50 % im Vergleich zu NRZ.

5.5.9

Manchester II

This line code is the opposite of the Manchester encoding. In Manchester II, a one-bit is encoded with a falling edge, and a zero-bit is encoded with a rising edge (see Figure 5.27).

Manchester II

Dieser Leitungscode ist das Gegenteil der Manchesterkodierung. Bei Manchester II wird der Datenwert Eins mit einer fallenden Flanke kodiert und der Datenwert Null mit einer steigenden Flanke (siehe Abbildung 5.27).

Figure 5.27: Manchester II

As with Manchester, clock recovery is possible for the receiver, and the usage of the signal levels is equally distributed.

Wie bei Manchester ist die Taktrückgewinnung für den Empfänger möglich und die Belegung der Signalpegel gleichverteilt.

5.5.10

Differential Manchester Encoding

This variant of the Manchester encoding is also called *Conditional DePhase Encoding* (CDP). Here, too, a signal level change inside each bit cell occurs for clock recovery.

If the next data to be encoded is a one-bit, the signal level does not change at the beginning of the bit cell, but only in the middle. If the next data to be encoded is a zero-bit, a signal level change also takes place at the beginning of the bit cell. Depending on the initial level, two signal sequences are possible, which are inverse to each other (see Figure 5.28).

This line code is used by Token Ring (IEEE 802.5). Also with this variant of Manchester coding, clock recovery is possible for the receiver,

Differentielle Manchesterkodierung

Diese Variante der Manchesterkodierung heißt auch *Conditional DePhase Encoding* (CDP). Auch hier findet innerhalb jeder Bitzelle ein Pegelwechsel zur Taktrückgewinnung statt.

Ist der nächste zu kodierende Datenwert Eins, findet am Anfang der Bitzelle kein Wechsel des Signalpegels statt, sondern erst in der Mitte. Ist der nächste zu kodierende Datenwert Null, findet auch am Anfang der Bitzelle ein Pegelwechsel statt. Abhängig vom Anfangspegel ergeben sich zwei mögliche, zueinander inverse Signalfolgen (siehe Abbildung 5.28).

Diesen Leitungscode verwendet Token Ring (IEEE 802.5). Auch bei dieser Variante der Manchesterkodierung ist die Taktrückgewinnung für

Figure 5.28: Differential Manchester Encoding

and the usage of the signal levels is equally distributed.

den Empfänger möglich und die Belegung der Signalpegel gleichverteilt.

5.6

Line Codes which improve the Payload

None of the line codes presented so far is free of drawbacks. Baseline wander is a problem with NRZ for series of zero-bits and one-bits. For NRZI, MLT-3, unipolar RZ and AMI, sequences of zero-bits are a problem. Clock recovery is not guaranteed with NRZ, NRZI, MLT-3, unipolar RZ and AMI.

With the variants of the Manchester encoding and with RZ, there is a signal level change for each bit, and with the AMI variant, B8ZS frequent signal level changes are guaranteed. This means that clock recovery is not a problem with these line codes. Even baseline wander cannot occur, because the usage of the signal levels is equally distributed. However, the efficiency is poor, especially for the Manchester coding variants.

Modern network technologies encode the payload bit stream first with a line code that works efficiently, but also ensures clock recovery and avoids baseline wander. These line codes *improve* the payload bit stream in a way that a further encoding with the line codes NRZ, NRZI and MLT-3 is possible without problems. The aim is therefore to achieve the positive characteristics of the Manchester line code and the best possible overall efficiency.

Examples of line codes, which improve the payload bit stream first, include 4B5B, 5B6B and 8B10B. These line codes encode input

Nutzdaten mit Leitungscodes verbessern

Keiner der bis bislang vorgestellten Leitungscodes ist frei von Nachteilen. Die Verschiebung des Durchschnitts ist bei NRZ ein Problem bei Serien von Nullen und Einsen. Bei NRZI, MLT-3, Unipolarem RZ und AMI besteht nur das Problem aufeinanderfolgender Nullen. Die Taktrückgewinnung ist bei NRZ, NRZI, MLT-3, Unipolarem RZ und AMI nicht garantiert.

Bei den Varianten der Manchesterkodierung und bei RZ gibt es bei jedem Bit eine Änderung des Signalpegels und bei der AMI-Variante B8ZS sind häufige Änderungen des Signalpegels garantiert. Damit ist die Taktrückgewinnung kein Problem bei diesen Leitungscodes. Auch Durchschnittsverschiebungen können nicht auftreten, weil die Belegung der Signalpegel gleichverteilt ist. Allerdings ist besonders bei den Varianten der Manchesterkodierung die Effizienz schlecht.

Moderne Netzwerktechnologien kodieren die Nutzdaten zuerst mit einer Kodierung, die einerseits Effizienz verspricht, aber auch die Taktrückgewinnung garantiert und die Verschiebung des Durchschnitts vermeidet. Diese Kodierungen *verbessern* die Nutzdaten dahingehend, dass eine weitere Kodierung mit den Leitungscodes NRZ, NRZI oder MLT-3 ohne Probleme möglich ist. Das Ziel ist also, die positiven Eigenschaften der Manchesterkodierung und eine möglichst hohe Effizienz zu erreichen.

Beispiele für Leitungscodes, die die Nutzdaten zuerst aufbereiten, sind 4B5B, 5B6B und 8B10B. Diese Leitungscodes kodieren Eingabeblöcke fester Größe in Ausgabeblöcke fester Größe.

blocks of fixed size into output blocks of fixed size.

5.6.1

4B5B Encoding

This line code maps groups of four payload bits onto groups of five code bits. With five bits, 32 encodings are possible. Of these, only 16 encodings are used for data (`0`–`9` and `A`–`F`). Some of the remaining 16 encodings are used for connection control purposes. Because of the additional bit, the output bit rate increases by a factor of 5/4 compared to the payload bit rate (see Table 5.11). Hence, the efficiency of the 4B5B encoding is 80 %.

Each 5-bit encoding has a maximum of one leading zero-bit and has a maximum of three consecutive zero-bits in the output data stream. Therefore, clock recovery for the receiver is possible.

After encoding with 4B5B, another encoding takes place. By a combination of 4B5B, for example, with NRZI (for two signal levels) or MLT-3 (for three signal levels), baseline wander cannot occur.

The 5-bit combinations, which are missing in Table 5.11, are invalid because they have more than one leading zero-bits two consecutive zero-bits. 4B5B is used by Fast Ethernet 100BASE-TX and 100BASE-FX as well as by fiber-optic connections according to the FDDI standard. For Ethernet 100BASE-TX, after the encoding with 4B5B, another coding with MLT-3 takes place. For FDDI and Ethernet 100BASE-FX, after encoding with 4B5B, a further encoding with NRZI takes place.

4B5B-Kodierung

Bei dieser Kodierung werden 4 Nutzdatenbits auf 5 Codebits abgebildet. Mit 5 Bits sind 32 Kodierungen möglich. Davon werden nur 16 Kodierungen für Daten verwendet (`0`–`9` und `A`–`F`). Die übrigen 16 Kodierungen werden teilweise für Steuerzwecke verwendet. Wegen des zusätzlichen Bits steigt die kodierte Bitrate um den Faktor 5/4 gegenüber der Nutzdatenbitrate (siehe Tabelle 5.11). Die Effizienz der 4B5B-Kodierung ist also 80 %.

Jede 5-Bit-Kodierung hat maximal eine führende Null und im Ausgabedatenstrom gibt es maximal drei Nullen in Folge. Die Taktrückgewinnung für den Empfänger ist somit möglich.

Nach der Kodierung mit 4B5B erfolgt eine weitere Kodierung. Durch eine Kombination von 4B5B, zum Beispiel mit NRZI (für zwei Signalpegel) oder MLT-3 (für drei Signalpegel) kann keine Durchschnittsverschiebung auftreten.

Die in Tabelle 5.11 fehlenden 5-Bit-Kombinationen sind ungültig, da sie mehr als eine führende oder zwei aufeinanderfolgende Nullen besitzen. 4B5B wird bei Fast-Ethernet 100BASE-TX und 100BASE-FX sowie bei Glasfaserverbindungen nach dem FDDI-Standard verwendet. Bei Ethernet 100BASE-TX erfolgt nach der Kodierung mit 4B5B eine weitere Kodierung mit MLT-3. Bei FDDI und Ethernet 100BASE-FX erfolgt nach der Kodierung mit 4B5B eine weitere Kodierung mit NRZI.

5.6.2

5B6B Encoding

With 5B6B, five payload bits are mapped onto six code bits (see Table 5.12). Of the 32 possible 5-bit words, 20 are mapped to 6-bit words that contain the same number of one-bits and zero-bits (*neutral inequality*). For the remaining twelve 5-bit words exists a variant with two one-bits and four zero-bits (*positive inequality*) and a variant with four one-bits and two zero-bits (*negative inequality*).

5B6B-Kodierung

Bei 5B6B werden fünf Nutzdatenbits auf sechs Codebits abgebildet (siehe Tabelle 5.12). Von den 32 möglichen 5-Bit-Wörtern werden 20 auf 6-Bit-Wörter mit einer identischen Anzahl Einsen und Nullen (*neutrale Ungleichheit*) abgebildet. Für die verbleibenden zwölf 5-Bit-Wörter existiert je eine Variante mit zwei Einsen und vier Nullen (*positive Ungleichheit*) und eine mit vier Einsen und zwei Nullen (*negative Ungleichheit*).

Table 5.11: 4B5B

Label	4B	5B	Function
0	0000	11110	0 hexadecimal (payload)
1	0001	01001	1 hexadecimal (payload)
2	0010	10100	2 hexadecimal (payload)
3	0011	10101	3 hexadecimal (payload)
4	0100	01010	4 hexadecimal (payload)
5	0101	01011	5 hexadecimal (payload)
6	0110	01110	6 hexadecimal (payload)
7	0111	01111	7 hexadecimal (payload)
8	1000	10010	8 hexadecimal (payload)
9	1001	10011	9 hexadecimal (payload)
A	1010	10110	A hexadecimal (payload)
B	1011	10111	B hexadecimal (payload)
C	1100	11010	C hexadecimal (payload)
D	1101	11011	D hexadecimal (payload)
E	1110	11100	E hexadecimal (payload)
F	1111	11101	F hexadecimal (payload)
Q	—	00000	Quiet (the line is gone dead) $\Longrightarrow$ signal loss
I	—	11111	Idle (the line is idle) $\Longrightarrow$ pause
J	—	11000	Start (part 1)
K	—	10001	Start (part 2)
T	—	01101	Stop (part 1)
R	—	00111	Stop (part 2) $\Longrightarrow$ reset
S	—	11001	Set
H	—	00100	Halt (transmission failure)

As soon as the first 5-bit word without neutral inequality is to be processed, the variant with the positive inequality is used. For encoding the next 5-bit word without neutral inequality, the variant with the negative inequality is used. The variants with positive or negative inequality thus alternate.

After the encoding with 5B6B, another encoding with NRZ takes place. This is feasible because with 5B6B clock recovery is possible for the receiver and baseline wander cannot occur. Fast Ethernet 100Base-VG uses 5B6B. The advantage over Manchester encoding is its higher baud rate.

Sobald das erste 5-Bit-Wort ohne Abbildung mit neutraler Ungleichheit verarbeitet werden soll, wird auf die Variante mit der positiven Ungleichheit zurückgegriffen. Beim nächsten 5-Bit-Wort ohne neutrale Ungleichheit wird die Variante mit negativer Ungleichheit verwendet. Die Varianten mit positiver oder negativer Ungleichheit wechseln sich also ab.

Nach der Kodierung mit 5B6B erfolgt eine weitere Kodierung mittels NRZ. Das ist möglich, da bei 5B6B die Taktrückgewinnung für den Empfänger möglich ist und keine Durchschnittsverschiebungen auftreten können. 5B6B wird bei Fast-Ethernet 100Base-VG verwendet. Der Vorteil gegenüber der Manchesterkodierung ist die höhere Baudrate.

Table 5.12: 5B6B

5B	6B (neutral)	6B (positive)	6B (negative)	5B	6B (neutral)	6B (positive)	6B (negative)
00000		001100	110011	10000		000101	111010
00001	101100			10001	100101		
00010		100010	101110	10010		001001	110110
00011	001101			10011	010110		
00100		001010	110101	10100	111000		
00101	010101			10101		011000	100111
00110	001110			10110	011001		
00111	001011			10111		100001	011110
01000	000111			11000	110001		
01001	100011			11001	101010		
01010	100110			11010		010100	101011
01011		000110	111001	11011	110100		
01100		101000	010111	11100	011100		
01101	011010			11101	010011		
01110		100100	011011	11110		010010	101101
01111	101001			11111	110010		

5.6.3

8B10B Encoding

With 8B10B, eight payload bits are encoded to ten code bits. The efficiency is thus 80 %. 8B10B is used – among others – by Gigabit-Ethernet 1000Base-CX, 1000Base-SX, 1000Base-LX, Fibre Channel, InfiniBand, FireWire 800 (IEEE 1394b) and USB 3.0. Each 8B10B encoding is constructed in a way, such in the 10 bits either...

- five zero-bits and five one-bits occur (neutral inequality)
- six zero-bits and four one-bits occur (positive inequality)
- four zero-bits and six one-bits occur (negative inequality)

After encoding with 8B10B, another encoding with NRZ takes place. Baseline wander cannot occur, because some of the $2^8 = 256$ possible 8-bit words can be encoded in two different ways. This way, inequalities are balanced out. Each 10-bit encoding contains at least three signal level changes and not later than after five clock cycles, the signal level changes. This enables the receiver to recover the clock.

8B10B-Kodierung

Bei 8B10B werden 8 Nutzdatenbits auf 10 Codebits kodiert. Die Effizienz ist somit 80 %. 8B10B wird unter anderem bei Gigabit-Ethernet 1000Base-CX, 1000Base-SX, 1000Base-LX, Fibre Channel, InfiniBand, FireWire 800 (IEEE 1394b) und USB 3.0 verwendet. Jede 8B10B-Kodierung ist derart aufgebaut, dass in den 10 Bits entweder...

- 5 ×Null und 5 ×Eins vorkommt (neutrale Ungleichheit)
- 6 ×Null und 4 ×Eins vorkommt (positive Ungleichheit)
- 4 ×Null und 6 ×Eins vorkommt (negative Ungleichheit)

Nach der Kodierung mit 8B10B erfolgt eine weitere Kodierung via NRZ. Durchschnittsverschiebungen sind nicht möglich, weil einige der $2^8 = 256$ möglichen 8-Bit-Wörter auf zwei verschiedene Arten kodiert werden können. So werden Ungleichheiten ausgeglichen. Jede 10-Bit-Kodierung enthält mindestens drei Pegelsprünge und nach spätestens fünf Takten wechselt der

Signalpegel. Das ermöglicht dem Empfänger die Taktrückgewinnung.

5.7

Further Line Codes

In addition to the line codes discussed so far, there are other, more recent line codes, of which 8B6T is described here as a representative.

Weitere Leitungscodes

Neben den bislang behandelten Leitungscodes existieren weitere, aktuellere Leitungscodes, von denen an dieser Stelle stellvertretend 8B6T beschrieben wird.

Table 5.13: 8B6T

8B	6T	8B	6T	8B	6T	8B	6T
00	+-00+-	10	+0+--0	20	00-++-	30	+-00-+
01	0+-+-0	11	++0-0-	21	--+00+	31	0+--+0
02	+-0+-0	12	+0+-0-	22	++-0+-	32	+-0-+0
03	-0++-0	13	0++-0-	23	++-0-+	33	-0+-+0
04	-0+0+-	14	0++--0	24	00+0-+	34	-0+0-+
05	0+--0+	15	++00--	25	00+0+-	35	0+-+0-
06	+-0-0+	16	+0+0--	26	00-00+	36	+-0+0-
07	-0+-0+	17	0++0--	27	--+++-	37	-0++0-
08	-+00+-	18	0+-0+-	28	-0-++0	38	-0++0-
09	0-++-0	19	0+-0-+	29	--0+0+	39	0-+-+0
0A	-+0+-0	1A	0+-++-	2A	-0-+0+	3A	-+0-+0
0B	+0-+-0	1B	0+-00+	2B	0--+0+	3B	+0--+0
0C	+0-0+-	1C	0-+00+	2C	0--++0	3C	+0-0-+
0D	0-+-0+	1D	0-+++-	2D	--00++	3D	0-++0-
0E	-+0-0+	1E	0-+0-+	2E	-0-0++	3E	-+0+0-
0F	+0--0+	1F	0-+0+-	2F	0--0++	3F	+0-+0-

5.7.1

8B6T Encoding

The name 8B6T stands for *8 Binary 6 Ternary*. This line code encodes 8-bit blocks into six symbols, each of which can represent the state `-`, `0` or `+`. The symbols of the states represent electrical signal levels. The encoding is performed using a table (see Table 5.13) that contains all $2^8 = 256$ possible 8-bit combinations. For reasons of space, Table 5.13 only contains the first 48 of 256 possible 8-bit combinations. Fast Ethernet 100BASE-T4 uses this line code.

8B6T-Kodierung

Der Name 8B6T steht für *8 Binary 6 Ternary*. Dieser Leitungscode kodiert 8-Bit-Blöcke in je sechs Symbole, von denen jedes den Zustand `-`, `0` oder `+` repräsentieren kann. Die Symbole der Zustände sind stellvertretend für elektrische Signalpegel. Die Kodierung wird anhand einer Tabelle (siehe Tabelle 5.13) durchgeführt, die alle $2^8 = 256$ möglichen 8-Bit-Kombinationen enthält. Tabelle 5.13 enthält aus Platzgründen nur die ersten 48 der 256 möglichen 8-Bit-Kombinationen. Fast-Ethernet 100BASE-T4 verwendet diesen Leitungscode.

In contrast to the line codes 4B5B, 5B6B and 8B10B, which only *improve* the payload and require an encoding with NRZ(I) or MLT-3 afterward, 8B6T encoded data can be used directly for transmission.

As shown in Table 5.13, the output of 8B6T makes baseline wander impossible, and the many signal level changes allow clock recovery for the receiver.

Im Gegensatz zu den Leitungscodes 4B5B, 5B6B und 8B10B, die die Nutzdaten nur *verbessern* und anschließend eine Kodierung mit NRZ(I) oder MLT-3 erfordern, kann eine 8B6T-Kodierung unmittelbar für die Übertragung genutzt werden.

Wie in Tabelle 5.13 zu sehen ist, macht die Ausgabe von 8B6T Durchschnittsverschiebungen unmöglich und die häufigen Wechsel des Signalpegels ermöglichen die Taktrückgewinnung für den Empfänger.

6

Data Link Layer

The *Data Link Layer* is the second layer of the OSI reference model and the hybrid reference model. In this layer, the frames in the bitstream of the Physical Layer are identified, and the packets of the Network Layer are packed into *frames*.

One task of the Data Link Layer is to ensure the error-free transfer of frames on a physical network from one network device to another. The delivery of the frames requires physical addresses (MAC addresses) whose format defines the protocols of the Data Link Layer.

The Data Link Layer protocols attach a checksum to each frame for error detection so that frames with errors are detected and thrown away by the receiver. The Data Link Layer does not provide functionality to request again frames that have previously been discarded.

Also, protocols of this layer control the access to the transmission medium (for example, via CSMA/CD or CSMA/CA).

Sicherungsschicht

Die *Sicherungsschicht* ist die zweite Schicht des OSI-Referenzmodells und des hybriden Referenzmodells. In dieser Schicht werden die Rahmen im Bitstrom der Bitübertragungsschicht erkannt und die Pakete der Vermittlungsschicht werden in *Rahmen* verpackt.

Eine Aufgabe der Sicherungsschicht ist die korrekte Übertragung der Rahmen auf einem physischen Netz von einem Netzwerkgerät zum anderen zu gewährleisten. Für die Zustellung der Rahmen sind physische Adressen (MAC-Adressen) nötig, deren Format die Protokolle der Sicherungsschicht definieren.

Protokolle der Sicherungsschicht fügen zur Fehlererkennung jedem Rahmen eine Prüfsumme an, damit fehlerhafte Rahmen vom Empfänger erkannt und verworfen werden. Das erneute Anfordern verworfener Rahmen sieht die Sicherungsschicht aber nicht vor.

Zudem regeln Protokolle dieser Schicht den Zugriff auf das Übertragungsmedium (zum Beispiel via CSMA/CD oder CSMA/CA).

6.1

Devices of the Data Link Layer

Devices of the Physical Layer increase the maximum expansion of physical networks. However, if frames shall be forwarded from one physical network to another, *Bridges* are required. A Bridge only has two ports. Simple Bridges forward all incoming frames. Learning Bridges filter the data transfers by forwarding frames only if it makes sense because of the known or unknown position of the receiver.

Geräte der Sicherungsschicht

Geräte der Bitübertragungsschicht verlängern physische Netze. Sollen aber Rahmen von einem physischen Netz in andere weitergeleitet werden, sind *Bridges* nötig. Eine Bridge verfügt lediglich über zwei Schnittstellen. Einfaches Bridges leiten alle eintreffenden Rahmen weiter. Lernende Bridges filtern die Datenübertragungen, indem sie Rahmen nur weiterleiten, wenn es aufgrund der bekannten oder unbekannten Position des Empfängers sinnvoll ist.

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_6

Bridges with more than two ports are called *Multiport Bridge* or *Layer-2-Switch* (see Figure 6.1).

Bridges mit mehr als zwei Schnittstellen heißen *Multiport-Bridge* oder *Layer-2-Switch* (siehe Abbildung 6.1).

Figure 6.1: Layer-2-Switch with 16 Interfaces (Ports)

Bridges and Switches check the correctness of the frames by using checksums. They do not need an address to filter and forward the frames, because they do not actively participate in communication. They operate transparently like the devices of the Physical Layer because they do not communicate on a higher protocol layer than the Data Link Layer.

Examples of Bridges are WLAN Bridges, which integrate network devices (e.g., network printers, desktops, gaming consoles,...) with an Ethernet interface into a wireless local area network (WLAN) and Laser Bridges, which use a laser beam to connect two different buildings, as well as Modems, ISDN devices and WLAN access points.

Telephone Modems and ISDN devices use the Point-to-Point protocol (PPP), which belongs to the Data Link Layer, to establish, maintain and terminate point-to-point connections. (A)DSL Modems and Cable Modems use PPP over Ethernet (PPPoE) for the same purpose.

Gateways are protocol translators and enable communication between networks that use different protocols. A Gateway converts between protocols and can theoretically work on all layers of the hybrid reference model. VPN Gateways (Virtual Private Network) can also operate on the Data Link Layer and provide secure access to remote protected networks (e.g., university or corporate networks) via insecure public net-

Bridges und Switche untersuchen die Rahmen mit Prüfsummen auf Korrektheit. Zum Filtern und Weiterleiten der Rahmen brauchen sie keine Adresse, da sie selbst nicht aktiv an der Kommunikation teilnehmen. Sie arbeiten wie die Geräte der Bitübertragungsschicht transparent, da sie nicht auf einer höheren Protokollschicht als der Sicherungsschicht kommunizieren.

Beispiele für Bridges sind WLAN-Bridges, die Netzwerkgeräte (z.B. Netzwerkdrucker, Desktops, Spielkonsolen,...) mit Ethernet-Anschluss in ein WLAN integrieren und Laser-Bridges, die per Laserstrahl zwei unterschiedliche Gebäude verbinden sowie Modems, ISDN-Anschlussgeräte und WLAN-Basisstationen (Access Points).

Telefonmodems und ISDN-Anschlussgeräte verwenden das Protokoll der Sicherungsschicht Point-to-Point Protocol (PPP) um Punkt-zu-Punkt-Verbindungen aufzubauen, aufrecht zu erhalten und zu beenden. (A)DSL-Modems und Kabelmodems verwenden zum gleichen Zweck PPP over Ethernet (PPPoE).

Gateways sind Protokollumsetzer und ermöglichen Kommunikation zwischen Netzen, die auf unterschiedlichen Protokollen basieren. Ein Gateway konvertiert zwischen Protokollen und kann theoretisch auf allen Schichten des hybriden Referenzmodells arbeiten. Auch VPN-Gateways (Virtual Private Network) können auf der Sicherungsschicht arbeiten und ermöglichen über unsichere öffentliche Netze den sicheren

works. Services (e.g., email) that are only available inside the protected network are used via a tunneled connection.

Zugriff auf entfernte geschützte Netze (zum Beispiel Hochschul /Firmennetze). Dienste (zum Beispiel Email), die nur innerhalb des geschützten Netzes zur Verfügung stehen, werden über eine getunnelte Verbindung genutzt.

6.1.1

Learning Bridges

Figure 6.2 shows that it does not make sense for a Bridge to forward all frames. For example, if a frame from participant B for participant A arrives at port 1 of the Bridge, it is not required that the Bridge forwards this frame via port 2.

Learning Bridges

Abbildung 6.2 zeigt, dass es nicht sinnvoll ist, wenn eine Bridge alle Rahmen weiterleitet. Kommt zum Beispiel ein Rahmen von Teilnehmer B für Teilnehmer A an Schnittstelle 1 der Bridge an, ist es nicht nötig, dass die Bridge diesen Rahmen über Schnittstelle 2 weiterleitet.

Figure 6.2: A Bridge connects two physical Networks

Bridges need to learn which network devices are accessible via which port. Manual intervention is not required, because Bridges maintain their own *forwarding tables*. A Bridge stores the sender addresses of the frames that reaches it. Over time, the Bridge populates a forwarding table that shows which network devices are in the connected physical networks.

During the start-up of a Bridge, its forwarding table is empty. The entries are recorded over time. All entries have an expiration date – *Time To Live* (TTL). They are therefore only valid for a certain period. The forwarding table is not necessarily complete. However, this is not a problem as it serves for optimization. If there is no entry in the forwarding table for a network device, the Bridge forwards the frame in any case.

Die Bridges müssen lernen, welche Netzwerkgeräte über welche Schnittstelle erreichbar sind. Manuelle Eingriffe sind nicht nötig, da die Bridges ihre *Weiterleitungstabellen* selbst pflegen. Eine Bridge speichert die Absenderadressen der Rahmen, die sie erreichen. So kann sie sich mit der Zeit eine Weiterleitungstabelle aufbauen, aus der hervorgeht, welche Netzwerkgeräte sich in den verbundenen physischen Netzen befinden.

Beim Hochfahren einer Bridge ist ihre Weiterleitungstabelle leer. Die Einträge werden erst im Laufe der Zeit erfasst. Alle Einträge haben ein Verfallsdatum – *Time To Live* (TTL). Sie sind also nur eine bestimmte Zeit gültig. Die Weiterleitungstabelle ist nicht unbedingt vollständig. Das ist aber kein Problem, da sie zur Optimierung dient. Existiert für ein Netzwerkgerät kein Eintrag in der Weiterleitungstabelle, leitet die Bridge den Rahmen in jedem Fall weiter.

6.1.2

Loops on the Data Link Layer

Loops are potential problems because computer networks should have only a single path at a time to any possible destination on the Data Link Layer. This is to prevent frames from being duplicated, and arriving multiple times at their destination. Loops can reduce the performance of the network or even cause a network failure. On the other hand, redundant connections serve as a backup in case of a cable failure [15].

Kreise auf der Sicherungsschicht

Ein potentielles Problem sind *Kreise*, denn Computernetze sollten auf der Sicherungsschicht zu jedem möglichen Ziel immer nur einen Pfad haben. Das soll vermeiden, dass Rahmen dupliziert werden und mehrfach am Ziel eintreffen. Kreise können die Leistung des Netzes vermindern oder sogar zum Totalausfall führen. Andererseits dienen redundante Netzpfade als Backup für den Ausfall einer Leitung [15].

Figure 6.3: Cyclic Graph and Spanning Tree

Figure 6.4 illustrates the potential impact of loops on the Data Link Layer. In the presented scenario, which consists of several phases, three Switches are connected as a loop, and each is connected to one terminal device. In the context of this example, the terminal devices are called nodes. A similar example is shown in [15].

- Phase 1: Initially, the forwarding tables of the Switches are empty, and node A wants to send a frame to node B.
- Phase 2: The frame has reached Switch 1, and it records the local port to node A in its forwarding table. Since Switch 1 does not know the direct port to node C, it sends copies of the frame via all of its ports (except port 1).
- Phase 3: The copies of the frame reach Switch 2 and 3, which each record the local port to node A in their forwarding tables. However, they do not know the direct port to node C either, so they forward copies of the frame through all of their ports except the ports where the frame reached

Abbildung 6.4 verdeutlicht die möglichen Auswirkungen von Kreisen auf der Sicherungsschicht. Im gezeigten Szenario, das aus mehreren Phasen besteht, sind drei Switches zu einem Kreis und mit jeweils einem Endgerät verbunden. Im weiteren Verlauf des Beispiels heißen die Endgeräte einfach Knoten. Ein ähnliches Beispiel enthält [15].

- Phase 1: Initial sind die Weiterleitungstabellen der Switche leer und Knoten A will einen Rahmen an Knoten B senden.
- Phase 2: Der Rahmen hat Switch 1 erreicht und dieser trägt den lokalen Port zu Knoten A in seine Weiterleitungstabelle ein. Da Switch 1 den direkten Port zu Knoten C nicht kennt, sendet er Kopien des Rahmens über alle seine Ports (außer Port 1).
- Phase 3: Die Kopien des Rahmens erreichen Switch 2 und 3, die jeweils den lokalen Port zu Knoten A in ihre Weiterleitungstabellen eintragen. Den direkten Port zu Knoten C kennen sie allerdings auch nicht und darum leiten sie Kopien des Rahmens über alle ihre Ports weiter, au-

Switch 2 and 3. It is interesting to observe that a copy of this frame also reaches the destination (node B). However, because Ethernet does not implement acknowledgments, the Switches on the network and the sender (node A) do not learn anything about it.

- Phase 4: Again, copies of the frame arrive at Switch 2 and 3, and both Switches update their forwarding tables. Also, the Switches send copies of the frame over all their local ports, except via the ports where the frame has reached Switch 2 and 3. Moreover, a copy of the frame reaches the destination.

- Phase 5: Two copies of the frame reach Switch 1. Now copies of the original frame have returned to Switch 1 for the first time, and there is a loop. Switch 1 sends copies of the frame it received on port 2 to port 1 and 3, and it sends copies of the frame it received on port 3 to port 1 and 2. Each time, Switch 1 receives a copy of the frame, it also updates its forwarding table. It is impossible to predict the order in which the frames reach Switch 1. It is interesting to see that two copies of the frame reach the sender also in phase 5.

- Phase 6: Copies of the frame pass Switch 2 and 3 again, both Switches update their forwarding tables and create copies which they forward.

The example demonstrates that each frame of node A creates two copies that circle endlessly in the network. Ethernet has no time limit – Time To Live (TTL) – and does not define a maximum number of hops. In this specific case, the frames travel in the loop until the forwarding tables of the Switches contain a correct record for node B. Further frames sent by node A flood

ßer über die Ports, an denen der Rahmen Switch 2 und 3 erreicht hat. Interessanterweise erreicht eine Kopie des Rahmens auch das Ziel (Knoten B). Da Ethernet aber keine Bestätigungen vorsieht, erfahren die Switches und der Absender (Knoten A) nichts davon.

- Phase 4: Erneut erreichen Kopien des Rahmens Switch 2 und 3 und beide Switche aktualisieren ihre Weiterleitungstabellen. Zudem senden die Switche erneut Kopien des Rahmens über alle ihre Ports, außer über die Ports, an denen der Rahmen Switch 2 und 3 erreicht hat. Und erneut erreicht eine Kopie des Rahmens das Ziel.

- Phase 5: Zwei Kopien des Rahmens erreichen Switch 1. Nun sind erstmals Kopien des ursprünglichen Rahmens zu Switch 1 zurückgekehrt und es liegt eine Schleife vor. Switch 1 wird Kopien des Rahmens, die er über Port 2 empfangen hat, an Port 1 und 3 senden und er wird Kopien des Rahmens, die er über Port 3 empfangen hat, an Port 1 und 2 senden. Bei jedem Empfang einer Kopie des Rahmens wird Switch 1 zudem seine Weiterleitungstabelle aktualisieren. Die Reihenfolge, in der die Kopien des Rahmens Switch 1 erreichen, kann nicht vorhergesagt werden. Interessanterweise erreichen in Phase 5 auch zwei Kopien des Rahmens den Absender.

- Phase 6: Kopien des Rahmens passieren erneut Switch 2 und 3, beide Switche aktualisieren ihre Weiterleitungstabellen und erzeugen Kopien, die sie weiterleiten.

Das Beispiel zeigt, dass jeder Rahmen von Knoten A zwei Kopien verursacht, die endlos im Netz kreisen. Ethernet hat kein Zeitlimit – Time To Live (TTL) – und definiert auch keine maximale Anzahl an Weiterleitungen (Hops). Im konkreten Fall kreisen die Rahmen so lange, bis die Tabellen der Switche einen korrekten Eintrag für Knoten B enthalten. Das Senden weiterer

the network and cause it to collapse at some point.

Rahmen durch Knoten A flutet das Netz und lässt es irgendwann zusammenbrechen.

Figure 6.4: Possible Impact of Loops in the Data Link Layer

Therefore, Bridges must be able to handle loops. One solution is the *Spanning Tree Algorithm*. A computer network that consists of multiple physical networks is a graph that may contain loops. The spanning tree is a subgraph of the graph that covers all nodes, but

Bridges müssen in der Lage sein, Kreise zu handhaben. Eine Lösung ist der *Spanning Tree Algorithmus*. Ein Computernetz, das aus mehreren physischen Netzen besteht, ist ein Graph, der möglicherweise Kreise enthält. Der Spannbaum ist ein Teilgraph des Graphen, der alle

is cycle-free, because edges have been removed (see Figure 6.3). The implementation of the algorithm is the *Spanning Tree Protocol* (STP).

Knoten abdeckt, aber kreisfrei ist, weil Kanten entfernt wurden (siehe Abbildung 6.3). Die Implementierung des Algorithmus ist das *Spanning Tree Protocol* (STP).

6.1.3 Spanning Tree Protocol (STP)

By using STP, a group of Bridges can agree on a spanning tree. The computer network is reduced to a loop-free tree by removing individual ports of the Bridges. The algorithm works in a dynamic way. If a Bridge fails, a new spanning tree is created. The protocol and the structure of the configuration messages are described in detail in IEEE 802.1D.

Structure of the Identifier (Bridge ID)

For the STP to work, each Bridge must have a unique identifier. The length of the identifier (*Bridge ID*) is 8 bytes. Two different representations of the Bridge ID exist. According to the IEEE, it consists of the Bridge priority (2 bytes) and MAC address (6 bytes) of the Bridge port with the lowest port ID (see Figure 6.5). The administrator can set the Bridge priority with a value between 0 and 65,536. The default value is 32,768.

Spanning Tree Protocol (STP)

Mit dem STP kann sich eine Gruppe Bridges auf einen Spannbaum einigen. Dabei wird das Computernetz durch das Entfernen einzelner Ports der Bridges auf einen kreisfreien Baum reduziert. Der Algorithmus arbeitet dynamisch. Fällt eine Bridge aus, wird ein neuer Spannbaum erzeugt. Das Protokoll und der Aufbau der Konfigurationsnachrichten sind detailliert im Standard IEEE 802.1D beschrieben.

Aufbau der Kennung (Bridge-ID)

Damit das STP funktionieren kann, muss jede Bridge eine eindeutige Kennung haben. Die Kennung (*Bridge-ID*) ist 8 Bytes lang. Es existieren zwei unterschiedliche Darstellungen der Bridge-ID. Gemäß IEEE besteht sie aus der Bridge Priority (2 Bytes) und der MAC-Adresse (6 Bytes) des Bridge-Ports mit der niedrigsten Port-ID (siehe Abbildung 6.5). Die Bridge Priority kann vom Administrator selbst festgelegt werden und hat einen beliebigen Wert zwischen 0 und 65.536. Der Standard-Wert ist 32.768.

Figure 6.5: Structure of the Bridge ID

The extension of the Bridge ID, that was initially developed by Cisco, makes it possible for each virtual LAN (VLAN, see section 10.2) to build its own spanning tree. The original 2 bytes long part of the Bridge priority is subdivided for this purpose. 4 bits now represent the Bridge priority. This means that only 16 values can be represented. The value of the bridge priority must therefore be zero or a multiple of 4,096. Possible values are `0000` = 0, `0001` = 4,096, `0010` = 8,192, ... `1110` = 57,344, `1111` = 61,440. The default value is still 32,768. The remaining 12 bits encode the VLAN ID, the so-

Die ursprünglich von Cisco entwickelte Erweiterung der Bridge-ID macht es möglich, dass jedes virtuelle LAN (VLAN, siehe Abschnitt 10.2) seinen eigenen Spannbaum aufbaut. Dafür wird der ursprünglich 2 Bytes große Anteil der Bridge Priority unterteilt. 4 Bits kodieren nun die Bridge Priority. Damit lassen sich nur noch 16 Werte darstellen. Der Wert der Bridge Priority muss darum Null oder ein Vielfaches von 4.096 sein. Mögliche Werte sind `0000` = 0, `0001` = 4.096, `0010` = 8.192, ... `1110` = 57.344, `1111` = 61.440. Der Standard-Wert ist unverändert 32.768. Die übrigen 12 Bits ko-

called *Extended System ID*, which matches the content of the VLAN tag in the Ethernet frame (see Figure 6.6). With 12 bits, 4,096 different VLANs can be addressed.

dieren die VLAN-ID, die sogenannte *Extended System ID*, die mit dem Inhalt des VLAN-Tags im Ethernet-Rahmen übereinstimmt (siehe Abbildung 6.6). Mit 12 Bits können 4.096 unterschiedliche VLANs adressiert werden.

Figure 6.6: Cisco Extension of the Bridge ID, introducing the Extended System ID

Functioning of the STP

The messages that the Bridges communicate with are called *Bridge Protocol Data Unit* (BPDU). They are inserted in the payload field of Ethernet frames and sent via broadcast to the neighboring Bridges (see Figure 6.7).

First, the Bridges among themselves select the Bridge with the lowest Bridge priority in the Bridge ID as the *root* of the tree to be spanned. If the Bridge priority is identical for multiple Bridges, the Bridge with the lowest MAC address becomes the root.

For each physical network, one of the directly connected Bridges must be selected as responsible for forwarding the frames to the direction of the root. This Bridge is called *designated Bridge* for this specific network. The Bridge that is selected is always the one via which the root can be reached at the lowest path cost. The path cost to the root is calculated as the sum of the path cost of the individual physical networks on the way to the root. The various network technologies that can be used cause different path costs, which depend on the available data rate. Path costs are standardized by the IEEE (see Table 6.1) but can be adjusted manually.

If two or more Bridges of a physical network have the same path cost to the root, the Bridge with the lowest Bridge ID becomes the designated Bridge. Since each Bridge is connected to more than one network, it participates with each interface (port) in the selection for each connected network.

Arbeitsweise des STP

Die Nachrichten, mit denen die Bridges kommunizieren, heißen *Bridge Protocol Data Unit* (BPDU). Sie werden im Datenfeld von Ethernet-Rahmen via Broadcast an die benachbarten Bridges gesendet (siehe Abbildung 6.7).

Zuerst wählen die Bridges untereinander die Bridge mit der niedrigsten Bridge Priority in der Bridge-ID als *Wurzel* des aufzuspannenden Baums. Ist die Bridge Priority bei mehreren Bridges identisch, wird die Bridge mit der niedrigsten MAC-Adresse die Wurzel.

Für jedes physische Netz muss aus allen mit dem Netz direkt verbunden Bridges eine Bridge ausgewählt werden, die für die Weiterleitung der Rahmen in Richtung Wurzel zuständig ist. Dieses Bridge ist dann die *designierte Bridge* für das betreffende Netz. Es wird immer diejenige Bridge ausgewählt, über die zu den geringsten Pfadkosten die Wurzel erreicht werden kann. Die Pfadkosten zur Wurzel sind die Summe der Pfadkosten der einzelnen physischen Netze auf dem Weg zur Wurzel. Die unterschiedlichen Netzwerktechnologien, die zum Einsatz kommen können, verursachen unterschiedliche Pfadkosten, die von der möglichen Datendurchsatzrate abhängen. Die Pfadkosten sind durch die IEEE genormt (siehe Tabelle 6.1), können aber manuell angepasst werden.

Haben zwei oder mehr Bridges eines physischen Netzes die gleichen Pfadkosten zur Wurzel, wird diejenige Bridge mit der kleinsten Bridge-ID zur designierten Bridge. Da jede Bridge an mehr als ein Netz angeschlossen ist, nimmt sie mit jeder Schnittstelle (Port) bei der Auswahl für jedes verbundene Netz teil.

Figure 6.7: Configuration Messages (BPDU) of the STP

Table 6.1: Default Path Costs of the STP

Data rate	Path cost
10,000 Mbps	2
1,000 Mbps	4
100 Mbps	19
16 Mbps	62
10 Mbps	100
4 Mbps	250

Initially, each Bridge considers itself the root and sends a BPDU via all its ports, identifying itself as the root with zero path cost to the root. When a Bridge receives a BPDU, it checks whether the root information that is contained is better than that currently stored for that port. A BPDU is considered to be better when it identifies a root with...

- a lower Bridge ID.
- the same Bridge ID, but lower path costs.
- the same Bridge ID and the same path costs, but the Bridge ID of the sending Bridge is lower.

If an incoming BPDU contains better information than the current root information that is stored for that port, the Bridge adds its path costs and stores the new root information.

If a Bridge receives a BPDU, which implies that it is not the root, because the BPDU comes from a Bridge with a lower Bridge ID, it stops sending its own BPDUs (see Figure 6.8). This

Anfangs hält sich jede Bridge für die Wurzel und sendet eine BPDU über alle ihre Ports, mit der sie sich als Wurzel mit den Pfadkosten Null zur Wurzel identifiziert. Empfängt eine Bridge eine BPDU, prüft sie, ob die darin enthaltenen Wurzelinformationen besser sind als die momentan für diesen Port gespeicherten. Eine BPDU gilt als besser, wenn sie eine Wurzel...

- mit einer kleineren Bridge-ID identifiziert.
- mit gleicher Bridge-ID, aber geringeren Pfadkosten identifiziert.
- mit gleicher Bridge-ID und gleichen Pfadkosten identifiziert, aber die Bridge-ID der sendenden Bridge kleiner ist.

Sind die Daten einer eingetroffenen BPDU besser als die momentan für diesen Port gespeicherte Wurzelinformation, addiert die Bridge ihre eigenen Pfadkosten und speichert die neue Wurzelinformation.

Empfängt eine Bridge eine BPDU, aus der hervorgeht, dass sie nicht die Wurzel ist, weil die BPDU von einer Bridge mit kleinerer Kennung stammt, hört sie mit dem Senden eigener

Bridge then only forwards BPDUs from other Bridges after it has added its path costs.

BPDUs auf (siehe Abbildung 6.8). Die Bridge leitet dann nur noch BPDUs anderer Bridges weiter, nachdem sie ihre eigenen Pfadkosten aufaddiert hat.

Figure 6.8: If a Bridge receives a BPDU from a Bridge with a lower Bridge ID, it does not send own BPDUs any more

If a Bridge receives a BPDU, which implies that it is not the Designated Bridge for a physical network, that is connected to this local port, because the BPDU comes from a Bridge which can reach the Root Bridge for lesser path cost or has a lower Bridge ID, the Bridge which received the BPDU does not forward any more BPDUs via this port (see Figure 6.9).

Empfängt eine Bridge eine BPDU, aus der hervorgeht, dass sie nicht die designierte Bridge für das Netz an diesem Port ist, weil die Nachricht von einer Bridge stammt, die näher an der Wurzel liegt, oder eine kleinere Kennung hat, hört sie mit dem Weiterleiten von BPDUs über diesen Port auf (siehe Abbildung 6.9).

Figure 6.9: If a Bridge receives a BPDU, which implies that it is not the designated Bridge for a physical Network, which is connected to this local Port, it does not forward any more BPDUs via this Port

At some point, the system has stabilized. Then, by default, only the root sends BPDUs every two seconds across all its ports. Only the designated Bridges forward these BPDUs over the corresponding ports. In the event of a Bridge failure, the Bridges which are located behind do not receive any more BPDUs. After the expiration of a waiting period, these Bridges act again as roots and the algorithm starts from the beginning with the selection of a root and a Bridge for each network.

Irgendwann hat sich das System stabilisiert. Dann sendet nur noch die Wurzel standardmäßig alle zwei Sekunden BPDUs über alle ihre Ports. Nur die designierten Bridges leiten diese BPDUs über die entsprechenden Ports weiter. Fällt eine Bridge aus, erhalten die hinter der ausgefallenen Bridge liegenden Bridges keine BPDUs mehr. Nach Ablauf einer Wartezeit, geben sich diese Bridges wieder jeweils als Wurzeln aus und der Algorithmus startet von vorne mit der Auswahl einer Wurzel und einer Bridge für jedes Netz.

The STP can reconfigure the spanning tree after the failure of a Bridge. However, the pro-

Das STP ist in der Lage, nach dem Ausfall einer Bridge den Spannbaum neu zu konfigurie-

tocol is not able to route frames via alternative paths if a Bridge is overloaded.

ren. Das Protokoll ist aber nicht in der Lage, Rahmen über alternative Wege zu leiten, wenn eine Bridge überlastet ist.

Figure 6.10: Collision Domains of Bridges and Layer-2-Switches

6.1.4

Impact of Bridges on the Collision Domain

Bridges and Switches operate on the Data Link Layer and forward frames from one physical network to another. Each physical network is a separate collision domain. If a Bridge or a Switch split a physical network, then also the collision domain is split and the number of collisions thus decreases. For Bridges and Switches, each port forms its own collision domain (see Figure 6.10).

In a *fully switched network*, only one network device is connected to each port of a Switch. Such a network is free of collisions and state of the art.

Auswirkungen von Bridges auf die Kollisionsdomäne

Bridges und Switche arbeiten auf der Sicherungsschicht und leiten Rahmen von einem physischen Netz zu anderen. Jedes physische Netz ist eine eigene Kollisionsdomäne. Unterteilt man ein physisches Netz durch eine Bridge oder einen Switch, unterteilt man auch die Kollisionsdomäne und die Anzahl der Kollisionen sinkt. Bei Bridges und Switches bildet jeder Port eine eigene Kollisionsdomäne (siehe Abbildung 6.10).

In einem *vollständig geswitchten Netz* ist mit jedem Port eines Switches nur ein Netzwerkgerät verbunden. Ein solches Netzwerk ist frei von Kollisionen und Stand der Technik.

6.2

Addressing in the Data Link Layer

The protocols of the Data Link Layer define the format of the physical addresses. Terminal devices (hosts), Routers and Layer-3-Switches require physical addresses. They must be addressable on the Data Link Layer because they offer services on upper protocol layers. Bridges and Layer-2-Switches do not actively participate in the communication and do not require physical addresses for their basic functionality, which is the filtering and forwarding of frames.

Adressierung in der Sicherungsschicht

Die Protokolle der Sicherungsschicht definieren das Format der physischen Adressen. Endgeräte (Hosts), Router und Layer-3-Switche benötigen zwingend physische Adressen. Sie müssen weil sie auf der Sicherungsschicht adressierbar sein, um Dienste auf höheren Schichten anzubieten. Bridges und Layer-2-Switche nehmen nicht aktiv an der Kommunikation teil und brauchen für ihre Basisfunktionalität, also das Filtern und Weiterleiten der Rahmen, keine physischen Adressen.

Bridges and Switches require physical addresses if they implement the STP to avoid loops, or offer services from an upper protocol layer. Examples include monitoring services or graphical web interfaces for administration. Repeaters and Hubs that operate only on the Physical Layer have no addresses.

The physical addresses called *MAC addresses* (Media Access Control) are independent of the logical addresses of the Network Layer. Ethernet uses the *Address Resolution Protocol* (ARP) to resolve the logical addresses of the Network Layer (IPv4 addresses) into MAC addresses. IPv6 uses the *Neighbor Discovery Protocol* (NDP), which has identical functionality and operates in a similar way.

Bridges und Switche benötigen physische Adressen, wenn sie das STP zur Vermeidung von Kreisen anwenden, oder Dienste aus einer höheren Schicht anbieten. Beispiele sind Monitoring-Dienste zur Überwachung oder grafische Weboberflächen zur Administration. Repeater und Hubs, die nur auf der Bitübertragungsschicht arbeiten, haben keine Adressen.

Die physischen Adressen, die *MAC-Adressen* (Media Access Control) heißen, sind unabhängig von den logischen Adressen der Vermittlungsschicht. Mit dem *Address Resolution Protocol* (ARP) werden bei Ethernet die logischen Adressen der Vermittlungsschicht (IPv4-Adressen) in MAC-Adressen aufgelöst. Bei IPv6 wird das *Neighbor Discovery Protocol* (NDP) verwendet, dessen Funktionalität identisch ist und das ähnlich arbeitet.

6.2.1

Format of MAC Addresses

MAC addresses have a length of 48 bits (6 bytes). Thus, a total of 2^{48} addresses are representable. To make the representation compact and easy to read, MAC addresses are usually written in hexadecimal notation, and the individual bytes are separated from each other with dashes or colons. One example of this notation is `00-16-41-52-DF-D7`.

Each MAC address is intended to be permanently assigned to a network device and unique. However, it is often possible to modify MAC addresses by software. On the other hand, this modification only applies until the next reboot of the computer.

If a network device wants to send a frame to all other devices in the same physical network, it inserts MAC broadcast address in the destination address field of the frame. All 48 bits of this MAC address have the value `1`. The hexadecimal notation is `FF-FF-FF-FF-FF-FF`. Frames that have the broadcast address in the destination field are not forwarded to other physical networks by Bridges and Switches.

Format der MAC-Adressen

MAC-Adressen sind 48 Bits (6 Bytes) lang. Damit sind insgesamt 2^{48} Adressen darstellbar. Um die Darstellung kompakt und gut lesbar zu gestalten, sind MAC-Adressen normalerweise in hexadezimaler Schreibweise geschrieben und die einzelnen Bytes durch Bindestriche oder Doppelpunkte voneinander getrennt. Ein Beispiel für diese Schreibweise ist `00-16-41-52-DF-D7`.

Jede MAC-Adresse soll dauerhaft einem Netzwerkgerät zugewiesen und eindeutig sein. Es ist aber auch meist möglich, MAC-Adressen softwaremäßig zu ändern. Allerdings gilt diese Änderung nur bis zum nächsten Neustart des Rechners.

Möchte ein Netzwerkgerät einen Rahmen an alle anderen Geräte im gleichen physischen Netz senden, fügt es im Rahmen in das Feld der Zieladresse die Broadcast-Adresse ein. Bei dieser MAC-Adresse haben alle 48 Bits den Wert Eins. Die hexadezimale Schreibweise ist `FF-FF-FF-FF-FF-FF`. Rahmen, die im Zielfeld die Broadcast-Adresse tragen, werden von Bridges und Switches nicht in andere physische Netze übertragen.

Table 6.2: Some Organizationally Unique Identifiers of popular Hardware Vendors

MAC addresses	Manufacturer	MAC addresses	Manufacturer
`00-20-AF-xx-xx-xx`	3COM	`00-03-93-xx-xx-xx`	Apple
`00-0C-6E-xx-xx-xx`	Asus	`00-00-0C-xx-xx-xx`	Cisco
`00-50-8B-xx-xx-xx`	Compaq	`08-00-2B-xx-xx-xx`	DEC
`00-01-E6-xx-xx-xx`	HP	`00-02-55-xx-xx-xx`	IBM
`00-02-B3-xx-xx-xx`	Intel	`00-04-5A-xx-xx-xx`	Linksys
`00-09-5B-xx-xx-xx`	Netgear	`00-04-E2-xx-xx-xx`	SMC

6.2.2

Uniqueness of MAC Addresses

The Institute of Electrical and Electronics Engineers (IEEE) manages the first 24 bits of the MAC address space. These 24 bits long partial addresses are hardware vendor IDs, called *OUI* (Organizationally Unique Identifier) and can be checked in a database [8] that is maintained by the IEEE. The remaining 24 bits are specified by the hardware vendors themselves for each network device. This allows $2^{24} = 16,777,216$ individual device addresses. Table 6.2 contains some OUIs of popular hardware vendors.

There is also a small MAC address range that is intended for private individuals and small companies or organizations that require only a few addresses. This address range is called *IAB* (Individual Address Block) and starts with `00-50-C2`. This is followed by three further hexadecimal digits (12 bits), which are assigned to each organization. This allows $2^{12} = 4,096$ device addresses per IAB.

Eindeutigkeit von MAC-Adressen

Das Institute of Electrical and Electronics Engineers (IEEE) verwaltet die ersten 24 Bits des MAC-Adressraums. Diese 24 Bits langen Teiladressen sind Herstellerkennungen, heißen *OUI* (Organizationally Unique Identifier) und sind in einer Datenbank [8] des IEEE einsehbar. Die übrigen 24 Bits legen die Hersteller selbst für jedes Netzwerkgerät fest. Das ermöglicht $2^{24} = 16.777.216$ individuelle Geräteadressen. Tabelle 6.2 enthält einige Herstellerkennungen bekannter Gerätehersteller.

Neben den Herstellerkennungen existiert auch ein kleiner MAC-Adressbereich, der für Privatpersonen und kleine Firmen oder Organisationen vorgesehen ist, die nur wenige Adressen benötigen. Dieser Adressbereich heißt *IAB* (Individual Address Block) und beginnt mit `00-50-C2`. Darauf folgen drei weitere hexadezimale Ziffern (12 Bits), die für jede Organisation vergeben werden. Das ermöglicht $2^{12} = 4.096$ Geräteadressen pro IAB.

6.2.3

Security Aspects of MAC Addresses

In WLANs, the MAC address is often used with a MAC filter to protect access to the access point. This makes sense because the MAC address is the unique identification feature of a network device. However, the protection of MAC filters is weak because MAC addresses can be modified by software.

If a MAC filter is used in a WLAN, the wireless network is only protected as long as it is not used by any other node. Once network de-

Sicherheit von MAC-Adressen

In WLANs wird häufig mit einem MAC-Filter die MAC-Adresse als Zugangsschutz zur Basisstation (Access Point) verwendet. Im Prinzip ist das sinnvoll, da die MAC-Adresse das eindeutige Identifikationsmerkmal eines Netzwerkgeräts ist. Der Schutz von MAC-Filtern ist aber gering, da MAC-Adressen softwaremäßig verändert werden können.

Kommt ein MAC-Filter bei einem WLAN zum Einsatz, ist das Funknetz nur so lange sicher, solange es von keinem Teilnehmer ver-

vices connect to the access point, their MAC addresses can be monitored by third parties and used by attackers [2].

wendet wird. Sobald sich Netzwerkgeräte mit der Basisstation verbinden, können ihre MAC-Adressen durch Dritte mitgelesen und von Angreifern verwendet werden [2].

6.3

Framing

To divide the bitstream of the Physical Layer into frames and pack the data of the Network Layer into frames, the beginning of each frame must be highlighted. This is necessary to allow the receiver to detect the frame's boundaries. The benefits and drawbacks of the different approaches are discussed in the following sections.

Rahmen abgrenzen

Um den Bitstrom der Bitübertragungsschicht in Rahmen zu unterteilen und die Daten der Vermittlungsschicht in Rahmen zu verpacken, muss der Anfang jedes Rahmens markiert werden. Das ist nötig, damit der Empfänger die Rahmengrenzen erkennt. Die Vor- und Nachteile der unterschiedlichen Vorgehensweisen werden in den folgenden Abschnitten diskutiert.

6.3.1

Character Count in the Frame Header

A Data Link Layer protocol that specifies the frame boundary with length information in the header is the byte-oriented *Digital Data Communications Message Protocol* (DDCMP) of DECnet. The frame contains the field *Count*, which specifies the number of payload bytes (see Figure 6.12).

Längenangabe im Header

Ein Protokoll der Sicherungsschicht, das die Rahmengrenze mit einer Längenangabe im Header angibt, ist das Byte-orientierte *Digital Data Communications Message Protocol* (DDCMP) des DECnet. Im Rahmen befindet sich das Datenfeld *Count*, das die Anzahl der Nutzdaten in Bytes enthält (siehe Abbildung 6.12).

Figure 6.11: Frame of the Protocol DDCMP [4]

Another example is the very first Ethernet *DIX frame format* from the early 1980s which is no longer in use today. The abbreviation stands for the companies Digital Equipment Corporation, Intel and Xerox that were involved in its development. In contrast to the current Ethernet frame format (see Figure 6.16), this protocol also contains a field for specifying the user data in bytes [12].

This sort of highlighting of the frame's boundaries entails the risk that the receiver will no longer be able to correctly recognize the end of

Ein weiteres Beispiel ist das allererste und heute nicht mehr gebräuchliche *DIX-Rahmenformat* von Ethernet aus den frühen 1980er Jahren. Die Abkürzung steht stellvertretend für die bei der Entwicklung beteiligten Unternehmen Digital Equipment Corporation, Intel und Xerox. Auch bei diesem Protokoll gibt es, im Gegensatz zum aktuellen Rahmenformat von Ethernet (siehe Abbildung 6.16), ein Datenfeld zur Angabe der Nutzdaten in Bytes [12].

Diese Art der Rahmenmarkierung birgt das Risiko, dass der Empfänger das Ende des Rahmens nicht mehr korrekt erkennen kann, wenn

Figure 6.12: Frame of the obsolete DIX Ethernet Standard

the frame if the field *Count* is modified during the transmission [9].

das Datenfeld *Count* während der Übertragung verändert wird [9].

6.3.2

Byte/Character Stuffing

A protocol that marks the frame boundary with special characters is the byte-oriented protocol *Binary Synchronous Communication* (BISYNC) from the 1960s. Figure 6.13 shows the structure of a BISYNC frame.

Zeichenstopfen

Ein Protokoll, das die Rahmengrenze mit speziellen Zeichen markiert, ist das Byte-orientierte Protokoll *Binary Synchronous Communication* (BISYNC) aus den 1960er Jahren. Abbildung 6.13 zeigt den Aufbau eines BISYNC-Rahmens.

Figure 6.13: Frame of the Protocol BISYNC

Control characters, which in this context are also called *sentinel characters*, mark the structure of the frames. The `SYN` character highlights the beginning of a frame. The beginning of the header marks `SOH` (Start of Header). The payload is located between `STX` (Start of Text) and `ETX` (End of Text). If `ETX` occurs in the payload (body), it must be escaped (*quoted*) by a stuffed `DLE` (Data Link Escape) in the Data Link Layer. If `DLE` appears in the payload, it is also escaped by an additional stuffed `DLE` [9, 13, 17].

The method is called *Byte Stuffing* or *Character Stuffing* because the Data Link Layer of the sender inserts (*stuffs*) extra characters into the payload. The Data Link Layer of the receiver removes the *stuffed* characters from the received payload, before passing it to the Network Layer.

A disadvantage of this method is that it requires the use of ASCII character encoding. More recent protocols of this layer no longer operate in a byte-oriented way, but bit-oriented,

Steuerzeichen, die in diesem Kontext auch *Sentinel-Zeichen* heißen, markieren die Struktur der Rahmen. Den Anfang eines Rahmens markiert das `SYN`-Zeichen. Den Anfang des Headers markiert `SOH` (Start of Header). Die Nutzdaten befinden sich zwischen `STX` (Start of Text) und `ETX` (End of Text). Kommt `ETX` im Nutzdatenteil (Body) vor, muss es in der Sicherungsschicht durch ein `DLE`-Zeichen (Data Link Escape) geschützt (*maskiert*) werden. Kommt `DLE` in den Nutzdaten vor, wird es auch durch ein zusätzliches `DLE` maskiert [9, 13, 17].

Das Verfahren heißt *Zeichenstopfen* (englisch: Character Stuffing), weil die Sicherungsschicht des Senders zusätzliche Zeichen in die Nutzdaten einfügt (*stopft*). Die Sicherungsschicht des Empfängers entfernt *gestopfte* Zeichen aus den empfangenen Nutzdaten, bevor sie an die Vermittlungsschicht übergeben werden.

Ein Nachteil des Verfahrens ist das es die Nutzung der ASCII-Zeichenkodierung voraussetzt. Aktuellere Protokolle dieser Schicht arbeiten nicht mehr Byte-orientiert, sondern Bit-

because this allows using any character encoding.

orientiert, weil damit beliebige Zeichenkodierungen verwendet werden können.

6.3.3

Bit Stuffing

In bit-oriented protocols, each frame has a particular bit pattern as start and end sequence. Examples include the protocol *High-Level Data Link Control* – HDLC (RFC 4349) and the *Point-to-Point Protocol* – PPP (RFC 1661) that is based on it, where each frame begins and ends with the bit sequence `01111110` [17]. The principle is shown in Figure 6.14.

Bitstopfen

Bei Bit-orientierten Protokollen hat jeder Rahmen ein spezielles Bitmuster als Anfangs- und Endsequenz. Beispiele sind das Protokoll *High-Level Data Link Control* – HDLC (RFC 4349) und das darauf aufbauende *Point-to-Point Protocol* – PPP (RFC 1661), bei dem jeder Rahmen mit der Bitfolge `01111110` beginnt und endet [17]. Das Prinzip zeigt Abbildung 6.14.

Figure 6.14: Frame of the Protocol HDLC

With bit stuffing, it is guaranteed that the start and end sequence does not occur in the payload.

If the HDLC protocol in the Data Link Layer of the sender discovers five consecutive one-bits in the bitstream from the Network Layer, it *stuffs* a zero-bit into the bitstream. If the HDLC protocol in the Data Link Layer of the receiver discovers five consecutive one-bits, followed by a zero-bit in the bitstream from the Physical Layer, the *stuffed* zero-bit is removed.

Mit Bitstopfen (englisch: Bit Stuffing) wird garantiert, dass die Anfangs- und Endsequenz nicht in den Nutzdaten vorkommt.

Entdeckt das HDLC-Protokoll in der Sicherungsschicht des Senders fünf aufeinanderfolgende Einsen im Bitstrom von der Vermittlungsschicht, *stopft* es eine Null in den Bitstrom. Erkennt das HDLC-Protokoll in der Sicherungsschicht des Empfängers fünf aufeinanderfolgende Einsen und eine Null im Bitstrom, von der Bitübertragungsschicht, wird das *gestopfte* Null-Bit entfernt.

6.3.4

Line Code Violations

Depending on the line code used in the Physical Layer, invalid signal sequences can delimit frames. One network technology that uses this technique is Token Ring, which uses differential Manchester encoding. With this line code, for clock recovery, a signal level change takes place in each bit cell [17].

In Token Ring, frames begin with a byte (*starting delimiter*) containing four line code vi-

Verstöße gegen Regeln des Leitungscodes

Abhängig vom verwendeten Leitungscode in der Bitübertragungsschicht können ungültige Signalfolgen Rahmen abgrenzen. Eine Netzwerktechnologie, bei der dieses Verfahren angewendet wird, ist Token Ring, das die differentielle Manchesterkodierung verwendet. Bei diesem Leitungscode findet in jeder Bitzelle ein Pegelwechsel zur Taktrückgewinnung statt [17].

Bei Token Ring beginnen Rahmen mit einem Byte (*Starting Delimiter*), das vier Verstöße

olations. First, the line level remains unchanged at signal level 2 for an entire clock cycle and then stays at signal level 1 for an entire clock cycle. These are two line code violations. This is followed by a zero-bit and again the two line code violations. Finally, a sequence of three zero-bits follows (see Figure 6.15).

gegen die Kodierregeln enthält. Zuerst bleibt der Leitungspegel einen ganzen Taktzyklus lang unverändert auf Signalpegel 2 und dann einen ganzen Taktzyklus lang auf Signalpegel 1. Dabei handelt es sich um zwei Verstöße gegen die Kodierregeln. Es folgt der Datenwert Null und erneut die beiden Verstöße gegen die Kodierregeln. Abschließend folgt dreimal der Datenwert Null (siehe Abbildung 6.15).

Figure 6.15: Starting Delimiter of Token Ring

The second last byte (*ending delimiter*) of a Token Ring frame contains the same four line code violations as the *starting delimiter*.

Das vorletzte Byte (*Ending Delimiter*) eines Rahmens bei Token Ring enthält die gleichen vier Verstöße gegen die Kodierregeln wie der *Starting Delimiter*.

6.4

Framing in current Computer Networks

Rahmenformate aktueller Computernetze

Current Data Link Layer protocols like Ethernet and WLAN no longer work in a byte-oriented way, but bit-oriented, because any character encoding can be used in this way.

Aktuelle Protokolle der Sicherungsschicht wie Ethernet und WLAN arbeiten nicht mehr Byte-, sondern Bit-orientiert, weil so beliebige Zeichenkodierungen verwendet werden können.

6.4.1

Ethernet Frames

Rahmen bei Ethernet

The minimum size of an Ethernet frame including the preamble and the Start Frame Delimiter (SFD) is 72 bytes, and the maximum size is 1,526 bytes. Without preamble and SFD, the minimum size is 64 bytes, and the maximum size is 1,518 bytes. The VLAN (see section 10.2) tag increases the maximum size by 4 bytes.

The preamble is the 7 bytes long bit sequence `101010...1010` and it is used in bus networks (topologies) to synchronize the receiver to the clock rate and to identify the beginning of the

Die Mindestgröße eines Ethernet-Rahmens inklusive Präambel und Start Frame Delimiter (SFD) ist 72 Bytes und die maximale Größe 1.526 Bytes. Ohne Präambel und SFD ist die Mindestgröße 64 Bytes und die maximale Größe ist 1.518 Bytes. Mit VLAN-Tag (siehe Abschnitt 10.2) vergrößert sich die maximale Größe um 4 Bytes.

Die Präambel besteht aus der 7 Bytes langen Bitfolge `101010...1010` und dient bei Bus-Topologien dazu, den Empfänger auf die Taktrate zu synchronisieren und den Anfang des Rahmens eindeutig zu kennzeichnen [17]. Dar-

frame. This is followed by the 1 byte long SFD with the bit sequence `10101011`.

A comparison of the values described for the preamble and the SFD with their hexadecimal representations in Figure 6.16 may seem confusing. For instance, the hexadecimal representation of the bit sequence `10101010` is `AA` and not `55`, as shown in Figure 6.16. Also, the value of the SFD does not match the hexadecimal value `D5` at first sight. However, the representation in the figure is correct because Ethernet transmits the *most significant byte* first, but at the same time, Ethernet transmits the *least significant bit* (*LSB*) first within each byte.

For this reason, with Ethernet, each byte of the preamble is not transmitted as bit sequence `10101010`, but as bit sequence `010101` and the transmitted bit sequence of the SFD is `11010101` and not `10101011`.

The fields for the physical addresses (MAC addresses) of sender and destination are 6 bytes long each. The 4 bytes long optional VLAN tag contains, among others, a 12 bits long VLAN ID and a 3 bits long field for prioritization. The field *Type* contains the information what protocol is used in the next upper protocol layer. If IPv4 is used, the field has the value `0x0800`, and for IPv6 the field has the value `0x86DD`. If the payload contains an ARP message, the field has the value `0x0806`. Each frame can contain a maximum of 1,500 bytes of payload. With the *Pad* field, the frame length can be increased to the minimum frame size of 72 bytes if required. This is necessary so that collision detection can function via CSMA/CD. Finally follows a 32 bits long checksum, which does not include the preamble and the SFD (see Figure 6.16).

The *Interframe Spacing* or *Interframe Gap* is the minimum time interval between two transmitted frames on the transmission medium. The minimum idle period is 96 bit times (12 bytes). Thus, the Interframe Spacing for Ethernet at 10 Mbps corresponds to a time interval of 9.6 microseconds. For 100 Mbps it is 0.96 microseconds, and for 1 Gbps it is 96 nanoseconds.

Some network devices allow to reduce the Interframe Spacing period and enable a better

auf folgt der 1 Byte große SFD mit der Bitfolge `10101011`.

Ein Vergleich der hier beschrieben Werte für die Präambel und das SFD mit den hexadezimalen Darstellungen in Abbildung 6.16 mag verwirrend sein. Immerhin ist die hexadezimale Darstellung der Bitfolge `10101010` eigentlich `AA` und nicht `55`, wie es in Abbildung 6.16 dargestellt ist. Auch der Wert des SFD passt auf den ersten Blick nicht mit dem hexadezimalen Wert `D5` überein. Dennoch ist die Darstellung in der Abbildung korrekt, weil die Byte-Reihenfolge bei Ethernet zwar so ist, dass das höchstwertige Byte zuerst übertragen wird (*most-significant byte first*), aber gleichzeitig überträgt Ethernet innerhalb jeden Bytes das geringwertigste Bit (*LSB – least-significant bit*) zuerst.

Aus diesem Grund wird bei Ethernet jedes Byte der Präambel nicht als Bitfolge `10101010`, sondern als Bitfolge `01010101` übertragen und die übertragene Bitfolge des SFD ist nicht `10101011` sondern `11010101`.

Die Datenfelder für die physischen Adressen (MAC-Adressen) von Sender und Ziel sind jeweils 6 Bytes lang. Der 4 Bytes lange optionale VLAN-Tag enthält unter anderem eine 12 Bits lange VLAN-ID und ein 3 Bits großes Feld zur Priorisierung. Das Datenfeld *Typ* enthält das verwendete Protokoll der nächsthöheren Schicht. Bei IPv4 hat es den Wert `0x0800` und bei IPv6 den Wert `0x86DD`. Enthalten die Nutzdaten eine ARP-Nachricht, ist der Wert `0x0806`. Jeder Rahmen kann maximal 1.500 Bytes Nutzdaten enthalten. Mit dem Datenfeld *Pad* werden Rahmen bei Bedarf auf die erforderliche Mindestgröße von 72 Bytes gebracht. Das ist nötig, damit die Kollisionserkennung via CSMA/CD funktioniert. Abschließend folgt eine 32 Bits lange Prüfsumme, die aber nicht die Präambel und den SFD einschließt (siehe Abbildung 6.16).

Das *Interframe Spacing* bzw. *Interframe Gap* ist der minimale zeitliche Abstand zwischen zwei gesendeten Rahmen auf dem Übertragungsmedium. Die minimale Wartezeit entspricht 96 Bitzeiten (12 Bytes). Somit entspricht das Interframe Spacing bei Ethernet bei 10 MBit/s einem zeitlichen Abstand von 9,6 Mikrosekunden. Bei 100 MBit/s sind es 0,96 Mikrosekunden und bei bei 1 GBit/s sind es 96 Nanosekunden.

Einige Netzwerkgeräte erlauben es den Interframe Spacing zu verkürzen und ermöglichen

data rate under optimal conditions. However, this also increases the risk that the boundaries of frames are no longer correctly recognized.

dadurch unter optimalen Bedingungen einen höheren Datendurchsatz. Allerdings erhöht sich dadurch auch das Risiko, das die Grenzen von Rahmen nicht mehr korrekt erkannt werden.

6.4.2

WLAN Frames

While an Ethernet frame without preamble may have a maximum size of 1,518 bytes, the maximum size of a WLAN frame (only the part of the Data Link Layer) is 2,346 bytes (see Figure 6.17).

In the Physical Layer, the protocol adds a preamble to synchronize the receiver including a SFD. Their length and content follow either the standard *Long Preamble Format* or the optional standard *Short Preamble Format*. The latter is not supported by all devices and is only recommended under optimal conditions. Also, the Physical Layer of WLAN defines some small fields at the beginning of each frame. These are...

- *Signal* that indicates the data rate (Mbps),
- *Service* that may contain specific information about the modulation method used,
- *Length* that contains the time in microseconds required to transmit the frame, and
- *CRC* that contains a checksum over the fields Signal, Service and Length.

The field *Frame Control* is 2 bytes long and contains several smaller fields. Among other things, the protocol version is defined here (the field always has the value `0`). The field *Type* indicates what kind of frame it is. Frames are differentiated into management, control and data frames. The group of management frames includes among others the *Beacon Frames* (see section 5.1.3).

For management frames, the field *Type* has the value `00`. For control frames, it has the value `01`, and for data frames, it has the value `10`. The content of the field *Subtype* specifies

Rahmen bei WLAN

Während ein Ethernet-Rahmen ohne Präambel maximal 1.518 Bytes groß sein darf, ist die maximale Größe eines WLAN-Rahmens (nur der Teil der Sicherungsschicht) 2.346 Bytes (siehe Abbildung 6.17).

In der Bitübertragungsschicht fügt das Protokoll eine Präambel zur Synchronisierung des Empfängers inklusive SFD hinzu. Deren Länge und Inhalt entspricht entweder dem üblichen Standard *Long Preamble Format* oder dem optionalen Standard *Short Preamble Format*. Letzterer wird nicht von allen Geräten unterstützt und ist nur unter optimalen Bedingungen empfehlenswert. Zudem sieht die Bitübertragungsschicht von WLAN zu beginn jedes Rahmens noch einige kurze Datenfelder vor. Diese sind...

- *Signal*, das die Datenrate (MBit/s) angibt,
- *Service*, das spezielle Informationen zur verwendeten Modulation enthalten kann,
- *Länge*, das die zur Übertragung des Rahmens benötigte Zeit in Mikrosekunden enthält und
- *CRC*, das eine Prüfsumme über die Felder Signal, Service und Länge enthält.

Die *Rahmensteuerung* ist 2 Bytes lang und enthält mehrere kleinere Datenfelder. Hier ist unter anderem die Protokollversion definiert (das Datenfeld hat immer den Wert `0`). Im Datenfeld *Typ* ist angegeben, um was für eine Art Rahmen es sich handelt. Die Rahmen werden unterschieden in Management-, Kontroll- und Datenrahmen. Ein Beispiel für Management-Rahmen sind die *Beacons* (siehe Abschnitt 5.1.3).

Bei Managementrahmen hat das Datenfeld *Typ* den Wert `00`, bei Kontrollrahmen hat es den Wert `01` und bei Datenrahmen den Wert `10`. Der Inhalt des Datenfelds *Subtyp* definiert

Figure 6.16: Ethernet Frame (IEEE 802.3) including the VLAN Tag (IEEE 802.1Q)

Figure 6.17: WLAN Frame (IEEE 802.11)

Table 6.3: Selection of important Management, Control, and Data Frames [12]

Field *Type*	Field *Subtype*	Name	Description
00	0000	Association Request	Association request to an access point
00	0001	Association Response	Positive response to an association request
00	0010	Reassociation Request	Reassociation request to an access point
00	0011	Reassociation Response	Positive response to a reassociation request
00	0100	Probe Request	Search request for an access point with a specific SSID
00	0101	Probe Response	Response to a search request
00	1000	Beacon (*Leuchtfeuer*)	Periodic announcement of a WLAN cell by its access point
01	1011	Request To Send (RTS)	Reservation request for the medium to the access point
01	1100	Clear To Send (CTS)	Acknowledgment of a reservation request
01	1101	Acknowledgment (ACK)	Acknowledgment of a transmission
10	0000	Data	Data transmission

the content of the frame in a more detailed way. A selection of the essential WLAN frames and the proper values for the fields *Type* and *Subtype* is presented in Table 6.3.

If a packet of the Network Layer does not fit into a single frame, the packet is fragmented, which means it is distributed over several frames. If the field *More Fragments* contains the value 1, then this means that further fragments arrive. With the field *Retry*, the sender can indicate that the frame was sent again. If the field *Power Management* has the value 1, the sender informs that it is currently in power-saving mode. If *More Data* has the value 1, the access point informs a client in power-saving mode that it has buffered additional data for it. The WEP field indicates whether the frame is encrypted by using the WEP method. If the field *Order* has the value 1, this specifies that the frames are transmitted in an ordered sequence.

The 2 bytes long field *Duration* contains a value for updating the counter variable *Network Allocation Vector* (NAV).

The content of the four address fields, which all have a length of 6 bytes, depend on whether it is a WLAN in *infrastructure mode* or *ad-hoc mode* and whether the frame has been sent from a terminal device to an access point, from an access point to a terminal device or via a Distribution System (DS). A DS is a wireless network that consists of multiple access points to allow greater network coverage.

detaillierter den Inhalt des Rahmens. Eine Auswahl der wichtigsten WLAN-Rahmen und den zugehörigen Werten für die Datenfelder *Typ* und *Subtyp* enthält Tabelle 6.3.

Passt ein Paket der Vermittlungsschicht nicht in einen einzelnen Rahmen, wird das Paket fragmentiert, also auf mehrere Rahmen verteilt. Enthält das Datenfeld *More Fragments* den Wert 1, dann heißt das, dass weitere Fragmente eintreffen und mit dem Datenfeld *Retry* kann der Sender angeben, dass der Rahmen erneut gesendet wurde. Hat das Datenfeld *Power Management* den Wert 1, teilt der Sender damit mit, dass er sich momentan im Stromsparmodus befindet. Hat *More Data* den Wert 1, teilt die Basisstation einem Client im Stromsparmodus mit, dass sie noch weitere Daten für ihn gepuffert hat. Im Datenfeld WEP ist angegeben, ob der Rahmen mit dem WEP-Verfahren verschlüsselt ist. Hat das Datenfeld *Order* den Wert 1, bedeutet das, dass die Rahmen in geordneter Reihenfolge übertragen werden.

Das 2 Bytes lange Datenfeld *Dauer* enthält einen Wert für die Aktualisierung der Zählvariable *Network Allocation Vector* (NAV).

Der Inhalt der vier jeweils 6 Bytes großen Adressfelder hängt davon ab, ob es ein WLAN im *Infrastruktur-Modus* oder im *Ad-hoc-Modus* ist und ob der Rahmen von einem Endgerät an eine Basisstation, von einer Basisstation an ein Endgerät oder alternativ über ein Distribution System (DS) gesendet wurde. Ein Distribution System ist ein Funknetzwerk, das aus mehreren Basisstationen besteht, um eine größeren Netzabdeckung zu erreichen.

Table 6.4: Content of the four Address Fields in WLAN Frames

Mode	To DS	From DS	Address field 1	Address field 2	Address field 3	Address field 4
Ad-hoc mode	0	0	DA	SA	BSSID	—
Infrastructure mode (from the access point)	0	1	DA	BSSID	SA	—
Infrastructure mode (to the access point)	1	0	BSSID	SA	DA	—
Infrastructure mode (with distribution system)	1	1	RA	TA	DA	SA

DA = Destination Address
SA = Source Address
RA = Receiver Address
TA = Transmitter Address

Possible values in the address fields are the MAC addresses of the sender (*Source Address*) and the destination (*Destination Address*) as well as in a distribution system of the next station on the route to the destination (*Receiver Address*) and of the last station which forwarded the frame (*Transmitter Address*).

The fields may also contain the *Basic Service Set Identifier* (BSSID). For a WLAN in *infrastructure mode*, the BSSID is the MAC address of the access point, and for a WLAN in *ad-hoc mode*, the BSSID is a 46 bits long random number.

The content of the four address fields in relation to the fields *To DS* and *From DS* (*DS* = Distribution System) is shown in Table 6.4.

The 2 bytes long field *Sequence Control* consists of a 4 bits long fragment number and a 12 bits long sequence number. If a frame has been divided into several fragments, the sequence number is the same for all fragments. The final field contains a 32 bits long checksum for all fields, except the payload.

Mögliche Werte in den Adressfeldern sind die MAC-Adresse der Basisstation, von Sender (*Source Address*) und Empfänger (*Destination Address*) sowie bei einem Distribution System der nächsten Station auf dem Weg zum Ziel (*Receiver Address*) und von der letzten Station, die den Rahmen weitergeleitet hat (*Transmitter Address*).

Zudem können die Felder auch den *Basic Service Set Identifier* (BSSID) enthalten. Bei einem WLAN im *Infrastruktur-Modus* ist der BSSID die MAC-Adresse der Basisstation und bei einem WLAN im *Ad-hoc-Modus* ist die BSSID eine 46 Bits lange Zufallszahl.

Die Belegung der vier Adressfelder in Abhängigkeit von den Datenfeldern *To DS* (*DS* = Distribution System) und *From DS* enthält Tabelle 6.4.

Das 2 Bytes große Datenfeld *Sequenzkontrolle* besteht aus einer 4 Bits langen Fragmentnummer und einer 12 Bits langen Sequenznummer. Wurde ein Rahmen in mehrere Fragmente unterteilt, ist die Sequenznummer für alle Fragmente gleich. Abschließend folgt eine 32 Bits lange Prüfsumme, die aber nicht die Nutzdaten mit einschließt.

6.4.3

WLAN Control Frames

Some control frames have a different structure than the one shown in Figure 6.17. By way of example, Figure 6.18 shows the frame formats for the control characters *Request To Send* (RTS) and *Clear To Send* (CTS), as well as *Acknowledgment* (ACK), with which the receiver

Spezielle Rahmen bei WLAN

Einige Kontrollrahmen haben einen anderen Aufbau, als den in Abbildung 6.17 dargestellten. Exemplarisch zeigt Abbildung 6.18 die Rahmenformate für die Steuerzeichen *Request To Send* (RTS) und *Clear To Send* (CTS) sowie für *Acknowledgment* (ACK), mit dem der Empfän-

confirms the successful transmission of a frame at the sender.

ger die erfolgreiche Übertragung eines Rahmens beim Sender bestätigt.

Figure 6.18: Special Frames of WLAN

The RTS frame has a length of 20 bytes. With this frame, a station can send a reservation request for the transmission medium to the access point. The first address field contains the MAC address of the access point, the second address field contains the MAC address of the station which sent the request.

With a CTS frame that has a length of 14 bytes, an access point confirms the reservation request for the transmission medium. The address field contains the MAC address of the station, which sent the reservation request.

The ACK frame is also 14 bytes long, and the address field contains the MAC address of the station, which successfully transmitted the frame.

Der RTS-Rahmen hat eine Länge von 20 Bytes. Mit ihm kann ein Sender eine Reservierungsanfrage für das Übertragungsmedium an die Basisstation senden. Die MAC-Adresse der Basisstation befindet sich im ersten Adressfeld und die MAC-Adresse des anfragenden Endgeräts im zweiten Adressfeld.

Mit einem CTS-Rahmen, der eine Länge von 14 Bytes hat, bestätigt eine Basisstation die Reservierungsanfrage für das Übertragungsmedium. Das Adressfeld enthält die MAC-Adresse des Endgeräts, das die Reservierungsanfrage gestellt hatte.

Der ACK-Rahmen ist ebenfalls 14 Bytes lang und im Adressfeld befindet sich die MAC-Adresse des Senders, also der Station, die den Rahmen erfolgreich übertragen hat.

6.5

Maximum Transmission Unit (MTU)

The MTU is the maximum payload of a frame on the Data Link Layer. From the perspective of the Network Layer, the MTU determines the maximum packet length. The MTU of Ethernet is usually 1,500 bytes, but for PPP over Ethernet

Maximum Transmission Unit (MTU)

Die MTU ist die maximale Menge an Nutzdaten eines Rahmens auf der Sicherungsschicht. Aus Sicht der Vermittlungsschicht legt die MTU die maximale Paketlänge fest. Bei Ethernet ist die MTU meist 1.500 Bytes und bei PPP over Ether-

(PPPoE) it is 1,492 bytes, because 8 bytes are needed in every Ethernet frame for additional connection information. The use of additional protocols by some Internet service providers may further reduce the MTU. Table 6.5 contains a selection of typical MTU sizes of popular network technologies.

net (PPPoE) ist sie 1.492 Bytes weil 8 Bytes in jedem Ethernet-Rahmen für zusätzliche Verbindungsinformationen benötigt werden. Durch den Einsatz weiterer Protokolle bei einigen Internetanbietern kann sich die MTU weiter verringern. Tabelle 6.5 enthält eine Auswahl typischer MTU-Größen bekannter Vernetzungstechnologien.

Table 6.5: Typical MTU Sizes of popular Network Technologies

Network technology	MTU
Ethernet	usually 1,500 bytes
FDDI	4,352 bytes
ISDN	576 bytes
PPPoE	usually 1,492 bytes
Token Ring (4 Mbps)	4,464 bytes
Token Ring (16 Mbps)	17,914 bytes
WLAN	2,312 bytes

6.6

Error-detection Codes

For error detection, a checksum is attached to each frame by the sender. This way, frames with errors are detected and thrown away by the receiver. The Data Link Layer does not provide a functionality that allows the receiver to request previously discarded frames at the sender. Possible ways to detect errors are the *two-dimensional parity-check code* and the *polynomial code* (*Cyclic Redundancy Check*, CRC).

Fehlererkennung

Zur Fehlererkennung wird jedem Rahmen vom Sender eine Prüfsumme angefügt. So werden fehlerhafte Rahmen vom Empfänger erkannt und verworfen. Ein erneutes Anfordern verworfener Rahmen sieht die Sicherungsschicht nicht vor. Möglichkeiten der Fehlererkennung sind die *zweidimensionale Parität* und die *zyklische Redundanzprüfung*.

6.6.1

Two-dimensional Parity-check Code

This method is well-suited for error-detection, especially when the 7-bit US-ASCII character encoding is used (see section 2.4.1). For each 7-bit section, an additional *parity bit* is calculated and attached to balance out the number of one-bits in the byte.

If the protocol defines even parity, the parity bit is used to obtain an even number of one-bits in every byte. If odd parity is desired, the parity bit is used to obtain an odd number of one-bits in every byte. This method is called a *one-dimensional parity-check code*.

Zweidimensionale Parität

Dieses Verfahren eignet sich dann besonders gut, wenn die 7-Bit-Zeichenkodierung US-ASCII (siehe Abschnitt 2.4.1) verwendet wird. Zu jedem 7-Bit-Abschnitt wird ein zusätzliches *Paritätsbit* berechnet und angehängt, um die Anzahl der Einsen im Byte auszugleichen.

Definiert das Protokoll die gerade Parität, erhält das Paritätsbit bei Bedarf den Wert Eins oder Null, um eine gerade Anzahl von Einsen im Byte zu erwirken. Ist ungerade Parität gewünscht, erhält das Paritätsbit bei Bedarf den Wert Eins oder Null, um eine ungerade Anzahl

In addition to the parity bits in each byte, an additional *parity byte* exists for each frame. The content of the parity byte is calculated over each byte of the frame. This method is called *two-dimensional parity-check code*. It can detect all 1-, 2- and 3-bit errors as well as most 4-bit errors [9].

This method is used by BISYNC when transmitting data that is encoded with the 7-bit character encoding US-ASCII.

von Einsen im Byte zu erwirken. Das ist die *eindimensionale Parität*.

Neben den Paritätsbits in jedem Byte existiert für jeden Rahmen noch ein zusätzliches *Paritätsbyte*. Dessen Inhalt ist das Ergebnis der Paritätsberechnung quer über jedes Byte des Rahmens. Das ist die *zweidimensionale Parität*. Zweidimensionale Parität kann alle 1-, 2-, 3-Bit-Fehler und die meisten 4-Bit-Fehler erkennen [9].

Das Verfahren verwendet wird unter anderem von BISYNC bei der Übertragung von Daten verwendet, die mit der 7-Bit-Zeichenkodierung US-ASCII kodiert sind.

6.6.2

Cyclic Redundancy Check

The Cyclic Redundancy Check (CRC) makes use of the fact that bit sequences can be written as polynomials with the coefficients 0 and 1. A frame with k bits is considered as a polynomial of degree $k-1$. The most significant bit is the coefficient of x^{k-1}, the next bit is the coefficient of x^{k-2}, and so on. The bit sequence `10011010` thus corresponds to the following polynomial [9]:

Zyklische Redundanzprüfung

Die zyklische Redundanzprüfung – Cyclic Redundancy Check (CRC) basiert darauf, dass man Bitfolgen als Polynome mit den Koeffizienten 0 und 1 schreiben kann. Ein Rahmen mit k Bits wird als Polynom vom Grad $k-1$ betrachtet. Das werthöchste Bit ist der Koeffizient von x^{k-1}, das nächste Bit ist der Koeffizient für x^{k-2}, usw. Die Bitfolge `10011010` entspricht also dem Polynom [9]:

$$\begin{aligned} M(x) &= 1 \times x^7 + 0 \times x^6 + 0 \times x^5 + 1 \times x^4 + 1 \times x^3 + 0 \times x^2 + 1 \times x^1 + 0 \times x^0 \\ &= x^7 + x^4 + x^3 + x^1 \end{aligned}$$

The sending and receiving of messages can be imagined as an exchange of polynomials.

To perform the cyclic redundancy check, sender and receiver have to use a *generator polynomial* $C(x)$, which is a polynomial of degree n. For example: Be $C(x) = x^3 + x^2 + x^0 = 1101$, then $n = 3$, so the polynomial is of degree 3. The degree of the generator polynomial corresponds to the number of bits minus one.

The generator polynomial is determined by the Data Link Layer protocol used and the choice of the generator polynomial determines which error types are detected. Table 6.6 contains some generator polynomials which are often used in practice.

Das Senden und Empfangen von Nachrichten kann man sich als Austausch von Polynomen vorstellen.

Um die zyklische Redundanzprüfung durchführen zu können, müssen Sender und Empfänger ein *Generatorpolynom* $C(x)$ verwenden, das ein Polynom vom Grad n ist. Sei zum Beispiel $C(x) = x^3 + x^2 + x^0 = 1101$, dann ist $n = 3$, also ist das Polynom vom Grad 3. Der Grad des Generatorpolynoms entspricht der Anzahl der Bits minus eins.

Das Generatorpolynom wird vom verwendeten Protokoll der Sicherungsschicht festgelegt und diese Auswahl legt fest, welche Fehlerarten erkannt werden. Eine Auswahl häufig verwendeter Generatorpolynome enthält Tabelle 6.6.

Table 6.6: Selection of popular Generator Polynomials

Name	Application	Generator polynomial $C(x)$
CRC-5	USB	$x^5 + x^2 + x^0$
CRC-8	ISDN	$x^8 + x^2 + x^1 + x^0$
CRC-16	BISYNC	$x^{16} + x^{15} + x^2 + x^0$
CRC-32	Ethernet	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+x^0$

The following example uses the generator polynomial CRC-5. If the CRC checksum has to be calculated for a frame, n zero-bits are appended to the frame. n corresponds to the degree of the generator polynomial.

The generator polynomial consists of six digits, so five zero-bits are appended. The frame with the appended zero-bits is divided from the left via exclusive or `XOR` (`1 XOR 1 = 0`, `1 XOR 0 = 1`, `0 XOR 1 = 1`, `0 XOR 0 = 0`) by the generator polynomial. This procedure always starts with the first column where dividend and divisor both have a bit with value `1`.

Das folgende Beispiel verwendet das Generatorpolynom CRC-5. Soll für einen Rahmen die CRC-Prüfsumme berechnet werden, werden n Nullen an den Rahmen angehängt. n entspricht dem Grad des Generatorpolynoms.

Das Generatorpolynom hat sechs Stellen, also werden fünf Nullen ergänzt. Der Rahmen mit Anhang wird von links mit exklusivem Oder `XOR` (`1 XOR 1 = 0`, `1 XOR 0 = 1`, `0 XOR 1 = 1`, `0 XOR 0 = 0`) durch das Generatorpolynom dividiert. Dabei fängt man immer mit der ersten Spalte an, in der Dividend und Divisor beide den Wert `1` haben.

Generator polynomial:	`100110`
Frame (payload):	`10101`
Frame with appended zero-bits:	`1010100000`
Remainder:	`10100`
Frame including the checksum	`1010110100`

The sender calculates the checksum.

```
    1010100000:100110
    100110
XOR ------
      110000
      100110
  XOR ------
       101100
       100110
   XOR ------
          10100 = checksum
```

The receiver verifies the transmission.

```
    1010110100:100110
    100110
XOR ------
      110101
      100110
  XOR ------
       100110
       100110
   XOR ------
            0000 => error-free
```

Figure 6.19: Error Detection with the Cyclic Redundancy Check Method

The remainder (`10100`) is the *checksum*, which is appended to the payload. The remainder must have a length of n bits, where n is still the degree of the generator polynomial. The transferred frame with the checksum is therefore `1010110100`.

Der Rest (`10100`) ist die *Prüfsumme*, die an die Nutzdaten angehängt wird. Der Rest muss aus n Bits bestehen, wobei n nach wie vor der Grad des Generatorpolynoms ist. Der übertragene Rahmen mit Prüfsumme ist also `1010110100`.

The receiver of the frame can verify if the frame has arrived error-free. Using a division via XOR by the generator polynomial, faulty transmissions are detected. If the remainder of the division has value zero, the transmission has been error-free.

If one bit in the transmitted frame is defective, the remainder of the division with XOR by the generator polynomial does not have value zero.

This method identifies all one-bit errors, every odd number of defective bits, and all burst errors of length n, where n is the degree of the generator polynomial [6, 19].

Der Empfänger des Rahmens kann überprüfen, ob dieser korrekt angekommen ist. Mit einer Division via XOR durch das Generatorpolynom werden fehlerhafte Übertragungen erkannt. Ist der Rest der Division gleich null, war die Übertragung fehlerfrei.

Ist im übertragenen Rahmen ein Bit fehlerhaft, ist der Rest der Division mit XOR durch das Generatorpolynom ungleich null.

Das Verfahren erkennt neben allen Einbitfehlern auch jede ungerade Anzahl fehlerhafter Bits und alle Bündelfehler der Länge n, wobei n der Grad des Generatorpolynoms ist [6, 19].

6.7

Error-correction Codes

Error correction methods are sometimes used in data transmissions when data is transmitted over just via a single channel, and the transmission mode is simplex. In such a scenario, it is impossible for the receiver to request retransmissions [17]. A typical application example is data transmissions from space probes, because numerous transmission errors can be expected due to the large distances and the given signal interferences.

Apart from such unusual applications, error correction in network protocols is rare in practice. The popular network protocols of the Data Link Layer such as Ethernet, WLAN or Bluetooth only implement error detection with cyclic redundancy checks. This is motivated by the fact that error detection is more efficient than error correction, because lesser parity bits need to be transmitted when error detection is used.

Fehlerkorrektur

Verfahren zur Fehlerkorrektur kommen bei Datenübertragungen bisweilen dort zum Einsatz, wo Daten nur über einen Kanal übertragen werden und die Übertragungsart Simplex ist. In einem solchen Fall kann der Empfänger keine Sendewiederholungen anfordern [17]. Ein typisches Anwendungsbeispiel sind Datenübertragungen von Raumsonden, da bei den großen Distanzen und den gegebenen Beeinflussungen mit zahlreichen Übertragungsfehlern zu rechnen ist.

Abgesehen von solch ungewöhnlichen Anwendungen kommt Fehlerkorrektur in Netzwerkprotokollen in der Praxis selten vor. Bekannte Netzwerkprotokolle der Sicherungsschicht wie Ethernet, WLAN oder Bluetooth ermöglichen ausschließlich Fehlererkennung mit zyklischer Redundanzprüfung. Der Grund dafür ist, dass Fehlererkennung effizienter ist als Fehlerkorrektur, da bei Fehlererkennung weniger Prüfbits übertragen werden müssen.

6.8

Media Access Control Methods

With Ethernet 10BASE2/5 and WLAN, network devices or stations use a shared transmission medium. To coordinate media access and prevent collisions, or at least to avoid them,

Medienzugriffsverfahren

Bei Ethernet 10BASE2/5 und WLAN verwenden die Netzwerkgeräte bzw. Stationen ein gemeinsames Übertragungsmedium. Um den Medienzugriff zu koordinieren und Kollisionen zu

Ethernet implements the media access control method CSMA/CD. WLAN uses CSMA/CA.

Bluetooth is not discussed here, because Bluetooth devices are organized in *piconets* (see section 5.1.4). A master coordinates media access in each piconet.

vermeiden, oder zumindest zu minimieren, verwendet Ethernet das Medienzugriffsverfahren CSMA/CD und WLAN verwendet CSMA/CA.

Bluetooth wird an dieser Stelle nicht behandelt, da sich Bluetooth-Geräte in *Piconetzen* (siehe Abschnitt 5.1.4) organisieren. In jedem Piconetz koordiniert ein Master den Medienzugriff.

6.8.1

Media Access Control Method of Ethernet

Unlike Token Ring, the waiting time and transferable amount of data cannot be precisely predicted in an Ethernet. Regarding the media access, all participants are in direct competition to each other. Waiting time and the amount of data depend on the number of participants and the amount of data each participant sends. Ethernet uses the media access control method *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD).

Carrier Sense (CS) means that each network device monitors the channel (the line) before sending and only transmits frames when the channel is idle. Network devices can, therefore, distinguish between an idle and a busy line. *Multiple Access* (MA) means that all network devices access the same transmission medium in a competitive way. *Collision Detection* (CD) means that every network device monitors the channel even during transmission to detect any collisions that occur and to perform error handling, if necessary.

If a network device wants to transmit frames via Ethernet, it operates according to the following steps in the following sequence (see Figure 6.20): [15, 23, 28]

1. Monitor the transmission medium:
   - Medium is idle $\Longrightarrow$ step 2
   - Medium is busy $\Longrightarrow$ step 3
2. Start the transmission and continue to monitor the transmission medium:
   - If the transmission was successful, this is reported to upper protocol layers $\Longrightarrow$ step 5

Medienzugriffsverfahren bei Ethernet

Anders als bei Token Ring sind bei Ethernet die Wartezeit und übertragbare Datenmenge nicht eindeutig vorhersagbar. Alle Teilnehmer stehen in Bezug auf den Medienzugriff im direktem Wettbewerb. Wartezeit und Datenmenge hängen von der Anzahl der Teilnehmer und der Datenmenge ab, die die einzelnen Teilnehmer versenden. Ethernet verwendet das Medienzugriffsverfahren *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD).

Carrier Sense (CS) heißt, dass jedes Netzwerkgerät vor dem Senden den Kanal (also die Leitung) abhört und nur dann sendet, wenn der Kanal frei ist. Die Netzwerkgeräte können also zwischen einer freien und einer besetzten Verbindungsleitung unterscheiden. *Multiple Access* (MA) heißt, dass alle Netzwerkgeräte auf dasselbe Übertragungsmedium konkurrierend zugreifen. *Collision Detection* (CD) sagt aus, dass jedes Netzwerkgerät auch während des Sendens den Kanal abhört, um auftretende Kollisionen zu entdecken und wenn nötig eine Fehlerbehandlung durchzuführen.

Will ein Netzwerkgerät via Ethernet Datenrahmen übertragen, hält es den folgenden Ablauf (siehe Abbildung 6.20) ein: [15, 23, 28]

1. Übertragungsmedium überwachen:
   - Freies Medium $\Longrightarrow$ Schritt 2
   - Belegtes Medium $\Longrightarrow$ Schritt 3
2. Rahmen senden und das Übertragungsmedium weiter abhören:
   - War die Übertragung erfolgreich, wird dieses an höhere Protokollebenen gemeldet $\Longrightarrow$ Schritt 5

- If a collision has been detected, the transmission process is aborted and the 48 bits long *jam signal* is sent to announce the collision ⟹ step 3

3. The transmission medium is busy. Check the number of transmission attempts:
   - If the maximum number is not reached yet, a random waiting time (*backoff time*) is calculated and awaited ⟹ step 1
   - If the maximum number is reached ⟹ step 4
4. The maximum number of transmission attempts has been reached. The error is reported to upper network layers ⟹ step 5
5. Leave the transmission mode.

To calculate the backoff time, a random value between the minimum and the maximum value of the *Contention Window* (CW) is selected, and this random value is multiplied by the *Slot Time*. The minimum and maximum value of the CW and the Slot Time depend on the modulation method used. These values are fixed (see Table 6.7).

Collisions must be detected by the sender. It is essential that the transmission of a frame has not yet finished when a collision occurs. Otherwise, the network device might already be finished with sending the frame, assuming that the transmission has been successful. Each frame must have a certain minimum length, dimensioned such that the transfer time for a frame of minimum length is not lower than the maximum *Round Trip Time* (RTT).

The RTT is the time it takes for a frame to be transmitted from sender to destination and back. In this case, the maximum RTT is critical, i.e., the time it takes for a frame to travel from one end of the network to the most distant end of the network and back.

- Wurde eine Kollision entdeckt, wird der Sendevorgang abgebrochen und das 48 Bits lange Störsignal (*Jam-Signal*) gesendet, um die Kollision bekannt zu geben ⟹ Schritt 3

3. Übertragungsmedium ist belegt. Anzahl der Übertragungsversuche prüfen:
   - Ist das Maximum noch nicht erreicht, wird eine zufällige Wartezeit (*Backoffzeit*) berechnet und abgewartet ⟹ Schritt 1
   - Ist das Maximum erreicht ⟹ Schritt 4
4. Die maximale Anzahl der Übertragungsversuche ist erreicht. Der Fehler wird an höhere Netzwerkschichten gemeldet ⟹ Schritt 5
5. Übertragungsmodus verlassen.

Zur Berechnung der Backoffzeit wird ein zufälliger Wert zwischen dem minimalem und maximalem Wert des *Contention Window* (CW) bestimmt, und dieser zufällige Wert wird mit der *Slot Time* multipliziert. Der minimale und maximale Wert des CW sowie die Slot Time hängen vom verwendeten Modulationsverfahren ab und sind fest vorgegeben (siehe Tabelle 6.7).

Eine Kollision muss vom Sender erkannt werden. Es ist wichtig, dass ein Rahmen noch nicht fertig gesendet ist, wenn es zur Kollision kommt. Ansonsten ist das sendende Netzwerkgerät vielleicht schon mit dem Aussenden des Rahmens fertig und glaubt an eine erfolgreiche Übertragung. Jeder Rahmen muss eine gewisse Mindestlänge habe, die so dimensioniert ist, dass die Übertragungsdauer für einen Rahmen minimaler Länge die maximale *Round-Trip-Time* (RTT) nicht unterschreitet.

Die RTT, die auch Rundlaufzeit heißt, ist die Zeit, die ein Rahmen benötigt, um von der Quelle zum Ziel und zurück übertragen zu werden. In diesem Fall ist die maximale RTT entscheidend, also die Zeit, die ein Rahmen benötigt, um vom einen Ende des Netzes zum weitest entfernten anderen Ende des Netzes und wieder zurück zu gelangen.

Figure 6.20: Functioning of the Media Access Control Method CSMA/CD of Ethernet

The minimum frame length ensures that a collision reaches the sender before the sender has finished the transmission. If a sender detects a collision, it knows that its frame has not arrived correctly at the receiver and, so it can try again later.

Ethernet specifies a maximum network size and a minimum frame length. The *minimum frame length*, where collision detection is still possible, is calculated as follows:

Dank der Mindestlänge ist garantiert, dass eine Kollision noch den Sender erreicht, bevor dieser mit dem Senden fertig ist. Erkennt der Sender eine Kollision, weiß er, dass sein Rahmen nicht richtig beim Empfänger angekommen ist, und kann es später erneut versuchen.

Für Ethernet ist eine maximal zulässige Netzwerkausdehnung und eine minimale Rahmenlänge festgelegt. Die *minimale Rahmenlänge*, bei der die Kollisionserkennung noch möglich ist, wird folgendermaßen berechnet:

$$P = 2 \times U \times \frac{D}{V}$$

P is the minimum frame length in bits, U is the data rate of the transmission medium in bits per second (bps), D is the network length in meters and V is the signal speed on the transmission medium in meters per second.

The following example calculates the minimum frame length for Thick Ethernet (10BASE5) with 10 Mbit/s and coaxial cables as a transmission medium:

- $U = 10\,\text{Mbps} = 10,000,000\,\text{bps}$
- $D = 2,500\,\text{m}$ (this is the maximum length for 10BASE5)
- $V =$ speed of light $\times$ velocity factor

The speed of light is $299,792,458\,\text{m/s}$. The velocity factor, which is also called wave propagation speed, depends on the transmission medium. It describes the signal propagation speed inside a transmission medium in relation to the speed of light, and it is 1 for vacuum, 0.64 for twisted pair cables Cat-5e, 0.66 for coaxial cables RG-58 that are used for Ethernet 10BASE2, 0.67 for optical fiber and 0.77 for coaxial cables RG-8 that are used for Ethernet 10BASE5 [29].

P ist die minimale Rahmenlänge in Bits, U ist die Datenübertragungsgeschwindigkeit des Übertragungsmediums in Bits pro Sekunde, D ist die Länge des Netzes in Metern und V ist die Signalgeschwindigkeit auf dem Übertragungsmedium in Metern pro Sekunde.

Das folgende Rechenbeispiel berechnet die minimale Rahmenlänge für Thick Ethernet (10BASE5) mit 10 Mbit/s und Koaxialkabeln als Übertragungsmedium:

- $U = 10\,\text{Mbit/s} = 10.000.000\,\text{Bits/s}$
- $D = 2.500\,\text{m}$ (das ist die Maximallänge für 10BASE5)
- $V =$ Lichtgeschwindigkeit $\times$ Ausbreitungsfaktor

Die Lichtgeschwindigkeit ist $299.792.458\,\text{m/s}$ und der Ausbreitungsfaktor, der auch Verkürzungsfaktor genannt wird, hängt vom Übertragungsmedium ab. Er beschreibt die Signalgeschwindigkeit in einem Übertragungsmedium in Relation zur Lichtgeschwindigkeit und ist 1 für Vakuum, 0,64 für Twisted-Pair-Kabel Cat-5e, 0,66 für Koaxialkabel RG-58 wie bei Ethernet 10BASE2, 0,67 für Glasfaser und 0,77 für Koaxialkabel RG-8 wie bei Ethernet 10BASE5 [29].

$$V = 299,792,458\,\text{m/s} \times 0.77 \approx 231,000,000\,\text{m/s}$$
$$P = 2 \times 10 \times 10^6\,\text{bps} \times \frac{2,500\,\text{m}}{231 \times 10^6\,\text{m/s}} \approx 217\,\text{bits} \approx 28\,\text{bytes}$$

The minimum frame length of 64 bytes for Ethernet is, therefore, more than enough.

The *maximum network size* between two network devices, where collision detection is still possible, is calculated as follows:

$$2 \times S_{\text{max}} = V \times t_{\text{frame}}$$

S_{max} is the maximum network size with collision detection, V is the signal speed inside the transmission medium in meters per second

Die minimale Rahmenlänge von 64 Bytes bei Ethernet ist also mehr als ausreichend.

Die *maximale Ausdehnung* zwischen zwei Netzwerkgeräten, bei der die Kollisionserkennung noch funktioniert, wird folgendermaßen berechnet:

$$2 \times S_{\text{max}} = V \times t_{\text{Rahmen}}$$

S_{max} ist die maximale Ausdehnung mit Kollisionserkennung, V ist die Signalgeschwindigkeit auf dem Übertragungsmedium in Metern pro

and t_{frame} is the transmission time of a frame in seconds.

The following example calculates the maximum network size for Thick Ethernet (10BASE5) with 10 Mbit/s and coaxial cables as a transmission medium:

Sekunde und t_{Rahmen} ist die Übertragungsdauer eines Rahmens in Sekunden.

Das folgende Rechenbeispiel berechnet die maximale Ausdehnung wieder für Thick Ethernet (10BASE5) mit 10 Mbit/s und Koaxialkabel als Übertragungsmedium:

$$V = 231,000,000\,\text{m/s} = 231 \times 10^6\,\text{m/s}$$

The transmission time t_{frame} is the transmission time for a single bit multiplied by the number of bits in a frame ($\Longrightarrow$ 512 bits = 64 byte). The transmission time for a single bit at 10 Mbps is 0.1 microseconds. Thus, a frame with a minimum permitted frame length of 64 Bytes needs 51.2 μs to be sent completely. A frame with a duration of 51.2 μs travels the following distance in a coaxial cable:

Die Übertragungsdauer t_{Rahmen} ist die Übertragungsdauer für ein Bit multipliziert mit der Anzahl der Bits in einem Rahmen ($\Longrightarrow$ 512 Bits = 64 Byte). Die Übertragungsdauer für ein Bit bei 10 Mbit/s ist 0,1 Mikrosekunden. Ein Rahmen mit der kleinsten erlaubten Rahmenlänge vom 64 Byte benötigt somit 51,2 μs, um vollständig gesendet zu werden. Ein 51,2 μs langer Rahmen legt im Koaxialkabel folgende Strecke zurück:

$$231 \times 10^6\,\text{m/s} \times 51.2 \times 10^{-6}\,\text{s} = 11{,}827.20\,\text{m} = 11.83\,\text{km}$$

With a maximum permitted network size of 2.5 km, collision detection is therefore possible.

The CSMA/CD media access control method is only mandatory for Ethernet networks with the physical bus topology because there, all network devices are directly connected to a shared transmission medium. Almost all Ethernet-based networks today are *fully switched networks* and therefore free of collisions.

Bei einer maximal erlaubten Ausdehnung von 2,5 km ist Kollisionserkennung somit möglich.

Das Medienzugriffsverfahren CSMA/CD ist nur bei Ethernet-Netzen mit der physischen Bus-Topologie zwingend nötig, weil dort alle Netzwerkgeräte direkt mit einem gemeinsamen Medium verbunden sind. Fast alle auf Ethernet basierende Netze sind heute *vollständig geswitcht* und darum frei von Kollisionen.

6.8.2

Media Access Control Method of WLAN

CSMA/CD cannot be used with wireless networks, because in contrast to cable-based Ethernet, not all collisions can be detected in wireless networks like WLAN.

With CSMA/CD, the sender detects collisions that occur. In wired networks with a shared transmission medium, each node receives the transmissions of all other nodes, and thus each node receives every collision. This is not always the case in wireless networks such as WLAN. For this reason, the media access control method *Carrier Sense Multiple Access/Collision Avoi-*

Medienzugriffsverfahren bei WLAN

CSMA/CD versagt bei Funknetzen, denn anders als beim kabelgebundenen Ethernet kann man bei Funknetzen wie WLAN Kollisionen nicht immer erkennen.

Bei CSMA/CD stellt der Sender auftretende Kollisionen fest. Bei kabelgebundenen Netzen mit gemeinsamem Übertragungsmedium empfängt jeder Teilnehmer die Übertragungen aller anderer Teilnehmer und damit bekommt auch jeder Teilnehmer jede Kollision mit. Bei Funknetzen wie WLAN ist das nicht immer der Fall. Aus diesem Grund will man das Entstehen von Kollisionen mit dem Medienzugriffsverfahren

dance (CSMA/CA) is used, which tries to minimize the occurrence of collisions.

In wireless networks, particular characteristics of the transmission medium result in collisions that may not be detectable by the receiver. These characteristics are the Hidden terminal problem as well as Fading.

In the *Hidden Terminal* problem, invisible or hidden stations are the reason for collisions. Figure 6.21 shows an example. Here, two stations X and Y send data to the access point. Because of obstacles, the stations cannot receive each other's transmissions, although they interfere at the access point.

Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) minimieren.

Spezielle Eigenschaften des Übertragungsmediums führen bei Funknetzen zu unerkannten Kollisionen beim Empfänger. Dabei handelt es sich um Hidden-Terminal und um Fading.

Bei *Hidden-Terminal* sind unsichtbare bzw. versteckte Stationen der Grund für Kollisionen. Abbildung 6.21 zeigt beispielhaft, wie die Stationen X und Y Daten an die Basisstation (Access Point) senden. Wegen Hindernissen können die Stationen ihre Übertragungen gegenseitig nicht erkennen, obwohl sie an der Basisstation interferieren.

Figure 6.21: Hidden Terminal (invisible or hidden Stations)

Fading means a decrease in signal strength. The electromagnetic waves of the wireless network are gradually weakened (attenuated) by obstacles and in free space. Figure 6.22 shows an example of how the two stations X and Y send data to the access point. Due to the positions of the stations to each other, their signals are so weak that the stations cannot detect each other's transmissions. In this scenario, too, the transmissions interfere with each other.

Fading bedeutet abnehmende Signalstärke. Die elektromagnetischen Wellen des Funknetzes werden durch Hindernisse und im freien Raum allmählich abgeschwächt (gedämpft). Abbildung 6.22 zeigt beispielhaft, wie die Stationen X und Y Daten an die Basisstation senden. Durch die Positionen der Stationen zueinander sind deren Signale zu schwach, als dass sie ihre Übertragungen gegenseitig wahrnehmen können. Auch in diesem Szenario stören sich die Übertragungen gegenseitig.

Figure 6.22: Fading (decreasing Signal Strength)

These features make multiple access in wireless networks more complex than in cable-based networks [23]. WLAN (802.11) implements the three media access control methods *CSMA/CA*, *CSMA/CA RTS/CTS* and *CSMA/CA PCF*.

Diese Eigenschaften machen den Mehrfachzugriff bei Funknetzen komplexer, als das bei Netzen der Fall ist [23]. WLAN (802.11) kennt die drei Medienzugriffsverfahren *CSMA/CA*, *CSMA/CA RTS/CTS* und *CSMA/CA PCF*.

CSMA/CA

If a sending station detects a collision via CSMA/CD (Ethernet), it stops sending the frame. WLAN does not use collision detection, but collision avoidance (CSMA/CA). Actually, it is only a method for collision minimization. If a station has started to send a frame, it transmits the whole frame in any case. Once a station has started transmitting, there is no turning back. The sender must, therefore, be able to detect if a frame has not arrived correctly at the receiver. The condition is realized by acknowledgments (ACK) with which the receiver confirms the correct reception of the frame.

CSMA/CA

Erkennt bei CSMA/CD (Ethernet) ein sendender Teilnehmer eine Kollision, bricht er das Senden des Rahmens ab. WLAN verwendet aber keine Kollisionserkennung, sondern mit CSMA/-CA eine Kollisionsvermeidung – eigentlich ist es nur eine Kollisionsminimierung. Hat eine Station mit dem Senden eines Rahmens begonnen, überträgt sie den vollständigen Rahmen in jedem Fall. Es gibt also kein Zurück mehr, wenn eine Station einmal mit dem Senden begonnen hat. Der Sender muss darum erkennen können, wenn ein Rahmen nicht korrekt beim Empfänger angekommen ist. Realisiert wird die Bedingung durch Bestätigungen (ACK) mit denen der Empfänger den korrekten Empfang des Rahmens bestätigt.

Figure 6.23: Media Access Control Method CSMA/CA

First, the sender *listens* to the transmission medium (⟹ Carrier Sense). The medium must be idle for the duration of a *Distributed Interframe Spacing* (DIFS). If the medium is idle for the duration of one DIFS, the sender can transmit a frame (see Figure 6.23).

If a station receives a frame that passes the CRC check, it waits for a *Short Interframe Spacing* (SIFS) time period. Then the receiver sends an *acknowledgment frame* (ACK). The acknowledgment by ACK is not sent via broadcast. DIFS and SIFS guarantee a minimum spacing for CSMA/CA between consecutive frames.

Zuerst *horcht* der Sender am Übertragungsmedium (⟹ Carrier Sense). Das Medium muss für die Dauer eines *Distributed Interframe Spacing* (DIFS) frei sein. Ist das Medium für die Dauer des DIFS frei, kann der Sender einen Rahmen aussenden (siehe Abbildung 6.23).

Empfängt eine Station einen Rahmen, der die CRC-Prüfung besteht, wartet sie einen *Short Interframe Spacing* (SIFS) ab. Danach sendet der Empfänger einen *Bestätigungsrahmen* (ACK). Die Empfangsbestätigung durch ACK erfolgt nicht bei einem Broadcast. DIFS und SIFS garantieren bei CSMA/CA einen Mindestabstand zwischen aufeinanderfolgenden Rahmen.

Further stations that want to send frames wait after the ACK of the receiver for the period of a DIFS with free transmission medium and a *backoff time*. This waiting time is calculated by each station, independently from other stations. The frame is sent after the backoff time has elapsed. If another station occupies the transmission medium during the backoff time, the counter is stopped until the medium is idle again for at least one DIFS [26]. The backoff time is calculated by selecting a random value between the minimum and maximum values of the *Contention Window* (CW) and by multiplying this random value by the *Slot Time*.

The minimum and maximum values of the CW and the Slot Time depend on the modulation method used. The values are fixed (see Table 6.7). The lower and upper limits of the CW are always powers of two, whereby the value 1 is subtracted from the result. If, for example, a WLAN uses the modulation method DSSS, at the first transmission attempt is $31 \leq \text{CW} \leq 63$. For the next transmission attempt applies $63 \leq \text{CW} \leq 127$. Further transmission attempts cause the value of the CW to increase exponentially until the maximum value is reached. As soon as a frame was acknowledged with ACK, i.e., the transmission was successful, the lower limit of the CW again corresponds to the value of the minimum CW in Table 6.7 [12].

CSMA/CA is the standard media access control method for WLAN and is implemented in all WLAN devices.

The duration of a SIFS and a DIFS depends on the modulation method used by the WLAN (see Table 5.5). Table 6.7 contains the different periods of a SIFS and a DIFS.

Another networking technology that uses CSMA/CA as the media access control method is *PowerLAN*, also called *Powerline Communication*, which uses the power grid of a building as a shared transmission medium [22].

CSMA/CA RTS/CTS

The media access control method CSMA/CA reduces the number of collisions, but cannot avoid all collisions. A better collision avoidance is implemented by CSMA/CA RTS/CTS. Sender

Weitere Stationen, die Rahmen senden wollen, warten nach dem ACK des Empfängers ein DIFS mit freiem Übertragungsmedium und eine *Backoffzeit* ab. Diese Wartezeit berechnet jede Station unabhängig von den anderen Stationen. Nach dem Ablauf der Backoffzeit wird der Rahmen gesendet. Belegt während der Backoffzeit eine andere Station das Übertragungsmedium, wird der Zähler so lange angehalten, bis das Medium mindestens ein DIFS lang wieder frei ist [26]. Die Backoffzeit wird berechnet, indem ein zufälliger Wert zwischen minimalem und maximalem Wert des *Contention Window* (CW) bestimmt wird, und dieser zufällige Wert wird mit der *Slot Time* multipliziert.

Der minimale und maximale Wert des CW sowie die Slot Time hängen vom verwendeten Modulationsverfahren ab und sind fest vorgegeben (siehe Tabelle 6.7). Die untere und obere Schranke des CW sind immer Zweierpotenzen, wobei vom Ergebnis der Wert 1 abgezogen wird. Verwendet ein WLAN beispielsweise das Modulationsverfahren DSSS ist beim ersten Sendeversuch $31 \leq \text{CW} \leq 63$. Beim nächsten Sendeversuch gilt $63 \leq \text{CW} \leq 127$. Weitere Sendeversuche lassen den Wert von CW weiter exponentiell ansteigen, bis der maximale Wert erreicht ist. Sobald ein Rahmen durch ACK bestätigt wurde, also die Übertragung erfolgreich war, entspricht die untere Schranke des CW wieder dem Wert des minimalen CW in Tabelle 6.7 [12].

CSMA/CA ist das standardmäßige Medienzugriffsverfahren bei WLAN und bei allen WLAN-Geräten implementiert.

Auch die Dauer eines SIFS und eines DIFS hängt davon ab, welches Modulationsverfahren das WLAN verwendet (siehe Tabelle 5.5). Tabelle 6.7 enthält die jeweilige Dauer eines SIFS und eines DIFS.

Eine weitere Vernetzungstechnologie, die CSMA/CA als Medienzugriffsverfahren verwendet, ist *PowerLAN*, das auch *Powerline Communication* heißt, und das das Stromnetz eines Gebäudes als gemeinsames Übertragungsmedium verwendet [22].

CSMA/CA RTS/CTS

Das Medienzugriffsverfahren CSMA/CA verringert die Anzahl der Kollisionen, kann aber nicht alle vermeiden. Eine bessere Kollisionsvermeidung ermöglicht CSMA/CA RTS/CTS. Sender

Table 6.7: SIFS, DIFS, Slot Time and Contention Window (CW) of the different Modulation Methods of WLAN

Modulation method	SIFS	DIFS[a]	Slot Time	Min. CW	Max. CW
FHSS (802.11)	28 μs	128 μs	50 μs	15	1023
DSSS (802.11b)	10 μs	50 μs	20 μs	31	1023
OFDM (802.11a/h/n/ac)	16 μs	34 μs	9 μs	15	1023
OFDM (802.11g)[b]	16 μs	34 μs	9 μs	15	1023
OFDM (802.11g)[c]	10 μs	50 μs	20 μs	15	1023

[a] DIFS = SIFS + 2 × Slot Time

[b] Supports data rates 1–54 Mbps

[c] Supports only data rates > 11 Mbps

and receiver in this method exchange *control frames* (see section 6.4.3) before the sender begins the transmission. This way, all stations in reach are aware that a transmission is about to begin. The control frames are called *Request To Send* (RTS) and *Clear To Send* (CTS).

und Empfänger tauschen bei diesem Verfahren *Kontrollrahmen* (siehe Abschnitt 6.4.3) aus, bevor der Sender mit der Übertragung beginnt. So wissen alle erreichbaren Stationen, dass demnächst eine Übertragung beginnt. Die Kontrollrahmen heißen *Request To Send* (RTS) und *Clear To Send* (CTS).

Figure 6.24: Media Access Control Method CSMA/CA RTS/CTS

After the DIFS, the sender transmits a *RTS* frame to the receiver (see Figure 6.24). The RTS frame contains a field that indicates for which time period the sender wants to reserve (use) the transmission medium (the channel). The receiver acknowledges this by waiting for the period of a SIFS and then by transmitting a *CTS* frame, which also contains the occupancy time for the transmission medium. This way, the receiver acknowledges the occupancy time for the frame to be transmitted.

After the receiver successfully received the frame, it waits for the period of a SIFS and sends an ACK frame to the sender. If the transmission medium (the channel) is occupied, no further transmission attempts are made, until

Der Sender überträgt nach dem DIFS einen *RTS*-Rahmen an den Empfänger (siehe Abbildung 6.24). Der RTS-Rahmen enthält ein Feld, das angibt wie lange der Sender das Übertragungsmedium (den Kanal) reservieren (benutzen) will. Der Empfänger bestätigt dies nach Abwarten des SIFS mit einem *CTS*-Rahmen, der ebenfalls die Belegungsdauer für das Übertragungsmedium enthält. Der Empfänger bestätigt somit die Belegungsdauer für den zu übertragenden Datenrahmens.

Der Empfänger sendet nach dem erfolgreichem Erhalt des Datenrahmens und nach dem Abwarten des SIFS einen ACK-Rahmen an den Sender. Ist das Übertragungsmedium (der Kanal) belegt, finden bis zum Ablauf des *Network*

the *Network Allocation Vector* (NAV) has expired.

The NAV is a counter variable that each station manages by itself. The NAV contains the time that the transmission medium is expected to be occupied. For example, if a station receives the information "*The transmission medium is occupied for the next x payload frames*", it stores the expected occupancy time in its NAV. The NAV is decremented over time until it reaches the value zero. As long as the NAV is greater than zero, a station does not attempt to transmit. It does not matter whether the transmission medium is idle or occupied.

Collisions because of the invisible or hidden terminal problem can only occur during the transmission of RTS and CTS frames. Figure 6.25 shows a scenario where station Y cannot receive the RTS frame from station X, but can receive the CTS frame from the access point.

Allocation Vectors (NAV) keine weiteren Sendeversuche statt.

Der NAV ist eine Zählvariable, die jede Station selbst verwaltet. Der NAV enthält die Zeit, die das Übertragungsmedium voraussichtlich belegt sein wird. Empfängt eine Station zum Beispiel die Information „*Das Übertragungsmedium ist für die nächsten x Datenrahmen belegt*", trägt sie die erwartete Belegungszeitspanne in ihren NAV ein. Der NAV wird mit der Zeit dekrementiert, bis er den Wert Null erreicht. Solange der NAV größer Null ist, unternimmt eine Station keine Sendeversuche. Dabei ist es egal, ob das Übertragungsmedium frei oder belegt ist.

Kollisionen sind wegen des Problems unsichtbarer bzw. versteckter Stationen (Hidden-Terminal) nur während dem Senden von RTS- und CTS-Rahmen möglich. Abb 6.25 zeigt ein Szenario, bei dem eine Station Y nicht den RTS-Rahmen von Station X empfangen kann, aber dafür den CTS-Rahmen der Basisstation (Access Point).

Figure 6.25: When using CSMA/CA RTS/CTS, Collisions that are caused by the Hidden Terminal Problem can only occur during the Transmission of RTS and CTS Frames

CSMA/CA RTS/CTS reduces collisions. Also, the energy consumption of the individual stations is reduced because there are no transmission attempts while the NAV is decremented. However, delays are caused by the reservations of the transmission medium, which are overhead. CSMA/CA RTS/CTS is optional for WLAN, but it is mostly implemented. In practice, it is

Durch CSMA/CA RTS/CTS werden Kollisionen reduziert. Zudem wird der Energieverbrauch der einzelnen Stationen reduziert, da es keine Sendeversuche gibt, während der NAV dekrementiert wird. Allerdings entstehen Verzögerungen durch die Reservierungen des Übertragungsmediums, die zudem einen Overhead darstellen. CSMA/CA RTS/CTS ist bei WLAN optional und meistens implementiert. In der Pra-

used to reserve channels for the transmission of long frames.

Usually, the firmware of access points or the driver software of WLAN devices allows for an RTS threshold value to be set. This makes it possible to influence the behavior of a station so that RTS/CTS is only used if a frame is longer than the threshold value. In practice, the preset threshold is often equal to or greater than the maximum frame length (2,346 bytes) of IEEE 802.11. As a consequence, a station skips the RTS/CTS sequence for all payload sent [23].

xis wird es zur Reservierung von Kanälen zur Übertragung langer Datenrahmen verwendet.

Üblicherweise ermöglicht die Firmware von Basisstationen bzw. Treibersoftware von WLAN-Schnittstellen einen RTS-Schwellenwert festlegen. Damit ist es möglich das Verhalten einer Station dahingehend zu beeinflussen, dass RTS/CTS nur dann verwendet wird, wenn ein Rahmen länger ist, als der Schwellenwert groß ist. In der Praxis entspricht der voreingestellte Schwellenwert häufig der maximalen Rahmenlänge (2.346 Bytes) bei IEEE 802.11 oder ist sogar größer. Als Konsequenz wird eine Station die RTS/CTS-Sequenz für alle gesendeten Datenrahmen weggelassen [23].

CSMA/CA PCF

With this method, the access point centrally controls the media access by requesting the stations registered with it to send frames. This approach is called *polling*. PCF stands for Point Coordination Function. Since CSMA/CA PCF is an optional method and is rarely used, it is not further described here.

CSMA/CA PCF

Bei diesem Verfahren steuert die Basisstation den Medienzugriff zentral, indem sie die bei ihr angemeldeten Stationen zum Senden von Datenrahmen auffordert. Dieses Vorgehen heißt *Polling*. PCF steht für Point Coordination Function. Da CSMA/CA PCF ein optionales Verfahren ist und nur selten verwendet wird, wird es an dieser Stelle nicht weiter näher beschrieben.

6.9

Flow Control

To prevent slow receivers in a physical network from receiving more frames than they can process – which would inevitably result in the loss of frames – depending on the specific application scenario, a protocol with flow control makes sense. With such a protocol, the receiver dynamically controls the transmission speed of the sender and ensures the integrity of the data transmission this way.

Examples of Data Link Layer protocols that provide flow control include High-Level Data Link Control (HDLC) and the Point-to-Point Protocol (PPP) that is based on it. Another example is *G.hn*, also called *HomeGrid*. Similar to PowerLAN or Powerline, this networking technology uses the power, telephone or cable television cables of a building as a shared transmission medium.

Flusskontrolle

Damit langsame Empfänger in einem physischen Netzwerk nicht mehr Rahmen erhalten als Sie verarbeiten können – was zwangsläufig den Verlust von Rahmen zur Folge hätte – ist je nach konkretem Anwendungsszenario ein Protokoll mit Flusskontrolle sinnvoll. Bei einem solchen Protokoll steuert der Empfänger die Sendegeschwindigkeit des Senders dynamisch und sichert so die Vollständigkeit der Datenübertragung.

Beispiele für Protokolle der Sicherungsschicht, die Flusskontrolle bieten, sind das High-Level Data Link Control (HDLC) und das darauf aufbauende Point-to-Point Protocol (PPP). Ein weiteres Beispiel ist *G.hn*, das auch *HomeGrid* heißt. Diese Vernetzungstechnologie nutzt ähnlich wie PowerLAN bzw. Powerline die Strom-, Telefon- oder Kabelfernsehkabel eines Gebäudes als gemeinsames Übertragungsmedium.

Ethernet, WLAN and Bluetooth do not provide flow control. If communication takes place via one of these networking technologies, flow control, where it is necessary, is usually implemented by the Transport Layer Protocol TCP (see section 8.4.3).

Ethernet, WLAN und Bluetooth bieten keine Flusskontrolle. Findet Kommunikation über eine dieser Vernetzungstechnologien statt, wird Flusskontrolle, da wo es erforderlich ist, meist vom Transportprotokoll TCP realisiert (siehe Abschnitt 8.4.3).

6.10

Address resolution with ARP

The Address Resolution Protocol – ARP (RFC 826) translates IP addresses of the Network Layer into MAC addresses of the Data Link Layer. If a network device wants to send data to a receiver, it uses the receiver's IP address (on the Network Layer). However, on the Data Link Layer the MAC address is required, which is why the address resolution must be carried out here.

To obtain the MAC address of a network device within the LAN, ARP sends a frame with the MAC broadcast address `FF-FF-FF-FF-FF-FF` as the destination address. This frame is received and analyzed by each network device in the LAN. The frame contains the IP address of the network device that is being searched for. If a network device identifies this IP address as its own, it sends an ARP response to the sender. The reported MAC address stores the sender in its local ARP cache.

This cache is used to speed up address resolution and contains a table in which protocol type (IP), protocol address (IP address), hardware address (MAC address) and expiration time – Time To Live (TTL) – are stored for each entry. The operating system determines the TTL. If an entry in the table is used, the TTL is extended.

ARP messages (see Figure 6.26) are transmitted as a payload in Ethernet frames. The field *HLEN* contains the length of the hardware addresses (MAC addresses) in bytes and *PLEN* contains the length of the protocol addresses (IP addresses) in bytes. MAC addresses of Ethernet are 6 bytes long, and IPv4 addresses are 4 bytes long.

Adressauflösung mit ARP

Das Address Resolution Protocol – ARP (RFC 826) übersetzt IP-Adressen der Vermittlungsschicht in MAC-Adressen der Sicherungsschicht. Will ein Netzwerkgerät Daten an einen Empfänger senden, gibt es auf der Vermittlungsschicht dessen IP-Adresse an. Auf der Sicherungsschicht ist aber die MAC-Adresse nötig, darum muss hier die Adressauflösung erfolgen.

Um die MAC-Adresse eines Netzwerkgeräts innerhalb des LAN zu erfahren, versendet ARP einen Rahmen mit der MAC-Broadcast-Adresse `FF-FF-FF-FF-FF-FF` als Zieladresse. Dieser Rahmen wird von jedem Netzwerkgerät im LAN entgegengenommen und ausgewertet. Der Rahmen enthält die IP-Adresse des gesuchten Netzwerkgeräts. Fühlt sich ein Gerät mit dieser IP-Adresse angesprochen, schickt es eine ARP-Antwort an den Sender. Die gemeldete MAC-Adresse speichert der Sender in seinem lokalen ARP-Cache.

Dieser Cache dient zur Beschleunigung der Adressauflösung und enthält eine Tabelle, in der für jeden Eintrag Protokolltyp (IP), Protokolladresse des Senders (IP-Adresse), Hardware-Adresse des Senders (MAC-Adresse) und Ablaufzeit – Time To Live (TTL) – festgehalten werden. Die TTL wird vom Betriebssystem festgelegt. Wird ein Eintrag in der Tabelle verwendet, verlängert sich die TTL.

ARP-Nachrichten (siehe Abbildung 6.26) werden im Nutzdatenteil von Ethernet-Rahmen übertragen. Das Feld *H-Länge* enthält die Länge der Hardwareadressen (MAC-Adressen) in Bytes und *P-Länge* die Länge der Protokolladressen (IP-Adressen) in Bytes. Bei Ethernet sind MAC-Adressen 6 Bytes lang und bei IPv4 sind IP-Adressen 4 Bytes lang.

Figure 6.26: Fields of ARP Messages

The sender MAC address is the MAC address of the requesting network device in an ARP request and the MAC address of the responding network device in an ARP response. The destination MAC address is not relevant in an ARP request and contains the MAC address of the requesting network device in an ARP response. The sender IP address is the IP of the requesting network device in an ARP request and the IP of the responding network device in an ARP response. The destination IP address is the IP address of the requested network device in an ARP request and it is the IP address of the requesting network device in an ARP response.

Die Sender-MAC-Adresse ist die MAC-Adresse des anfragenden Netzwerkgeräts bei einer ARP-Anforderung und die MAC-Adresse des antwortenden Netzwerkgeräts bei einer ARP-Antwort. Die Ziel-MAC-Adresse ist in einer ARP-Anforderung gleichgültig und enthält in einer ARP-Antwort die MAC-Adresse des anfragenden Netzwerkgeräts. Die Sender-IP-Adresse ist bei einer ARP-Anforderung die IP des anfragenden Netzwerkgeräts und bei einer ARP-Antwort die IP des antwortenden Netzwerkgeräts. Die Ziel-IP-Adresse ist bei einer ARP-Anforderung die IP-Adresse des gesuchten Netzwerkgeräts und bei einer ARP-Antwort die IP-Adresse des anfragenden Netzwerkgeräts.

7

Network Layer

The *Network Layer*, called *Internet Layer* in the TCP/IP reference model, is the third layer of the OSI reference model and the hybrid reference model. In this layer, the segments of the Transport Layer are packed in *packets* at the sender. At the receiver, the Network Layer detects the packets inside the frames of the Data Link Layer.

One task of the Network Layer is to determine the best path to the destination (routing) and the forwarding of the packets between logical networks, i.e. across different physical networks. For this so-called *Internetworking*, the delivery of packets across different physical networks, which can base on different networking technologies, logical addresses (IP addresses) are required, and the Network Layer specifies their format.

The most commonly used Network Layer protocol is the connectionless *Internet Protocol* (IP). Connectionless means that each IP packet is routed independently to its destination. The path is not recorded. Until the 1990s, other protocols than IP were popular in the Network Layer, such as Novell IPX/SPX and DECnet. Because these protocols do not play a role today, this chapter is wholly oriented towards IP.

Vermittlungsschicht

Die *Vermittlungsschicht*, die im TCP/IP-Referenzmodell *Internetschicht* heißt, ist die dritte Schicht des OSI-Referenzmodells und des hybriden Referenzmodells. In dieser Schicht werden beim Sender die Segmente der Transportschicht in *Pakete* verpackt. Beim Empfänger erkennt die Vermittlungsschicht die Pakete in den Rahmen der Sicherungsschicht.

Eine Aufgabe der Vermittlungsschicht ist die Ermittlung des besten Weges (Routing) und die Weiterleitung (Forwarding) der Pakete zwischen verschiedenen Netzen, also über physische Übertragungsabschnitte hinweg. Für dieses *Internetworking*, also die Zustellung der Pakete über verschiedene physische Netze, die auf beliebigen Vernetzungstechnologien basieren können, hinweg, sind logische Adressen (IP-Adressen) nötig, deren Format die Vermittlungsschicht definiert.

Das am häufigsten eingesetzte Verbindungsprotokoll ist das verbindungslose *Internet Protocol* (IP). Verbindungslos bedeutet, dass jedes IP-Paket unabhängig an sein Ziel vermittelt (*geroutet*) und der Pfad dabei nicht aufgezeichnet wird. Bis in die 1990er Jahre waren außer IP auch andere Protokolle in der Vermittlungsschicht wie zum Beispiel Novell IPX/SPX und DECnet populär. Da diese Protokolle heute aber keine Rolle mehr spielen, orientiert sich dieses Kapitel ganz am IP.

7.1

Devices of the Network Layer

A *Router* forwards packets between networks with different logical address ranges and, same as with Hubs and Switches, is equipped with

Geräte der Vermittlungsschicht

Ein *Router* leitet Datenpakete zwischen Netzen mit eigenen logischen Adressbereichen weiter und besitzt genau wie Hubs und Switche meh-

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_7

several interfaces. Also, Routers allow the *Local Area Network* to be connected to a *Wide Area Network* (e.g., via DSL or 3G/4G/5G mobile communication).

Layer-3-Switches provide similar functionalities as Routers. However, Layer-3-Switches are only used within local area networks to implement different logical address ranges, and they do not allow a connection with a WAN. Another typical functionality of Layer-3-Switches is the provision of virtual (logical) networks (see section 10.2).

Gateways (Protocol Translators) enable communication between networks based on different protocols by converting their protocols. Gateways that operate on the Network Layer are also called *Multiprotocol Routers*.

Modern computer networks almost exclusively operate with the Internet Protocol (IP). Therefore, a protocol conversion on the Network Layer is usually not necessary today. In former times, when configuring a terminal device, the IP address of the Gateway was specified as *Default Gateway*. Today, this field contains the Router address, because a Gateway is usually not required any longer. Thus, the term *Default Router* is more appropriate nowadays.

There are also VPN Gateways that can operate on the Network Layer (see section 10.1.1).

rere Schnittstellen. Zudem ermöglichen Router die Verbindung des lokalen Netzes (*Local Area Network*) mit einem *Wide Area Network* (z. B. über DSL oder Mobilfunk via 3G/4G/5G).

Eine ähnliche Funktionalität wie Router haben *Layer-3-Switche*. Diese werden aber nur innerhalb lokaler Netze verwendet, um verschiedene logischen Adressbereiche zu realisieren und sie ermöglichen keine Verbindung mit einem WAN. Eine weitere typische Funktionalität von Layer-3-Switchen ist die Bereitstellung virtueller (logischer) Netzwerke (siehe Abschnitt 10.2).

Gateways (Protokollumsetzer) ermöglichen Kommunikation zwischen Netzen, die auf unterschiedlichen Protokollen basieren, indem sie deren Protokolle konvertieren. Gateways, die auf der Vermittlungsschicht arbeiten, heißen auch *Mehrprotokoll-Router* oder *Multiprotokoll-Router*.

Moderne Computernetze arbeiten fast ausschließlich mit dem Internet Protocol (IP). Darum ist eine Protokollumsetzung auf der Vermittlungsschicht heute meist nicht nötig. In früheren Zeiten wurde bei der Konfiguration eines Endgeräts die IP-Adresse des Gateway als *Default Gateway* eintragen. Heute trägt man in diesem Feld den Router ein, weil man üblicherweise keinen Gateway mehr braucht. Der Begriff *Default Router* ist heute also eigentlich passender.

Es existieren auch VPN-Gateways die auf der Vermittlungsschicht arbeiten können (siehe Abschnitt 10.1.1).

7.1.1

Impact of Routers on the Collision Domain

Routers and Layer-3-Switches split the collision domain just like Bridges and Layer-2-Switches (see Figure 7.1).

Auswirkungen von Routern auf die Kollisionsdomäne

Router und Layer-3-Switches teilen genau wie Bridges und Layer-2-Switches die Kollisionsdomäne auf (siehe Abbildung 7.1).

7.1.2

Broadcast Domain

The *broadcast domain* is a logical part of a computer network where a broadcast reaches all network devices that belong to that part. Network devices that operate on the Physical Layer (*Repeaters* and *Hubs*) and the Data Link Layer

Broadcast-Domäne (Rundsendedomäne)

Die *Broadcast-Domäne* ist ein logischer Teil eines Computernetzes, bei dem ein Broadcast alle Netzwerkgeräte, die zu diesem Teil gehören, erreicht. Netzwerkgeräte der Bitübertragungsschicht (*Repeater* und *Hubs*) und Sicherungs-

Figure 7.1: Collision Domains for Routers and Layer-3-Switches

(*Bridges* and *Layer-2-Switches*) do not divide the broadcast domain. They operate transparently for logical networks. Network Layer devices (*Routers* and *Layer-3-Switches*), on the other hand, divide the broadcast domain (see Figure 7.2).

Broadcast domains consist of one or more collision domains. Devices of the Physical Layer do not divide the collision domain. The devices of the Data Link Layer and the Network Layer are different because they divide the collision domain.

Routers operate on the Network Layer. This means that each port of a Router is connected to a different IP network. This information is relevant for calculating the required number of subnets. Multiple Hubs, Switches, Repeaters or Bridges can operate in the same IP subnet, but not one IP subnet on multiple ports of a Router.

schicht (*Bridges* und *Layer-2-Switche*) teilen die Broadcast-Domäne nicht. Sie arbeiten aus Sicht logischer Netze transparent. Geräte der Vermittlungsschicht (*Router* und *Layer-3-Switche*) hingegen teilen die Broadcast-Domäne auf (siehe Abbildung 7.2).

Broadcast-Domänen bestehen aus einer oder mehreren Kollisionsdomänen. Geräte der Bitübertragungsschicht teilen die Kollisionsdomäne nicht. Anders die Geräte der Sicherungsschicht und Vermittlungsschicht, denn diese teilen die Kollisionsdomäne.

Router arbeiten auf der Vermittlungsschicht. Das heißt, an jedem Port eines Routers hängt ein anderes IP-Netz. Das ist wichtig, wenn man die Anzahl der nötigen Subnetze berechnen will. Man kann mehrere Hubs, Switche, Repeater oder Bridges im gleichen IP-Subnetz betreiben, aber nicht ein IP-Subnetz an mehreren Ports eines Routers.

7.2

Addressing in the Network Layer

Because of maintainability, it is not useful to exclusively use physical addressing via MAC addresses in computer networks with potentially global dimensions. Logical addresses are necessary, which are independent of the particular hardware. With logical addressing, the view of humans (logical addresses) is separated from the internal view of computers and software (physical addresses).

Every Network Layer packet has a destination address (IP address). An IP address (see Figure 7.3) can be assigned to a single receiver

Adressierung in der Vermittlungsschicht

Ausschließlich physische Adressierung via MAC-Adressen ist in Computernetzen mit eventuell globalen Ausmaßen wegen der Wartbarkeit nicht sinnvoll. Es sind logische Adressen nötig, die von der konkreten Hardware unabhängig sind. Mit logischer Adressierung wird die Teilnehmersicht für Menschen (logische Adressen) von der internen Sicht für Rechner und Software (physische Adressen) getrennt.

Zu jedem Paket auf der Vermittlungsschicht gehört eine Empfängeradresse (IP-Adresse). Eine IP-Adresse (siehe Abbildung 7.3) kann einen

Figure 7.2: Broadcast Domains and Collision Domains

Figure 7.3: An IP Address may address one or more Receivers

(*Unicast*) or a group of receivers (*Multicast*) or *Broadcast*). Multiple IP addresses can also be assigned to a network device. With *Anycast*, a single receiver of a group can be reached via a

einzelnen Empfänger (*Unicast*) oder eine Gruppe von Empfängern bezeichnen (*Multicast* oder *Broadcast*). Einem Netzwerkgerät können auch mehrere IP-Adressen zugeordnet sein. Bei *Anycast* erreicht man über eine Adresse einen einzel-

single address. Here, the receiver replies which can be accessed via the shortest route.

Examples of Multicast applications are the routing protocols RIPv2 (see section 7.4) and OSPF (see section 7.5) and the Network Time Protocol (NTP) for synchronizing clocks between computer systems [11]. An application example for Anycast is the Domain Name System, where some root name servers are implemented as logical servers that forward requests via Anycast to different physical servers (see section 9.1.1).

nen Empfänger aus einer Gruppe. Es antwortet der Empfänger, der über die kürzeste Route erreichbar ist.

Anwendungsbeispiele für Multicast sind die Routing-Protokolle RIPv2 (siehe Abschnitt 7.4) und OSPF (siehe Abschnitt 7.5) sowie das Network Time Protocol (NTP) zur Synchronisierung von Uhren zwischen Computersystemen [11]. Ein Anwendungsbeispiel für Anycast ist das Domain Name System, bei dem einige Root-Nameserver als logische Server realisiert sind, die Anfragen via Anycast auf verschiedene physische Server weiterleiten (siehe Abschnitt 9.1.1).

7.2.1

Format of IPv4 Addresses

IPv4 addresses have a length of 32 bits (4 bytes). Therefore $2^{32} = 4,294,967,296$ addresses can be represented. The usual representation is the so-called *dotted decimal notation*. Thereby the four octets are written as four integers values in decimal representation in the range from 0 to 255, separated from each other by dots. An example is 141.52.166.25.

Initially, IPv4 addresses were categorized into classes from A to C. The 32 bits of an IPv4 address consist of the two fields *Network identifier* (network ID) and *Host identifier* (host ID). A class A network has 7 bits for the representation of the network ID, and 24 bits for the host ID. A class B network has 14 bits for the network ID and 16 bits for the host ID. A class C network has 21 bits for the network ID and 8 bits for the host ID (see Figure 7.4).

Aufbau von IPv4-Adressen

IPv4-Adressen sind 32 Bits (4 Bytes) lang. Daher sind $2^{32} = 4.294.967.296$ Adressen darstellbar. Üblich ist die Darstellung in der sogenannten *Dotted Decimal Notation*. Dabei werden die vier Oktette als vier durch Punkte voneinander getrennte ganze Zahlen in Dezimaldarstellung im Bereich von 0 bis 255 geschrieben. Ein Beispiel ist 141.52.166.25.

Ursprünglich wurden IPv4-Adressen in Klassen von A bis C eingeteilt. Die 32 Bits einer IPv4-Adresse bestehen aus den beiden Feldern *Netzadresse* (Network Identifier bzw. Netzwerk-ID) und *Hostadresse* (Host Identifier bzw. Host-ID). In einem Klasse A-Netz gibt es 7 Bits für die Netzadresse und 24 Bits für die Hostadresse. In einem Klasse B-Netz gibt es 14 Bits für die Netzadresse und 16 Bits für die Hostadresse. In einem Klasse C-Netz gibt es 21 Bits für die Netzadresse und 8 Bits für die Hostadresse (siehe Abbildung 7.4).

Figure 7.4: Classes of IPv4 Addresses

The prefixes specify the address classes (see Table 7.1) and their address ranges. IPv4 is able to address $2^7 = 128$ class A networks with a maximum of $2^{24} = 16,777,216$ host addresses each. There are also $2^{14} = 16,384$ class B networks with a maximum of $2^{16} = 65,536$ host addresses each and $2^{21} = 2,097,152$ class C networks with a maximum of $2^8 = 256$ host addresses each.

Additionally, the classes D and E for special purposes existed. Class D contains Multicast addresses (for example for IPTV), and class E is reserved for future purposes and experimentation.

The idea to keep a part of the IPv4 address space unassigned for future applications has become absurd in the mean time because because a successor exists with IPv6. One reason for class E addresses not being used on the Internet is that some operating systems (e.g., Windows) did not accept these addresses and did not communicate with these addresses. Modifying this behavior did not seem to make sense, because it would not resolve the main problem of the address space being too small [3].

Die Präfixe legen die Netzklassen (siehe Tabelle 7.1) und ihre Adressbereiche fest. Bei IPv4 gibt es $2^7 = 128$ Klasse A-Netze mit jeweils maximal $2^{24} = 16.777.216$ Hostadressen. Zudem gibt es noch $2^{14} = 16.384$ Klasse B-Netze mit jeweils maximal $2^{16} = 65.536$ Hostadressen und $2^{21} = 2.097.152$ Klasse C-Netze mit jeweils maximal $2^8 = 256$ Hostadressen.

Es existierten zusätzlich die Klassen D und E für spezielle Aufgaben. Klasse D enthält Multicast-Adressen (zum Beispiel für IPTV) und Klasse E ist für zukünftige Verwendungen und Experimente reserviert.

Die Idee einen Teil des IPv4-Adressraums für zukünftige Verwendungen freizuhalten ist sinnlos geworden, weil mit IPv6 ein Nachfolger existiert. Ein Grund, dass in der Vergangenheit die Klasse E-Adressen im Internet nicht verwendet wurden, ist, dass einige Betriebssysteme (z.B. Windows) diese Adressen nicht akzeptieren und auch nicht mit ihnen kommunizieren. Eine Änderung dieses Verhaltens erschien nicht sinnvoll, weil dadurch das Hauptproblem, der zu knappe Adressraum, nicht umfassend gelöst würde [3].

Table 7.1: Prefixes and Address Ranges of the Address Classes of IPv4 Addresses

Class	Prefix	Address range	Network ID	Host ID
A	`0`	0.0.0.0 – 127.255.255.255	7 bits	24 bits
B	`10`	128.0.0.0 – 191.255.255.255	14 bits	16 bits
C	`110`	192.0.0.0 – 223.255.255.255	21 bits	8 bits
D	`1110`	224.0.0.0 – 239.255.255.255	—	—
E	`1111`	240.0.0.0 – 255.255.255.255	—	—

Only the classes A, B, and C are relevant in practice. Initially, the intention was to uniquely identify physical networks using their network ID. However, this procedure has some drawbacks. At a university with several internal networks, for example, at least one class C network with a maximum of 255 addresses would have to be reserved for each network. A company with more than 255 network devices would need at least one class B network.

Some drawbacks of the network classes are that they cannot be adapted dynamically to modifications of the network infrastructure and waste many addresses. A class C network with

Praktisch relevant sind nur die Klassen A, B und C. Ursprünglich war beabsichtigt, durch die Netzadresse physische Netze eindeutig zu identifizieren. Dieses Vorgehen bringt aber eine Reihe von Nachteilen mit sich. An einer Hochschule mit mehreren internen Netzen müsste man zum Beispiel für jedes Netz mindestens ein Klasse C-Netz mit maximal 255 Adressen reservieren. Ein Unternehmen mit mehr als 255 Netzwerkgeräten würde mindestens ein Klasse B-Netz benötigen.

Einige Nachteile der Netzklassen sind, dass sie nicht dynamisch an Veränderungen der Netzwerkinfrastruktur angepasst werden können und viele Adressen verschwenden. Ein Klasse C-Netz

two devices wastes 253 addresses. Class C networks can quickly run out of address space, and a class B network with 256 devices already wastes over 64,000 addresses. Also, there are only 128 class A networks. A later-on migration of that many devices when switching to another network class is a complex task.

mit zwei Geräten verschwendet 253 Adressen. Bei Klasse C-Netzen kann der Adressraum rasch knapp werden und ein Klasse B-Netz mit 256 Geräten verschwendet schon über 64.000 Adressen. Zudem gibt es nur 128 Klasse A-Netze und eine spätere Migration so vieler Geräte beim Wechsel der Netzklasse ist aufwändig.

7.2.2

Subnets in the IPv4 Address Space

For using the address space of the possible IPv4 addresses more efficiently, logical networks are divided into *partial networks*, which are usually called *subnets*. For this purpose, the administrator divides the available addresses in the host part of the IP address among several subnets.

The drawbacks of the strict division of the address space into network classes caused in 1993 the introduction of classless routing – *Classless Interdomain Routing* (CIDR).

To form subnets, a *(sub-)netmask* is necessary. All nodes of a network get a subnet mask assigned, which, like an IPv4 address, is 32 bits long and which determines the number of subnets and hosts. The subnet mask divides the host ID of the IP address into *subnet ID* and *host ID* (see Figure 7.5). The network ID remains unchanged. The subnet mask adds another hierarchy level to the IP address.

Subnetze im IPv4-Adressraum

Um den Adressraum der möglichen IPv4-Adressen effizienter zu verwenden, unterteilt man logische Netze in *Teilnetze*, die meist *Subnetze* genannt werden. Dafür teilt der Administrator die verfügbaren Adressen im Hostteil der IP-Adresse auf mehrere Subnetze auf.

Die Nachteile der strikten Unterteilung des Adressraums in Netzklassen führte 1993 zur die Einführung des klassenlosen Routings – *Classless Interdomain Routing* (CIDR).

Um Subnetze zu bilden, ist eine *Netzmaske* nötig. Alle Knoten in einem Netzwerk bekommen eine Netzmaske zugewiesen, die wie eine IPv4-Adresse 32 Bits lang ist und mit der die Anzahl der Subnetze und Hosts festgelegt wird. Die Netzmaske unterteilt die Hostadresse der IP-Adresse in *Subnetznummer* und *Hostadresse* (siehe Abbildung 7.5). Die Netznummer bleibt unverändert. Die Netzmaske fügt eine weitere Hierarchieebene in die IP-Adresse ein.

Figure 7.5: The Subnet Mask splits the Host ID of an IP Address into Subnet ID and Host ID

For example, to divide a class C network into 14 subnets, 4 bits are required. Each subnet needs a subnet number, and this must be represented in binary form. If 4 bits are used for the representation of the subnet numbers, 4 bits remain for the host ID part.

Um zum Beispiel ein Klasse C-Netz in 14 Subnetze aufzuteilen, sind 4 Bits nötig. Jedes Subnetz braucht nämlich seine eigene Subnetznummer und diese muss binär dargestellt werden. Werden 4 Bits für die Darstellung der Subnetz-

Since the introduction of the CIDR, IP address ranges are assigned in this notation: `First address/mask bits`. The mask bits are the number of one-bits in the subnet mask (see Table 7.2).

nummern verwendet, bleiben noch 4 Bits für den Hostteil.

Seit Einführung des CIDR werden IP-Adressbereiche in der Notation `Anfangsadresse/Netzbits` vergeben. Die Netzbits sind die Anzahl der Einsen in der Netzmaske (siehe Tabelle 7.2).

Table 7.2: Division of a Class C Network into Subnets

Mask bits (prefix)	/24	/25	/26	/27	/28	/29	/30	/31	/32
Subnet mask	0	128	192	224	240	248	252	254	255
Subnet bits	0	1	2	3	4	5	6	7	8
Subnets IDs	1	2	4	8	16	32	64	128	256
Host bits	8	7	6	5	4	3	2	1	0
Host IDs	256	128	64	32	16	8	4	2	—
Hosts (maximum)	254	126	62	30	14	6	2	0	—

Two host addresses cannot be assigned to nodes. The reason for this is that each (sub)network needs a subnet address for the network itself and a broadcast address to address all nodes in the network. In the subnet address, every bit of the host ID has the value zero and in the broadcast address, every bit of the host ID has the value one.

Also, the subnet IDs, which consist entirely of zero-bits and one-bits, should not be used. Although this rule is obsolete, it is still applied frequently. Modern Routers and network software can assign all possible subnet IDs to existing subnets.

Zwei Hostadressen dürfen nicht an Knoten vergeben werden. Der Grund dafür ist, dass jedes (Sub-)Netz eine Subnetzadresse für das Netz selbst benötigt und eine Broadcast-Adresse, um alle Knoten im Netz zu adressieren. Bei der Subnetzadresse haben alle Bits der Hostadresse den Wert Null und bei der Broadcast-Adresse haben alle Bits der Hostadresse den Wert Eins.

Zudem sollen die Subnetznummern, die ausschließlich aus Nullen und ausschließlich aus Einsen bestehen, nicht verwendet werden. Diese Regel ist zwar veraltet, wird aber häufig noch angewendet. Moderne Router und Netzwerksoftware haben kein Problem damit, wenn alle möglichen Subnetznummern für existierende Subnetze vergeben werden.

By using Table 7.2, it is simple to determine the required bits for subnets. If, for example, a network administrator wants to divide a class C network into five subnets with a maximum of 25 hosts each, it must only be clear that three subnet bits are required to implement five subnets because each subnet requires a subnet ID. With the remaining 5 bits in the host part, up to $32 - 2 = 30$ hosts can be addressed in each subnet. Thus, the subnet mask with the prefix `/27` is well suited for this use case.

Anhand Tabelle 7.2 ist es einfach, die nötigen Bits für Subnetze zu bestimmen. Soll ein Netzwerkverwalter beispielsweise ein Klasse C-Netz in fünf Subnetze mit jeweils maximal 25 Hosts aufteilen, muss nur klar sein, dass zur Realisierung von fünf Subnetzen drei Subnetzbits nötig sind, weil jedes Subnetz eine Subnetznummer benötigt. Mit Hilfe der restlichen 5 Bits im Hostteil können in jedem Subnetz bis zu $32 - 2 = 30$ Hosts adressiert werden. Somit ist die Schrägstrichdarstellung `/27` geeignet.

Calculation example for Subnetting with IPv4

A sample calculation illustrates the correlations. The number behind the slash of the IP address

Rechenbeispiel zu Subnetzen bei IPv4

Ein Rechenbeispiel verdeutlicht die Zusammenhänge. Die Zahl hinter dem Schrägstrich

172.21.240.90/27 is the number of one-bits in the subnet mask. The subnet address is calculated by performing the `AND` (`1 AND 1 = 1, 1 AND 0 = 0, 0 AND 1 = 0, 0 AND 0 = 0`) operation on the IP address and the subnet mask.

der IP-Adresse 172.21.240.90/27 ist die Anzahl der Einsen in der Netzmaske. Werden IP-Adresse und Netzmaske via `AND` (`1 AND 1 = 1, 1 AND 0 = 0, 0 AND 1 = 0, 0 AND 0 = 0`) verknüpft, erhält man die Subnetzadresse.

IP address	172.21.240.90	10101100 00010101 11110000 01011010
Subnet mask	255.255.255.224	11111111 11111111 *11111111 111*00000
Subnet address	172.21.240.64	10101100 00010101 11110000 01000000
Subnet ID	1922	10101100 00010101 **11110000 01000000**

The boundary between the subnet ID and the network ID results from the network class of the IP address. 172.21.240.90/27 is an IP address from a class B network. This results from the prefix (see Table 7.1) of the IP address.

The host ID is calculated by performing the `AND` operation on the IP address and the inverse subnet mask.

Die Grenze zwischen Subnetznummer und Netzadresse ergibt sich aus der Netzklasse der IP-Adresse. 172.21.240.90 ist eine IP-Adresse aus einem Klasse B-Netz. Das ergibt sich aus dem Präfix (siehe Tabelle 7.1) der IP-Adresse.

Wird die IP-Adresse mit der negierten Netzmaske via `AND` verknüpft, erhält man die Hostadresse.

IP address	172.21.240.90	10101100 00010101 11110000 01011010
Subnet mask	255.255.255.224	11111111 11111111 11111111 111*00000*
Inv. subnet mask	000.000.000.31	00000000 00000000 00000000 000*11111*
Host ID	26	00000000 00000000 00000000 00011010

For a subnet mask with 27 one-bits, 11 bits are used for the representation of the subnet ID in a class B network. 5 bits and thus $2^5 = 32$ addresses remain for the representation of the host IDs. One of these addresses is required for the network itself (all bits of the host ID are zero-bits), and one address is used as the broadcast address (all bits of the host ID are one-bits). This means that 30 host addresses are available for network devices.

Bei einer Netzmaske mit 27 gesetzten Bits werden in einem Klasse-B-Netz 11 Bits für die Subnetznummer verwendet. Es verbleiben 5 Bits und damit $2^5 = 32$ Adressen für den Hostteil. Davon wird eine Adresse (Netzadresse für das Netz selbst (alle Bits im Hostteil auf Null) und eine als Broadcast-Adresse (alle Bits im Hostteil auf Eins) benötigt. Somit sind 30 Hostadressen für Netzwerkgeräte verfügbar.

7.2.3

Private IPv4 Address Spaces

IP addresses must also be assigned in private computer networks, and these should not collide with existing Internet services. For this purpose, address ranges with private IP addresses (RFC 1918) exist, which are not routed on the Internet (see Table 7.3).

If a private network shall be connected to the Internet, the local Router usually performs an automatic replacement of the private addresses with the public address that is provided by the Internet Service Provider (ISP) [20]. This tech-

Private IPv4-Adressbereiche

Auch in privaten Computernetzen müssen IP-Adressen vergeben werden und diese sollten nicht mit real existierenden Internetangeboten kollidieren. Dafür existieren Adressbereiche mit privaten IP-Adressen (RFC 1918), die im Internet nicht geroutet werden (siehe Tabelle 7.3).

Soll ein privates Netz mit dem Internet verbunden werden, erfolgt in der Praxis üblicherweise durch den lokalen Router ein automatisches Ersetzen der privaten Adressen in die vom Internet Service Provider (ISP) zugewiesene öf-

Table 7.3: Private IPv4 Address Spaces

Address space:	10.0.0.0 to 10.255.255.255
CIDR notation:	10.0.0.0/8
Number of addresses:	$2^{24} = 16{,}777{,}216$
Address class:	Class A. One private network with 16,777,216 addresses
Address space:	172.16.0.0 to 172.31.255.255
CIDR notation:	172.16.0.0/12
Number of addresses:	$2^{20} = 1{,}048{,}576$
Address class:	Class B. 16 private networks with 65,536 addresses each
Address space:	192.168.0.0 to 192.168.255.255
CIDR notation:	192.168.0.0/16
Number of addresses:	$2^{16} = 65{,}536$
Address class:	Class C. 256 private networks with 256 addresses each

nique is called *Network Address Translation* – NAT (see section 7.7).

fentliche Adresse [20]. Dieses Verfahren heißt *Network Address Translation* – NAT (siehe Abschnitt 7.7).

7.2.4

Structure of IPv4 Packets

Figure 7.6 shows the structure of IPv4 packets. The first field is 4 bits long and contains the protocol version. For IPv4 packets, the field contains the value `4`. For IPv6 packets, it contains the value `6`. The content of the field *IHL* (IP Header Length) contains the length of the IP header in multiples of 4 bytes and indicates where the payload starts. For example, if the field contains the value `5`, this means that the length of the IP header is $5 * 4 = 20$ bytes. The field *Service* can be used to prioritize IP packets (Quality of Service). The field has been modified several times (RFC 791, RFC 2474, RFC 3168).

The length of the entire IP packet in bytes including the header is specified in the field *Total length*. This field is 16 bits in size, and therefore the maximum packet length that is possible with IPv4 is 65,535 bytes. The fields *Identification*, *Flags*, and *Fragment offset* control the assembly of previously fragmented IP packets. With *Flags*, the sender indicates whether the packet is allowed to be fragmented and the receiver learns if more fragments follow. *Identification* contains an unique identifier of the IP packet and *Fragment offset* a number that indicates the

Aufbau von IPv4-Paketen

Abbildung 7.6 zeigt den Aufbau von IPv4-Paketen. Das erste Datenfeld ist 4 Bits lang und enthält die Version des Protokolls. Bei IPv4-Paketen enthält das Feld den Wert `4`. Bei IPv6-Paketen enthält es den Wert `6`. Das Datenfeld *IHL* (IP Header Length) enthält die Länge des IP-Headers in Vielfachen von 4 Bytes und zeigt an, wo die Nutzdaten beginnen. Enthält das Feld beispielsweise den Wert `5`, bedeutete das, dass die Länge des IP-Headers $5*4 = 20$ Bytes ist Mit dem Datenfeld *Service* ist eine Priorisierung von IP-Paketen möglich (Quality of Service). Das Feld wurde mehrfach verändert (RFC 791, RFC 2474, RFC 3168).

Die Länge des gesamten IP-Pakets in Bytes inklusive Header ist im Feld *Paketlänge* angegeben. Dieses Datenfeld ist 16 Bits groß und darum ist die maximale Paketlänge, die mit IPv4 möglich ist, 65.535 Bytes. Die Datenfelder *Kennung*, *Flags* und *Fragment Offset* steuern das Zusammensetzen von zuvor fragmentierten IP-Paketen. Mit *Flags* gibt der Sender an, ob das Paket fragmentiert werden darf und der Empfänger erfährt, ob noch weitere Fragmente folgen. *Kennung* enthält eine eindeutige Kennung des IP-Pakets und *Fragment Offset* eine Nummer, die besagt, ab welcher Position inner-

Figure 7.6: Fields of an IPv4 Packet

position within the unfragmented packet where the fragment starts.

The content of *Time To Live* specifies the maximum time in seconds that an IP packet is allowed to exist during transmission. If the value is zero, the Router discards the packet. In practice, the field contains the maximum number of possible hops, and each Router on the way to the destination decrements the value by one. This is to prevent undeliverable IP packets from circling endlessly in the network.

The field *Protocol ID* contains the number of the higher-level protocol. If there is an ICMP message in the payload, the field has the value `1`. For TCP segments the value is `6`, and for UDP segments the value is `17`. If there is a message of the link state routing protocol OSPF in the payload, the value of the field is `89`.

Each IPv4 packet also contains a field for a 16 bits long checksum which is calculated from the data of the header. Because every Router on the path to the destination modifies the content of the *Time To Live* field, every Router would have to verify the checksum, recalculate it and insert it into the header. Routers usually ignore the checksum to enable faster forwarding of the packets. Therefore IPv6 packets do not contain a checksum field.

halb des unfragmentierten Pakets das Fragment anfängt.

Der Inhalt von *Time To Live* gibt die maximale Zeitdauer der Existenz eines IP-Pakets während der Übertragung in Sekunden an. Ist der Wert Null, wird das Paket vom Router verworfen. In der Praxis enthält das Datenfeld die maximal möglichen Hops und jeder Router auf dem Weg zum Ziel verringert den Wert um eins. So wird verhindert, dass unzustellbare IP-Pakete endlos im Netz umherirren (kreisen).

Das Datenfeld *Protokoll-ID* enthält die Nummer des übergeordneten Protokolls. Befindet sich im Nutzdatenbereich eine ICMP-Nachricht, hat das Feld den Wert `1`. Bei TCP-Segmenten ist der Wert `6` und bei UDP-Segmenten ist der Wert `17`. Befindet sich im Nutzdatenbereich eine Nachricht des Link-State-Routing-Protokolls OSPF, hat das Feld den Wert `89`.

Jedes IPv4-Paket enthält auch ein Feld für eine 16 Bits große Prüfsumme über die Daten des Headers. Weil sich bei jedem Router auf dem Weg zum Ziel der Inhalt des Datenfelds *Time To Live* ändert, müsste jeder Router die Prüfsumme überprüfen, neu berechnen und in den Header einsetzen. Die Router ignorieren die Prüfsumme üblicherweise, um die Pakete schneller weiterleiten zu können. Darum enthalten IPv6-Pakete auch kein Datenfeld für eine Prüfsumme.

The 32 bits long field *IP address (sender)* contains the IP address of the sender, the field *IP address (destination)* contains the IP address of the destination. The *Options/Padding* field can contain additional information such as a timestamp. This final field before the payload is filled with padding bits (zero-bits) if necessary, because its size must be a multiple of 32 bits, just like the entire header. The final area contains the data of the Transport Layer.

Das 32 Bits große Datenfeld *IP-Adresse (Sender)* enthält die IP-Adresse des Senders und das Datenfeld *IP-Adresse (Ziel)* die IP-Adresse des Ziels. Das Feld *Optionen/Füllbits* kann Zusatzinformationen wie einen Zeitstempel enthalten. Dieses letzte Feld vor dem Datenbereich mit den Nutzdaten wird gegebenenfalls mit Füllbits (Nullen) aufgefüllt, weil es wie der vollständige Header auch ein Vielfaches von 32 Bits groß sein muss. Der abschließende Datenbereich enthält die Daten der Transportschicht.

7.2.5

Fragmentation of IPv4 Packets

The splitting (and reassembly) of longer IP packets into smaller packets is called *fragmentation* and is usually performed by Routers at the boundaries of logical networks. The sender can also perform fragmentation. The reason why IP packet fragmentation exists at all is that there are networks with different maximum packet lengths – *Maximum Transmission Unit* (MTU). For example, the MTU of Ethernet is usually 1,500 bytes, for PPPoE it is 1,492 bytes, and for WLAN it is 2,312 bytes. Table 6.5 contains a selection of typical MTU sizes of popular networking technologies.

Each fragment is a separate IP packet that is transmitted independently of the other fragments [9].

IP packets contain a flag with which fragmentation can be suppressed. If a Router needs to fragment a packet because it is too big to forward, but fragmentation is suppressed as defined in the packet, the Router discards the packet because it cannot forward it.

Network devices that do not receive all fragments of an IP packet within a specified period (a few seconds) discard all received fragments.

Routers can split IP packets into smaller fragments if the MTU requires this and if it is not suppressed in the packets. However, Routers are not able to reassemble the fragments of a packet in order to create a larger fragment. Fragments can only be reassembled by the receiver.

Fragmentieren von IPv4-Paketen

Das Zerlegen (und Zusammensetzen) längerer IP-Pakete in kleinere Pakete heißt *Fragmentieren* und wird in der Regel von Routern an den Übergängen von logischen Netzen durchgeführt. Fragmentieren kann aber auch der Sender durchführen. Der Grund, warum das Fragmentieren von IP-Paketen überhaupt existiert, ist weil es Netze mit unterschiedlicher maximaler Paketlänge – *Maximum Transmission Unit* (MTU) gibt. Beispielsweise ist die MTU von Ethernet meist 1.500 Bytes, bei PPPoE ist sie 1.492 Bytes und bei WLAN ist sie 2.312 Bytes. Tabelle 6.5 enthält eine Auswahl typischer MTU-Größen bekannter Vernetzungstechnologien.

Jedes Fragment ist ein eigenständiges IP-Paket, das unabhängig von den anderen Fragmenten übertragen wird [9].

In IP-Paketen existiert ein Flag, mit dem das Fragmentieren untersagt werden kann. Müsste ein Router ein Paket fragmentieren, weil es für die Weiterleitung zu groß ist, aber die Fragmentierung ist im Paket untersagt, verwirft der Router das Paket, da er es nicht weiterleiten kann.

Netzwerkgeräte, die nicht alle Fragmente eines IP-Pakets innerhalb einer bestimmten Zeitspanne (wenige Sekunden) erhalten, verwerfen alle empfangenen Fragmente.

Router können IP-Pakzusammenete in kleinere Fragmente unterteilen, wenn die MTU dieses nötig macht und es in den Paketen nicht untersagt ist. Kein Router kann aber Fragmente eines Pakets zu einem größeren Fragment zusammenfügen. Das Zusammenfügen der Fragmente ist nur beim Empfänger möglich.

7.2.6

Representation and Structure of IPv6 Addresses

In recent years, the introduction of IPv6 for large networks has been promoted, which makes sense due to the limited address space of IPv4. IPv6 addresses (RFC 4291) consist of 128 bits (16 bytes). Therefore it is possible to represent $2^{128} \approx 3.4 \times 10^{38}$, i.e. more than 340 sextillion, possible addresses. Considering that the surface of the planet earth covers about 510 million square kilometers, each square millimeter (including the oceans) can get more than 660 billion IPv6 addresses assigned.

Due to the large address space, the decimal representation of IPv6 addresses is somewhat confusing in comparison to the more intuitive IPv4 addresses. For this reason, it is common to represent IPv6 addresses in hexadecimal notation. In this case, 4 bits each are represented as a hexadecimal number, and four hexadecimal numbers each are grouped into blocks. Colons separate the blocks. One example of this representation of an IPv6 address is:

Darstellung und Aufbau von IPv6-Adressen

Seit einigen Jahren wird für große Netze die Einführung von IPv6 forciert, was wegen des begrenzten Adressraums von IPv4 sinnvoll ist. IPv6-Adressen (RFC 4291) bestehen aus 128 Bits (16 Bytes). Darum ist es möglich $2^{128} \approx 3,4 * 10^{38}$, also über 340 Sextillionen, mögliche Adressen darzustellen. Bedenkt man, dass die Oberfläche der Erde ca. 510 Millionen Quadratkilometer umfasst, können jedem Quadratmillimeter (inklusive der Ozeane) über 660 Billiarden IPv6-Adressen zugewiesen werden.

Wegen des großen Adressraums ist die Dezimaldarstellung von IPv6-Adressen im Vergleich zu den eingängigen IPv4-Adressen eher unübersichtlich. Darum ist es üblich, IPv6-Adressen in hexadezimaler Schreibweise darzustellen. Hierbei werden je 4 Bits als eine hexadezimale Zahl dargestellt und je vier Hexadezimalzahlen werden zu Blöcken gruppiert. Die Blöcke werden durch Doppelpunkte getrennt. Ein Beispiel für diese Darstellung einer IPv6-Adresse ist:

```
2001:0db8:85a3:08d3:1319:8a2e:0370:7344
```

The final 4 bytes of an IPv6 address may also be written in decimal notation. This is useful to embed the IPv4 address space into the IPv6 address space.

For a more compact representation, different rules for simplification exist (RFC 5952). For example, leading zeros may be omitted within a block. Also, consecutive blocks whose values are zero (or 0000) may be omitted exactly once within an IPv6 address. Two consecutive colons indicate the omission. If multiple groups of zero blocks exist, it is recommended to truncate the group with the most zero blocks.

Die letzten 4 Bytes einer IPv6-Adresse dürfen auch in dezimaler Notation geschrieben werden. Das ist sinnvoll, um den IPv4-Adressraum in den IPv6-Adressraum einzubetten.

Zur kompakteren Darstellung existieren verschiedene Regeln zur Vereinfachung (RFC 5952). So dürfen führende Nullen innerhalb eines Blocks ausgelassen werden. Zudem dürfen aufeinanderfolgende Blöcke, deren Wert 0 (bzw. 0000) ist, genau einmal innerhalb einer IPv6-Adresse ausgelassen werden. Das Auslassen zeigen zwei aufeinander folgende Doppelpunkte an. Gibt es mehrere Gruppen aus Null-Blöcken, ist es empfehlenswert die Gruppe mit den meisten Null-Blöcken zu kürzen.

The DNS root name server (see section 9.1) with the URL `j.root-servers.net` serves as an example here. This can be accessed via this IPv6 address:

Als Beispiel dient an dieser Stellte der DNS-Root-Nameserver (siehe Abschnitt 9.1) mit der URL `j.root-servers.net`. Dieser ist unter folgender IPv6-Adresse erreichbar:

```
2001:0503:0c27:0000:0000:0000:0002:0030
```

The rules described for simplification shorten the address representation as follows:

Durch die beschrieben Regeln zur Vereinfachung verkürzt sich die Darstellung der Adresse wie folgt:

```
2001:503:c27::2:30
```

If, in addition to the IP address, the port number (see section 8.2) of the destination process shall be specified, the IPv6 address is enclosed in square brackets [18][32]. This prevents the port number, which is appended to the IP address with a colon, to be interpreted as a part of the IPv6 address. One example for this representation is:

Soll zusätzlich zur IP-Adresse noch die Portnummer (siehe Abschnitt 8.2) des Zielprozesses angegeben sein, wird die IPv6-Adresse in eckigen Klammern eingeschlossen dargestellt [18][32]. Das verhindert, das die Portnummer, die mit einem Doppelpunkt an die IP-Adresse angehängt ist, als Teil der IPv6-Adresse interpretiert wird. Ein Beispiel für diese Darstellung ist:

```
http://[2001:500:1::803f:235]:8080/
```

In this way, in addition to the protocol of the Application Layer, the addresses of the Network Layer and the Transport Layer are specified in a non-ambiguous way.

As with IPv4 before the introduction of classless interdomain routing (see section), IPv6 addresses consist of two fields (see Figure 7.7), of which the first field (*Prefix* or *Network Prefix*) identifies the network and the second field (*Interface Identifier* or *Interface ID*) identifies the network device in a network.

Auf diese Art und Weise sind neben dem Protokoll der Anwendungsschicht die Adressen der Vermittlungsschicht und der Transportschicht eindeutig angegeben.

Genau wie bei IPv4 vor der Einführung des klassenlosen Routings (siehe Abschnitt) bestehen IPv6-Adressen aus zwei Feldern (siehe Abbildung 7.7), von denen das erste Feld (*Präfix* bzw. *Network Prefix*) das Netz kennzeichnet und das zweite Feld (*Interface Identifier* bzw. *Interface-ID*) das Netzwerkgerät in einem Netz.

64 Bits	64 Bits
Network Prefix	Interface Identifier
2001:638:208:ef34	:0:ff:fe00:65

Figure 7.7: Fields of an IPv6 Address

The interface identifier can be set manually, assigned via DHCPv6 or derived from the MAC address of the network interface. If the interface identifier is derived from the MAC address, it is called the *Extended Unique Identifier* (EUI). In this case, the MAC address is converted into the so-called modified EUI-64 address format. In other words: The 48 bits long MAC address is converted to a 64 bits long address.

Using this conversion, the MAC address is divided into two parts, each 24 bits long. The first part is the first 24 bits and the second part the final 24 bits of the modified EUI-64 address. The free 16 bits in the middle of the EUI-64 address

Der Interface Identifier kann manuell festgelegt, via DHCPv6 zugewiesen oder aus der MAC-Adresse der Netzwerkschnittstelle gebildet werden. Wird der Interface Identifier aus der MAC-Adresse gebildet, heißt er *Extended Unique Identifier* (EUI). Dabei wird die MAC-Adresse in das sogenannte modifizierte EUI-64 Adressformat konvertiert. Einfacher ausgedrückt: Die 48 Bits lange MAC-Adresse wird in eine 64 Bits lange Adresse umgewandelt.

Bei dieser Umwandlung wird die MAC-Adresse in zwei je 24 Bits lange Teile unterteilt. Der erste Teil bildet die ersten 24 Bits und der zweite Teil die letzten 24 Bits der modifizierten EUI-64-Adresse. Die freien 16 Bits

get this bit pattern: `1111 1111 1111 1110`, which is represented in hexadecimal notation as: `FFFE`. Finally, the value of the seventh bit from the left is inverted [31]. The method is shown stepwise in Figure 7.8 on the basis of an example.

in der Mitte der EUI-64-Adresse erhalten folgendes Bitmuster: `1111 1111 1111 1110`, was in hexadezimaler Schreibweise folgendermaßen dargestellt wird: `FFFE`. Abschließend wird der Wert des siebten Bits von links invertiert [31]. Die Vorgehensweise zeigt Abbildung 7.8 anhand eines Beispiels Schritt für Schritt.

Figure 7.8: Conversion of a MAC Address into an Extended Unique Identifier (EUI)

7.2.7

Representation of Networks in the IPv6 Address Space

In contrast to IPv4, IPv6 has no subnet masks. The subdivision of address ranges into subnets is done by specifying the prefix length.

The definition of IPv6 networks is done precisely in the same way as with IPv4 in CIDR notation (see Figure 7.10). Internet Service Providers (ISP) and operators of large networks get assigned the first 32 bits or 48 bits by a Regional Internet Registry (RIR). Each ISP or network operator independently divides this address space into subnets. Consumers usu-

Darstellung von Netzen im IPv6-Adressraum

Im Gegensatz zu IPv4 existieren bei IPv6 keine Netzmasken. Die Unterteilung von Adressbereichen in Subnetze geschieht durch die Angabe der Präfixlänge.

Die Definition von IPv6-Netzen geschieht genau wie bei IPv4 in CIDR-Notation (siehe Abbildung 7.10). Internet Service Provider (ISP) und Betreiber großer Netze bekommen die ersten 32 Bits oder 48 Bits von einer Regional Internet Registry (RIR) zugewiesen. Diesen Adressraum teilt jeder Provider oder Netzbetreiber selbständig in Subnetze auf. Endkunden bekom-

ally get assigned a /64 network or even a /56 network [32].

men meist ein /64-Netz oder sogar /56-Netz zugeteilt [32].

Figure 7.9: Specification of IPv6 Networks in CIDR Notation

If a customer gets assigned a /56 network, the 8 bits between the prefix and the interface ID are the *Subnet prefix*. With this range, it is possible for a customer who operates several networks to implement these.

Bekommt ein Endkunde ein /56-Netz zugeteilt, sind die 8 Bits zwischen dem Präfix und der Interface-ID das *Subnet Präfix*. Mit diesem Bereich kann ein Endkunde, der mehrere Netze betreibt, diese realisieren.

7.2.8

Selected IPv6 Address Spaces

Some prefixes that are relevant in practice are shown in Table 7.4 [32].

Einige IPv6-Adressbereiche

Einige für die Praxis relevante Präfixe enthält Tabelle 7.4 [32].

Table 7.4: Some IPv6 Address Spaces

Address space	Purpose
fe80::/10	Link-local addresses. They are only valid in the local network and are therefore not forwarded by Routers.
2000::/3	Global unicast addresses. The addresses 2000... to 3fff... are forwarded worldwide by Routers.
ff00::/8	All addresses ff... are Multicast addresses. Since IPv6 has no broadcast addresses, Multicast addresses implement the broadcast functionality. The addresses ff01::1 and ff02::1 address all nodes in the local network and the addresses ff01::2, ff02::2 and ff05::2 address all local Routers.
2001:db8::/32	Addresses only for documentation purposes.
0:0:0:0:0:FFFF::/96	IPv4-mapped IPv6 address.

7.2.9

Embed IPv4 Addresses into the IPv6 Address Space

A globally routed (unicast) IPv4 address can be represented as an IPv6 address and thus be integrated into the IPv6 address space. This method is called *IPv4-mapped IPv6 address* [32]. The IPv4 address gets the 96 bytes long prefix: 0:0:0:0:0:0:FFFF::/96

IPv4-Adressen im IPv6-Adressraum einbetten

Eine global geroutete (Unicast) IPv4-Adresse kann als IPv6-Adresse dargestellt und somit in den IPv6-Adressraum integriert werden. Diese Vorgehensweise heißt *IPv4-mapped IPv6-Adresse* [32]. Die IPv4-Adresse erhält den 96 Bytes langen Präfix: 0:0:0:0:0:FFFF::/96

80 Bits	16 Bits	32 Bits
0000 0000 0000 0000 0000	FFFF	IPv4 address

Figure 7.10: Embed IPv4 Addresses into the IPv6 Address Space

The embedded IPv4 address may be represented in hexadecimal or, for better readability, in decimal notation [9].

For example, if the IPv4 address `131.246.107.35` shall be embedded in the IPv6 address space, the above mentioned prefix is added, resulting in this IPv6 address: `0:0:0:0:0:FFFF:83F6:6B23`. Valid abbreviations include `::FFFF:83F6:6B23` and `::FFFF:131.246.107.35` [17][20].

Die eingebettete IPv4-Adresse darf in hexadezimaler oder zur besseren Lesbarkeit auch in dezimaler Schreibweise dargestellt sein [9].

Soll beispielsweise die IPv4-Adresse `131.246.107.35` in den IPv6-Adressraum eingebettet werden, wird diese um den oben genannten Präfix ergänzt, was folgende IPv6-Adresse ergibt: `0:0:0:0:0:FFFF:83F6:6B23`. Zulässige Kurzschreibweisen sind `::FFFF:83F6:6B23` sowie `::FFFF:131.246.107.35` [17][20].

7.2.10

Structure of IPv6 Packets

Figure 7.11 shows the structure of IPv6 packets and their strong similarity to the structure of IPv4 packets (see Figure 7.6). The header of IPv6 packets has a fixed length of 320 bits (40 bytes) [23]. The field *Next Header* can refer to an extension header (*Extension Header*) or the protocol of the Transport Layer (e.g., TCP = Type 6 or UCP = Type 17) [17][18].

Aufbau von IPv6-Paketen

Abbildung 7.11 zeigt den Aufbau von IPv6-Paketen und deren große Ähnlichkeit zur Struktur von IPv4-Paketen (siehe Abbildung 7.6). Der Header von IPv6-Paketen hat eine feste Länge von 320 Bits (40 Bytes) [23]. Im Feld *Next Header* kann auf einen Erweiterungs-Kopfdatenbereich (*Extension Header*) oder das Protokoll der Transportschicht (z.B. TCP = Typ 6 oder UCP = Typ 17) verwiesen werden [17][18].

Figure 7.11: Fields of an IPv6 Packet

The extension headers offer a high degree of flexibility since they can be used to implement extensions in IPv6 that were not planned in the protocol during its development and do not require a modification of the actual IPv6 header. Another benefit is that there is no size limit for the extension headers. The only constraint is that the size of each extension header must be a multiple of 64 bits (8 bytes).

Each header refers to a subsequent header. This creates a chain until the header of the Transport Layer protocol is reached. Each extension header has a specific number. Some extension headers are:

- *Hop-By-Hop Options* (type 0, RFC 2460): Contains information that all Routers must observe on the way to the destination.
- *Routing* (type 43, RFC 2460): Influences the routing.
- *Fragment* (type 44, RFC 2460): Controls the reassembly of a fragmented packet.
- *Encapsulating Security Payload* (type 50, RFC 4303): Information for the encryption of the packet.
- *Authentication Header* (type 51, RFC 4302): Contains information to ensure the confidentiality of the packet.
- *No Next Header* (type 59, RFC 2460): Placeholder that indicates the end of the header stack.
- *Destination Options* (type 60, RFC 2460): Optional information that needs to be examined only by a packet's destination node(s).

Die Extension Header bieten eine hohe Flexibilität, da so auch nachträglich Erweiterungen in IPv6 implementiert werden können, die nicht von vornherein im Protokoll eingeplant waren und keine Änderung des eigentlichen IPv6-Headers erfordern. Ein weiterer Vorteil ist, dass es keine Größenbeschränkung für die Extension Header gibt. Die Größe jedes Extension Headers muss lediglich ein Vielfaches von 64 Bits (8 Bytes) sein.

Jeder Header verweist auf einen nachfolgenden Header. Dadurch entsteht eine Kette, bis der Header des Protokolls der Transportschicht erreicht ist. Jeder Extension Header trägt eine definierte Nummer. Einige Extension Header sind:

- *Hop-By-Hop Options* (Typ 0, RFC 2460): Enthält Informationen, die alle Router auf dem Weg zum Ziel beachten müssen.
- *Routing* (Typ 43, RFC 2460): Beeinflusst das Routing.
- *Fragment* (Typ 44, RFC 2460): Steuert das Zusammensetzen eines fragmentierten Pakets.
- *Encapsulating Security Payload* (Typ 50, RFC 4303): Daten zur Verschlüsselung des Pakets.
- *Authentication Header* (Typ 51, RFC 4302): Enthält Informationen, um die Vertraulichkeit des Pakets sicherzustellen.
- *No Next Header* (Typ 59, RFC 2460): Platzhalter, der das Ende des Header-Stapels anzeigt.
- *Destination Options* (Typ 60, RFC 2460): Optionale Informationen, die nur von dem bzw. den Zielknoten eines Pakets untersucht werden müssen.

7.3

Forwarding and Path Determination

The primary task of the Routers is the *Forwarding* of IP packets. To perform this task, the

Weiterleitung und Wegbestimmung

Die primäre Aufgabe der Router ist die *Weiterleitung* (*Forwarding*) der IP-Pakete. Um diese

Routers must determine the correct interface (port) for each incoming packet.

Each Router manages a local routing table containing the logical networks known to it. The table shows which network can be reached via which port. A Router therefore only has to send IP packets in the direction specified by the table.

Path determination (*Routing*) is the process by which the routing tables are created, using distributed algorithms to identify the best route to the destination, at the lowest cost. Routing is implemented using routing protocols that the Routers execute independently. The routing protocols base on distributed algorithms to achieve a better level of scalability.

Two general categories of routing protocols exist:

- *Distance vector routing protocols* which implement the *Bellman-Ford algorithm*.
- *Link-state routing protocols* which implement the *Dijkstra algorithm*.

One example for a distance vector protocol is the *Routing Information Protocol* (RIP). Examples of link state routing protocols include the *Border Gateway Protocol* (BGP) and *Open Shortest Path First* (OSPF).

Routing protocols can also be differentiated according to their area of application. Routers are organized in *Autonomous Systems* (AS). Each AS consists of a group of logical networks that use the Internet Protocol and are operated and managed by the same organization (e.g., an Internet Service Provider (ISP), a large corporation, or a university) and use the same routing protocol [23]. The interconnected Autonomous Systems together form the Internet. Each AS has a unique *Autonomous System Number* (ASN).

The operators of the Autonomous Systems are responsible for routing within Autonomous Systems, the so-called *Intra-AS-Routing*. Appropriate protocols for Intra-AS-Routing are, among others, RIP and OSPF.

Aufgabe zu erfüllen, müssen die Router für jedes eintreffende Paket die korrekte Schnittstelle (Port) ermitteln.

Jeder Router verwaltet eine lokale Routing-Tabelle mit den ihm bekannten logischen Netzen, aus der hervorgeht, über welchen Port welches Netz erreichbar ist. Ein Router muss die IP-Pakete also nur in die Richtung versenden, die die Tabelle vorgibt.

Die *Wegbestimmung* (*Routing*) ist der Prozess, bei dem die Weiterleitungstabellen mit Hilfe verteilter Algorithmen erstellt werden, um die Bestimmung des besten Weges, also zu den niedrigsten Kosten, zum Ziel zu ermöglichen. Das Routing wird durch Routing-Protokolle realisiert, die die Router selbständig ausführen. Dass die Routing-Protokolle auf verteilten Algorithmen basieren, liegt an der dadurch erreichbaren, höheren Skalierbarkeit.

Es existieren zwei Hauptklassen von Routing-Protokollen:

- *Distanzvektorprotokolle*, die den *Bellman-Ford-Algorithmus* verwenden.
- *Link-State-Routing-Protokolle*, die den *Dijkstra-Algorithmus* verwenden.

Ein Beispiel für ein Distanzvektorprotokoll ist das *Routing Information Protocol* (RIP). Beispiele für Link-State-Routing-Protokolle sind das *Border Gateway Protocol* (BGP) und *Open Shortest Path First* (OSPF).

Zudem unterscheidet man noch Routing-Protokolle anhand ihres Einsatzbereichs. Router sind in *autonomen Systemen* (AS) organisiert. Jedes AS besteht aus einer Gruppe von logischen Netzen, die das Internet Protocol verwenden und die von der gleichen Organisation (z. B. einem Internet Service Provider (ISP), einem großen Unternehmen oder einer Universität) betrieben und verwaltet werden sowie das gleiche Routing-Protokoll verwenden [23]. Die miteinander verbundenen autonomen Systeme bilden in ihrer Gesamtheit das Internet. Jedes AS verfügt eine eindeutige *Autonomous System Number* (ASN).

Für das Routing innerhalb autonomer Systeme, das sogenannte *Intra-AS-Routing*, sind die Betreiber der autonomen Systeme selbst verantwortlich. Geeignete Protokolle für Intra-AS-Routing sind u. a. RIP und OSPF.

For routing between the Autonomous Systems, the so-called *Inter-AS-Routing*, the routing protocol BGP is used. For further information on BGP, see RFC 1163 and RFC 4271.

Für das Routing zwischen den Autonomen Systeme, das sogenannte *Inter-AS-Routing*, wird meist das Routing-Protokoll BGP verwendet. Für weiterführende Informationen zum BGP, siehe RFC 1163 und RFC 4271.

7.4

Routing Information Protocol (RIP)

The *Routing Information Protocol* (RIP) is a *distance vector protocol* that enables routing inside Autonomous Systems. During initialization, each Router sends a *RIP Request* via all its ports using broadcast messages. A new Router requests all neighboring (reachable) Routers to transmit their routing tables in this way. With routing information from the incoming *RIP Response*, the Router fills its local routing table with records [11].

Every 30 seconds, each Router sends its routing table via the connectionless Transport Layer protocol UDP to its direct neighbors. By default, RIP uses port number 520. This periodic message is called *Advertisement*. If a Router receives a table, it checks whether entries in it are better than those currently stored in the routing table. If the received table contains a path that is more efficient, the Router updates the corresponding records in its routing table.

In addition to the periodic update message, a Router sends its routing table to its direct neighbors whenever it has been modified. These sporadic messages are called *Triggered Updates*.

The route costs (*metric*) to the destination network depend exclusively on the number of Routers which have to be passed on the route and are specified in *hops*. Each Router increases the value of the hops (*Hop count*) by one. The inaccessibility of a network is indicated by RIP with hop count 16 ($= \infty$ costs). This means that RIP only allows computer networks with a maximum length of 15 Routers.

Routing Information Protocol (RIP)

Das *Routing Information Protocol* (RIP) ist ein *Distanzvektorprotokoll*, das Routing innerhalb autonomer Systeme ermöglicht. Jeder Router sendet während der Initialisierung über alle seine Ports via Broadcast eine RIP-Anfrage (*RIP Request*). Ein neuer Router fordert damit alle benachbarten (erreichbaren) Router auf, ihre Routing-Tabellen zu übermitteln. Mit den Routing-Informationen der eintreffenden RIP-Antworten (*RIP Response*) füllt der Router seine lokale Routing-Tabelle mit Einträgen [11].

Alle 30 Sekunden sendet jeder Router seine Routing-Tabelle über das verbindungslose Transportprotokoll UDP an seine direkten Nachbarn. Standardmäßig verwendet RIP die Portnummer 520. Diese regelmäßige Nachricht heißt *Advertisement*. Empfängt ein Router eine Tabelle, überprüft er, ob Einträge darin besser sind, als die bislang in der Routing-Tabelle gespeicherten. Enthält die empfangene Tabelle günstigere Wege, aktualisiert der Router die entsprechenden Einträge in seiner lokalen Routing-Tabelle.

Zusätzlich zur periodischen Aktualisierungsnachricht sendet ein Router immer dann seine Routing-Tabelle an die direkten Nachbarn, wenn er eine Veränderung an seiner Routing-Tabelle vorgenommen hat. Diese unregelmäßigen Nachrichten heißen *Triggered Updates*.

Die Wegkosten (*Metrik*) zum Zielnetz hängen bei IP ausschließlich von der Anzahl der Router ab, die auf dem Weg passiert werden müssen und sind in *Hops* angegeben. Jeder Router erhöht den Wert der Hops (*Hopcount*) um eins. Die Unerreichbarkeit eines Netzwerks gibt RIP mit Hopcount 16 ($= \infty$ Kosten) an. RIP erlaubt also nur Computernetze mit einer maximalen Länge von 15 Routern.

7.4.1

Count-to-Infinity

A drawback of RIP is that incident notifications propagate slowly. The example in Figure 7.12 shows a network with three Routers. Table 7.5 contains the stored distance to Router R_A in the routing tables of R_A, R_B, and R_C after the respective advertisement rounds. After two advertising rounds, the state of convergence is reached.

Count-to-Infinity

Ein Nachteil von RIP ist, dass schlechte Nachrichten sich nur langsam verbreiten. Das Beispiel in Abbildung 7.12 zeigt ein Netzwerk mit drei Routern. Tabelle 7.5 enthält die gespeicherte Entfernung zu Router R_A in den Routing-Tabellen von R_A, R_B und R_C nach den jeweiligen Advertisement-Runden. Nach zwei Advertisement-Runden ist die Konvergenz erreicht.

Figure 7.12: A Network with three Routers and without Problems

The state of convergence is achieved when, after the network topology has changed, all Routers again have a unified view of the network. From this point on, the records in the local routing tables of the Routers are adjusted in such way that the change of the topology has been taken into account.

Der Zustand der Konvergenz ist dann erreicht, wenn nach einer Änderung der Netzwerk-Topologie, alle Router wieder eine einheitliche Sicht auf das Netzwerk haben. Ab diesem Zeitpunkt sind die Einträge in den lokalen Routing-Tabellen der Router dahingehend angepasst, dass die Änderung der Topologie berücksichtigt ist.

Table 7.5: Distance to Router R_A in the Routing Tables of R_A, R_b, and R_C

R_A	R_B	R_C	
0	∞	∞	Initial record
0	1	∞	After advertisement round 1
0	1	2	After advertisement round 2

However, if the connection to router R_A fails (see Figure 7.13), a *Count-to-Infinity* occurs. It takes 15 advertisement rounds until Router R_A is labeled as unreachable in the routing tables of the other Routers.

Table 7.6 again shows the stored distance to Router R_A in the routing tables of R_A, R_B and R_C after the respective advertisement rounds. In advertisement round 1, Router R_B no longer receives any information from Router R_A and now considers the path over Router R_C to be the best path to reach Router R_A. Router R_B, therefore, modifies the distance to Router R_A to value 3. In advertisement round 2, Router R_C is informed that its neighbor Router R_B has a route with distance 3 to Router R_A and

Fällt aber beispielsweise die Verbindung zu Router R_A aus (siehe Abbildung 7.13), kommt es zum sogenannten *Count-to-Infinity*. Es dauert 15 Advertisement-Runden, bis Router R_A in den Routing-Tabellen der übrigen Router als unerreichbar markiert ist.

Tabelle 7.6 enthält wieder die die gespeicherte Entfernung zu Router R_A in den Routing-Tabellen von R_A, R_B und R_C nach den jeweiligen Advertisement-Runden. Bei Advertisement-Runde 1 hört Router R_B nichts mehr von Router R_A und hält den Pfad über Router R_C nun am besten, um Router R_A zu erreichen. Router R_B ändert darum die Entfernung zu Router R_A auf den Wert 3. Bei Advertisement-Runde 2 erfährt Router R_C das sein Nachbar Router R_B eine Route mit der Länge 3 zu Router R_A hat

therefore adjusts the locally stored distance to Router R_A to value 4. In advertisement round 3, Router R_B is informed that its neighbor Router R_C has a route with distance 4 to Router R_A and sets the distance to Router R_A in its routing table to value 5, and so on.

und passt darum seine lokal gespeicherte Entfernung zu Router R_A auf 4 an. Bei Advertisement-Runde 3 erfährt Router R_B das sein Nachbar Router R_C eine Route mit der Länge 4 zu Router R_A hat und ändert in seiner Routing-Tabelle die Entfernung zu Router R_A auf den Wert 5 usw.

Figure 7.13: Scenario: The Link to Router R_A fails

Table 7.6: Distance to Router R_A in the Routing Tables of R_A, R_b, and R_C

R_A	R_B	R_C	
0	1	2	Initial record
0	3	2	After advertisement round 1
0	3	4	After advertisement round 2
0	5	4	After advertisement round 3
0	5	6	After advertisement round 4
0	7	6	After advertisement round 5
0	7	8	After advertisement round 6
0	9	8	After advertisement round 7
0	9	10	After advertisement round 8
0	11	10	After advertisement round 9
0	11	12	After advertisement round 10
0	13	12	After advertisement round 11
0	13	14	After advertisement round 12
0	15	14	After advertisement round 13
0	15	∞	After advertisement round 14
0	∞	∞	After advertisement round 15
			$\Longrightarrow$ Count-to-Infinity

7.4.2

Split Horizon

The example in Figure 7.13 and Table 7.6 shows that due to the Count-to-Infinity problem, it takes a long time to detect the inaccessibility of a route.

Advertisement messages are exchanged every 30 seconds. Without Triggered Updates, it can, therefore, take up to $15 * 30$ seconds = 7:30 minutes until a network failure between two Routers is detected and the affected Routes are marked as unreachable in the routing tables.

Split Horizon

Das Beispiel in Abbildung 7.13 und Tabelle 7.6 zeigt, dass wegen des Problems Count-to-Infinity viel Zeit verloren geht, bis die Unerreichbarkeit einer Route erkannt wird.

Advertisement-Nachrichten werden alle 30 Sekunden ausgetauscht. Ohne Triggered Updates kann es darum bis zu $15 * 30$ Sekunden = 7:30 Minuten dauern, bis ein Netzausfall zwischen zwei Routern erkannt wird und die betroffenen

One potential solution that improves the situation in some cases is *Split Horizon*, which prevents routing loops between *two routers*. A routing information must not be published via the port on which it was received. This prevents a Router from sending a route back to the Router from which it had learned it.

To enable Split Horizon, the routing table for each destination network must contain not only the number of hops and the address of the next Router (next hop) but also the information from which Router the information was learned.

To illustrate the principle of Split Horizon, we look again at the network with three Routers in Figure 7.12. Router R_C learned from Router R_B that Router R_A is reachable via him. If the connection to Router R_A fails, Router R_B sends Router R_C with the next update the information that Router R_A is unavailable. Router R_C adjusts its routing table and neither now nor in the future sends routing information for Router R_A to Router R_B.

Unfortunately, Split Horizon often fails in more complex network topologies than the one depicted in Figure 7.13, which again may result in a Count-to-Infinity.

Routen in den Routing-Tabellen als nicht erreichbar markiert sind.

Eine Lösungsmöglichkeit, die bei einigen Anwendungsfällen die Situation verbessert, ist *Split Horizon*, das Routing-Schleifen zwischen *zwei Routern* verhindert. Eine Routing-Informationen darf dabei nicht über den Port veröffentlicht werden, über den sie empfangen wurde. Das verhindert, das ein Router eine Route zurück an den Router übermittelt, von dem er sie gelernt hat.

Um Split Horizon zu ermöglichen, muss in der Routing-Tabelle für jedes Zielnetz nicht nur die Anzahl der Hops und die Adresse des nächsten Routers (nächster Hop) gespeichert werden, sondern auch die Information, von welchem Router die Informationen gelernt wurde.

Um das Prinzip von Split Horizon zu verdeutlichen, betrachten wir erneut in Abbildung 7.12 das Netzwerk mit drei Routern. Router R_C weiß von Router R_B, das Router R_A über ihn erreichbar ist. Fällt nun wie in Abbildung 7.13 zu sehen ist, die Verbindung zu Router R_A aus, sendet Router R_B beim nächsten Update an Router R_C die Information, dass Router R_A nicht erreichbar ist. Router R_C passt seine Routing-Tabelle an und sendet weder jetzt noch in Zukunft Routing-Informationen für Router R_A an Router R_B.

Leider versagt Split Horizon bei komplexeren Topologien als der in in Abbildung 7.13 häufig, was erneut den Count-to-Infinity zur Folge hat.

7.4.3

Conclusion on RIP

RIP version 1 (RFC 1058) was developed at a time when computer networks were relatively small-sized. For this reason, RIP version 1 only supports network classes and no subnets. When the protocol was developed, only in seldom scenarios were different transmission media with significant differences regarding connection quality and data rate used within a network. For this reason, using the metric hop count in modern computer networks often generates routes whose path is not optimal, because all network segments are weighted equally.

Fazit zu RIP

RIP Version 1 (RFC 1058) wurde zu einem Zeitpunkt entwickelt, als die Computernetze noch relativ klein waren. Aus diesem Grund unterstützt RIP Version 1 auch nur Netzklassen und keine Subnetze. Als das Protokoll entwickelt wurde, existierten innerhalb eines Netzes nur selten verschiedene Übertragungsmedien mit deutlichen Unterschieden bzgl. Verbindungsqualität und Übertragungsrate. Aus diesem Grund führt die Metrik Hopcount in modernen Computernetzen häufig zu Routen, deren Verlauf nicht optimal ist, weil alle Netzabschnitte werden gleich stark gewichtet.

Figure 7.14: If all Network Segments are weighted equally, the Metric Hop count often generates Routes which are not optimal

In the example in Figure 7.14, a path from Router R_A to Router R_D is to be determined. The path $R_A \rightarrow R_E \rightarrow R_D$ has a better hop count, but the path $R_A \rightarrow R_B \rightarrow R_C \rightarrow R_D$ leads across network segments with networking technologies that offer better performance.

Figure 7.15 shows the structure of RIPv1 messages (*Advertisements*). If the field *Command* contains the value 1, it is a *RIP Request*. This requests a Router to transmit its routing table. If the field contains the value 2, it is a *RIP Response*. This is how a Router transmits its routing table. For computer networks that use the Internet Protocol, the field *Address family* has the value 2.

RIP version 2 (RFC 2453) supports subnets and can distinguish between internal and external routes. Figure 7.16 shows the structure of RIPv2 messages. In the field *Route tag*, it is specified whether the route is internal or external.

Another version of the RIP protocol that supports IPv6 is *RIP next generation* – RIPng (RFC 2080).

With RIP, each Router only communicates with its direct neighbors. One advantage is the low overhead for network communication. A drawback is that compared to other routing protocols, it takes a long time to achieve convergence because updates spread slowly. Each Router is only aware of the records of its own routing table. No Router has an understand-

Im Beispiel in Abbildung 7.14 ist die Route von Router R_A zu Router R_D gesucht. Die Route $R_A \rightarrow R_E \rightarrow R_D$ hat den niedrigeren Hopcount, aber die Route $R_A \rightarrow R_B \rightarrow R_C \rightarrow R_D$ führt über Netzabschnitte mit leistungsfähigeren Vernetzungstechnologien.

Abbildung 7.15 zeigt den Aufbau von Nachrichten (*Advertisements*) von RIP Version 1. Enthält das Datenfeld *Kommando* den Wert 1, handelt es sich um eine RIP-Anfrage (*RIP Request*). Damit wird ein Router aufgefordert seine Routing-Tabelle zu übermitteln. Enthält das Datenfeld den Wert 2, handelt es sich um eine RIP-Antwort (*RIP Response*). Damit übermittelt ein Router seine Routing-Tabelle. Bei Computernetzen, die das Internet Protocol verwenden, hat das Datenfeld *Adressfamilie* den Wert 2.

RIP Version 2 (RFC 2453) unterstützt Subnetze und kann zwischen internen und externen Routen unterscheiden. Abbildung 7.16 zeigt den Aufbau von Nachrichten, von RIP Version 2. Im Datenfeld *Route Tag* ist festgelegt, ob es sich um eine interne oder externe Route handelt.

Eine weitere Version des RIP-Protokolls, die IPv6 unterstützt, ist *RIP next generation* – RIPng (RFC 2080).

Bei RIP kommuniziert jeder Router nur mit seinen direkten Nachbarn. Ein Vorteil dabei ist die geringe Belastung für das Netzwerk. Ein Nachteil ist, dass es im Vergleich zu anderen Routing-Protokollen lange dauert, bis der Zustand der Konvergenz erreicht ist, weil sich Aktualisierungen nur langsam verbreiten. Jeder Router kennt nur den Inhalt seiner eigenen Routing-Tabelle. Kein Router hat Kenntnis über

Figure 7.15: Structure of RIPv1 Messages

ing of the entire network topology inside the Autonomous System in which it is located.

die komplette Netzwerk-Topologie innerhalb des autonomen Systems, in dem er sich befindet.

7.5

Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) is a *Link-state routing protocol* which implements the algorithm *Shortest Path First* that was developed by Edsger Wybe Dijkstra. The algorithm is also known in computer science as *Dijkstra's algorithm*. It allows the calculation of the shortest path between a start node and all other nodes in an edge-weighted graph.

Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) ist ein *Link-State-Routing-Protokoll*, das den von Edsger Wybe Dijkstra entwickelten Algorithmus *Shortest Path First* verwendet. Der Algorithmus in der Informatik auch unter dem Namen *Dijkstra-Algorithmus* bekannt. Er ermöglicht die Berechnung des kürzesten Weges zwischen einem Start-

Figure 7.16: Structure of RIPv2 Messages

Just like RIP, the OSPF also enables routing inside Autonomous Systems. However, OSPF messages are transmitted directly in the payload field of IPv4 packets, i.e., without a Transport Layer protocol. The header of the IPv4 packet contains the value 89 in the field *Protocol ID*.

Due to the complexity of OSPF, only a general introduction to the protocol can be given. A detailed description can be found in RFC 2328.

knoten und allen anderen Knoten in einem kantengewichteten Graphen.

Genau wie RIP ermöglicht auch OSPF das Routing innerhalb autonomer Systeme. OSPF-Nachrichten werden aber direkt, also ohne ein Protokoll der Transportschicht, im Nutzdatenteil von IPv4-Paketen übertragen. Im Header des IPv4-Pakets steht im Datenfeld *Protokoll-ID* der Wert 89.

Wegen der Komplexität der Arbeitsweise von OSPF kann an dieser Stelle nur eine oberflächliche Beschreibung des Protokolls stattfinden. Eine detaillierte Beschreibung enthält RFC 2328.

7.5.1

Routing Hierarchy with OSPF

A considerable difference of OSPF compared to RIP is that in OSPF, routing hierarchies can be formed by subdividing Autonomous Systems into *areas*. Each area consists of a group of Routers and is invisible to the other areas of the Autonomous System. Each Router can be assigned to different areas. Two advantages that arise from having routing hierarchies are improved scalability and security [11, 23].

Four different types of Routers are distinguished in an OSPF routing hierarchy. These are Backbone Routers, Area Border Routers, Autonomous System Boundary Routers, and Internal Routers.

In any Autonomous System where OSPF is used, one or more Routers serve as *backbones*. In an OSPF hierarchy, there must always be precisely one *backbone area*. This is the connection between the areas of an Autonomous System. The Routers in the backbone, the *Backbone Routers*, forward the traffic between the areas.

For each area, there is an *Area Border Router*. It is located at the geographical boundary of the area to the backbone [20]. It forms the interface between the area and the backbone area.

Routers that are located at the border of an Autonomous System, i.e. that are connected to at least one other Autonomous System, are called *Autonomous System Boundary Routers*. These are responsible for exchanging routing information with other Autonomous Systems.

If a Router is not in the backbone area, it is an *Internal Router*.

Routing-Hierarchie mit OSPF

Ein großer Unterschied von OSPF gegenüber RIP ist, das mit OSPF Routing-Hierarchien gebildet werden können, indem autonome Systeme in Bereiche (*Areas*) unterteilt werden. Jede Area besteht aus einer Gruppe von Routern und ist für die übrigen Areas des autonomen Systems unsichtbar. Jeder Router kann verschiedenen Areas zugeordnet sein. Zwei Vorteile die sich aus Routing-Hierarchien ergeben, sind die bessere Skalierbarkeit und die höhere Sicherheit [11, 23].

Vier unterschiedliche Typen von Routern werden in einer OSPF-Routing-Hierarchie unterschieden. Dabei handelt es sich um Backbone Router, Area Border Router, Autonomous System Boundary Router und Internal Router.

In jedem autonomen System, in dem OSPF verwendet wird, erfüllen ein oder mehr Router die Aufgabe des *Backbones*. In einer OSPF-Hierarchie muss es immer exakt eine *Backbone Area* geben. Diese ist die Verbindung zwischen den Areas eines autonomen Systems. Die Router im Backbone, die *Backbone Router*, leiten den Datentransfer zwischen den Areas weiter.

Für jede Area gibt es einen *Area Border Router*. Dieser befindet sich an der geographischen Grenze der Area zum Backbone [20]. Er ist die Schnittstelle zwischen der Area und der Backbone Area.

Router, die sich an der Grenze eines autonomen Systems befinden, also mit mindestens einem weiteren autonomem System verbunden sind, heißen Grenzrouter (*Autonomous System Boundary Router*). Diese sind für den Austausch der Routing-Information mit anderen autonomen Systemen zuständig.

Befindet sich ein Router nicht in der Backbone Area, handelt es sich um einen internen Router (*Internal Router*).

7.5.2

Functioning of OSPF

As soon as a Router is started, it sends a so-called *Hello message* via broadcast and then receives the hello messages from the other Routers in the area. This tells a new Router about its neighbors and their connection status.

Arbeitsweise von OSPF

Sobald ein Router gestartet wurde, sendet er eine sogenannte *Hello-Nachricht* via Broadcast und empfängt daraufhin die Hello-Nachrichten der übrigen erreichbaren Router in der Area. So erfährt ein neuer Router, welche Nachbarn er

Figure 7.17: OSPF Routing Hierarchy inside an Autonomous System

When using OSPF, each Router creates a database with topology information about the network. At periodic intervals, the Routers send OSPF messages with their *link state information* via broadcast. This way, they flood the network. If the state of a link has changed, the affected Router immediately sends its updated link information via broadcast to the network.

With the help of the incoming link information and the resulting records in the local database with topology information, each Router calculates the shortest (cheapest) paths to the other Routers in its area. Each Router considers itself as root when calculating the shortest routes to the other Routers.

If there are multiple Routers, the Routers among themselves can select one *designated*

hat und wie der Verbindungsstatus zu diesen ist.

Bei OSPF erstellt sich jeder Router eine Datenbank mit Topologie-Informationen über das Netzwerk. Die Router senden in regelmäßigen Zeitintervallen OSPF-Nachrichten mit ihren Verbindungsinformationen (*Link State-Informationen*) via Broadcast und Fluten so das Netz. Ändert sich der Zustand einer Verbindung, sendet der betreffende Router sofort seine aktualisierten Verbindungsinformationen via Broadcast ins Netz.

Mit Hilfe der eintreffenden Verbindungsinformationen und der sich daraus ergebenden Einträge in der lokalen Datenbank mit Topologie-Informationen, berechnet jeder Router die kürzesten (günstigsten) Wege zu den übrigen Routern seiner Area. Jeder Router betrachtet sich selbst bei der Berechnung der kürzesten Wege zu den übrigen Routern als Wurzel.

Sind mehrere Router vorhanden, können die Router unter sich für jede Area einen *Designa-*

Router and one *backup designated Router* for each area. When selecting a designated Router, the decision criterion is the highest Router ID. If no Router ID has been configured to influence the selection process, the highest IP address assigned to the Router is the decision criterion.

All Routers of an area send their link status information to the designated Router of their Area. The designated Router of an area gathers the link state information and, distributes it to the other Routers in the area and forwards it to the designated Routers of other areas. The use of a designated Router is optional. If it exists, it has the advantage of reducing the network workload for distributing link information [11]. If a designated Router fails, the backup designated Router becomes the designated Router [20].

ted Router und einen *Backup Designated Router* wählen. Bei der Wahl eines Designated Routers ist das Entscheidungskriterium die höchste Router-ID. Wurde keine Router-ID konfiguriert, um in den Auswahlprozess einzugreifen, ist die höchste dem Router zugewiesene IP-Adresse das Entscheidungskriterium.

Alle Router einer Area senden ihre Verbindungsinformationen an den Designated Router ihrer Area. Der Designated Router einer Area sammelt die Verbindungsinformationen und verteilt sie an die übrigen Router in der Area und leitet sie an die Designated Router anderer Areas weiter. Die Verwendung eines Designated Routers ist optional. Wenn er existiert, hat das den Vorteil, dass sich die Netzwerkbelastung für das Verteilen der Verbindungsinformationen verringert [11]. Fällt ein Designated Router aus, übernimmt der Backup Designated Router die Rolle des Designated Routers [20].

7.5.3 Structure of OSPF Messages

OSPF specifies five different types of messages. One of them is the *hello message* that has already been mentioned, which is used to find the accessible Routers (neighbors) inside the area. With the so-called *database description message*, the Routers send their topology databases (i.e. routing tables) to keep the records in the local databases of the Routers within an area synchronized.

With a *link-state request message*, a Router can request one or more records of the topology database of another Router. The Routers respond with a *link-state update message* to such a request. Also, there is the so-called *link-state acknowledgment message* with which a Router acknowledges the reception of one or more link-state update messages.

The detailed structure of the different types of OSPF messages is not discussed here. However, all OSPF messages have a header which is the same for all message types. It is 24 bytes long (see Figure 7.18).

The field *Version* indicates the protocol version. If IPv4 is used, OSPF version 2 is used, and the field has the value 2. The *Type* field

Aufbau von OSPF-Nachrichten

OSPF definiert fünf verschiedene Typen von Nachrichten. Eine davon ist die bereits erwähnte *Hello-Nachricht*, um die erreichbaren Router (Nachbarn) in der Area aufzufinden. Mit der sogenannten *Database Description-Nachricht* versenden die Router ihre Topologie-Datenbanken (d.h. Routing-Tabellen), um die Einträge in den lokalen Datenbanken der Router innerhalb einer Area synchron zu halten.

Mit einer *Link State Request-Nachricht* kann ein Router einen oder mehrere Einträge der Topologie-Datenbank eines anderen Routers anfordern. Auf eine solche Anfrage antworten die Router mit einer *Link State Update-Nachricht*. Zudem existiert noch die sogenannte *Link State Acknowledgment-Nachricht*, mit der ein Router den Empfang einer oder mehrerer Link State Update-Nachrichten bestätigt.

Auf den detaillierten Aufbau der verschiedenen Typen von OSPF-Nachrichten wird hier nicht eingegangen. Alle OSPF-Nachrichten haben aber einen Header, der bei allen Nachrichtentypen gleich ist und der 24 Bytes lang ist (siehe Abbildung 7.18).

Das Datenfeld *Version* gibt die Protokollversion an. Wird IPv4 verwendet, wird OSPF Version 2 verwendet und das Datenfeld enthält

Figure 7.18: Structure of the Header of OSPF Messages

contains the type number that specifies which of the five possible types of OSPF messages this message belongs to. An overview of the type numbers and the associated OSPF message types is shown in Table 7.7.

den Wert 2. Im Datenfeld *Typ* befindet sich die Typ-Nummer, die festgelegt, zu welcher der fünf möglichen Typen von OSPF-Nachrichten, diese Nachricht gehört. Eine Übersicht über die Typ-Nummern und die zugehörigen OSPF-Nachrichten-Typen zeigt Tabelle 7.7.

Table 7.7: Different Types of OSPF Messages

Type number	Message type
1	Hello message
2	Database description message
3	Link-state request message
4	Link-state update message
5	Link-State acknowledgment message

The length of the OSPF message in bytes is specified in the *Length* field. The *Router ID* field contains the ID of the Router that sent the OSPF2 message. Usually, the Router ID is identical to the IP address. The ID of the area inside the Autonomous System is specified in the field *Area ID*. The area ID usually corresponds to the subnet number. If the field has the value 0, it is the backbone of the Autonomous System.

Die Länge der OSPF-Nachricht in Bytes ist im Datenfeld *Länge* angegeben. Das Datenfeld *Router-ID* enthält die ID des Routers, der die OSPF2-Nachricht gesendet hat. Üblicherweise ist die Router-ID identisch mit der IP-Adresse. Die ID der Area innerhalb des autonomen Systems ist im Datenfeld *Area-ID* definiert. Die Area-ID entspricht üblicherweise der Subnetznummer. Hat das Datenfeld den Wert 0, handelt es sich um das Backbone des autonomen Systems.

The *Checksum* is calculated over all fields of the OSPF message except the *Authentication* field. The authentication method is specified in the field *Authentication Type*. If this field contains the value 0, no authentication is used. If it contains the value 1, password-based authentication is used. The password can have

Die *Prüfsumme* wird über alle Datenfelder der OSPF-Nachricht, mit Ausnahme von *Authentifizierung*, gebildet. Die Art und Weise der Authentifizierung ist im Datenfeld *Authentifizierungsschema* festgelegt. Enthält dieses Datenfeld den Wert 0, wird keine Authentifizierung verwendet. Enthält es den Wert 1, wird eine Passwort-basierte Authentifizierung verwendet.

a length of up to 8 bytes and is stored in the *Authentication* field.

Das Passwort kann maximal 8 Bytes lang sein und ist im Datenfeld *Authentifizierung* abgelegt.

7.5.4

Conclusion on OSPF

While in RIP the Routers only communicate with their direct neighbors, in OSPF all Routers communicate with each other. Each Router builds a complex database of topology information. Flooding enables the protocol to respond quickly to topology changes and node failures. One advantage is that with OSPF, the state of convergence is reached after a short time because updates spread fast. However, flooding the network also causes more overhead in network communication.

While RIP does not support hierarchies, OSPF allows routing hierarchies to be created using areas. These improve security and scalability.

In contrast to RIP, OSPF assigns different weights to the network segments. With OSPF, the decision criterion that determines whether a route is favorable or not is not the number of Routers (*hops*) that have to be passed on the path to the destination, but the path costs. The *path costs* result from the data rate of the networks on the path to the destination. A higher data rate leads to lower path costs. RFC 2328 does not explicitly specify how the path costs are calculated. Network components by Cisco calculate the path costs of a route with the following equation, which assumes a reference data rate of 100 Mbit/s:

Fazit zu OSPF

Während bei RIP die Router nur mit ihren direkten Nachbarn kommunizieren, kommunizieren bei OSPF alle Router untereinander. Jeder Router baut eine komplexe Datenbank mit Topologie-Informationen auf. Durch das Fluten reagiert das Protokoll rasch auf Topologieänderungen und Knotenausfälle. Ein Vorteil ist, dass bei OSPF nach kurzer Zeit der Zustand der Konvergenz erreicht ist, weil sich Aktualisierungen rasch verbreiten. Das Fluten des Netzwerks verursacht aber auch mehr Overhead.

Während RIP keine Hierarchien unterstützt, können bei OSPF mit Hilfe von Areas Routing-Hierarchien gebildet werden. Diese verbessern die Sicherheit und Skalierbarkeit.

Im Gegensatz zu RIP werden bei OSPF die Netzabschnitte unterschiedlichen stark gewichtet. Bei OSPF ist das Entscheidungskriterium für die Vorteilhaftigkeit eines Weges nicht die Anzahl der Router (*Hops*), die auf dem Weg zum Ziel passiert werden müssen, sondern die Pfadkosten. Die *Pfadkosten* ergeben sich aus der Datenrate der Netze auf dem Weg zum Ziel. Eine höhere Datenrate führt zu niedrigeren Pfadkosten. RFC 2328 legt nicht explizit fest, wie die Pfadkosten berechnet werden. Die Firma Cisco berechnet in ihren Produkten die Pfadkosten einer Wegstrecke mit folgender Formel, die eine Referenz-Datenrate von 100 Mbit/s annimmt:

$$\text{Path cost} = \frac{100,000,000}{\text{Data rate [bits per second]}}$$

The calculation of path costs using the Cisco equation has a severe drawback. For Fast Ethernet with a data rate of 100 Mbit/s and modern networking technologies, the path costs always have the value 1. Therefore, a differentiation of the path costs is no longer possible with modern networking technologies (see Table 7.8).

Die Berechnung der Pfadkosten mit der Formel von Cisco hat einen gravierenden Nachteil. Bei Fast-Ethernet mit einer Datenrate von 100 Mbit/s und modernen Vernetzungstechnologien haben die Pfadkosten immer den Wert 1. Somit ist bei modernen Vernetzungstechnologien eine Abstufung der Pfadkosten nicht mehr möglich (siehe Tabelle 7.8).

Table 7.8: Path Costs of OSPF

Networking technology	Data rate	Path costs
Modem	56 kbit/s	1,785
ISDN	64 kbit/s	1,562
Token Ring	4 Mbit/s	25
Ethernet	10 Mbit/s	10
Token Ring	16 Mbit/s	6
Fast Ethernet	100 Mbit/s	1
Gigabit Ethernet	1,000 Mbit/s	1
10-Gigabit Ethernet	10,000 Mbit/s	1

7.6

Internetworking

The term *Internetworking* describes the communication between network devices over protocols of the Data Link Layer and the Network Layer via networks, which can be based on different networking technologies.

Figure 7.19 shows a plausible scenario for Internetworking. If terminal device X wants to send an IP packet to terminal device Y, it must be aware of its logical address (IP address). The physical address (MAC address) is also required for forwarding packets as payload of frames on the Data Link Layer.

Netzübergreifende Kommunikation

Internetworking heißt die Kommunikation zwischen Netzwerkgeräten über Protokolle der Sicherungsschicht und der Vermittlungsschicht über Netze, die auf unterschiedlichen Vernetzungstechnologien basieren können.

Abbildung 7.19 zeigt ein denkbares Szenario für Internetworking. Will Endgerät X ein IP-Paket an Endgerät Y senden, muss es dafür dessen logische Adresse (IP-Adresse) kennen. Für die Weiterleitung auf der Sicherungsschicht ist zudem die physische Adresse (MAC-Adresse) nötig.

Figure 7.19: Communication between Network Devices via different Networks (*Internetworking*)

Terminal device X calculates the subnet number of its network and the subnet number of terminal device Y by performing the `AND` operation of the netmask in its local network configuration both with its IP address and the IP address of terminal device Y. If the subnet numbers are identical, both communication partners are in the same logical subnet. A logical subnet covers at least one physical network and can only be connected to a single interface of a Router. If the subnet numbers are different, sender and receiver are located in different physical networks, just as in the example shown in Figure 7.19.

If two communication partners are located in the same physical and logical network, the sender can find out the MAC address of the receiver by using address resolution via ARP (see section 6.10). Because this is a communication across network boundaries, that is not sufficient, because ARP is only responsible for the resolution of the MAC addresses in the local physical network. The reason for this is that ARP requests are sent in frames on the Data Link Layer where the field for the destination address contains the broadcast address. However, such frames are not forwarded by Bridges and Switches (see section 6.2.1) and therefore ARP does not provide cross-network address resolution.

Figure 7.19 shows that the terminal devices X and Y are in two different physical and logical networks, which are connected by a Router or Layer-3-Switch. Both physical networks A and B may be based on different networking technologies.

To send the IP packet that was created on the Network Layer, to terminal device Y, which is known to be located in a different network, terminal X must specify its own MAC address as the source and the MAC address of the Router as the destination on the Data Link Layer. Terminal device X is aware of the IP address of the Router because it is configured as *Default Gateway*. Terminal device X learns the MAC address of the Router through address resolution with ARP.

The payload field of the frame contains the IP packet for terminal device Y with the IP

Endgerät X berechnet durch die `AND`-Verknüpfung der Netzmaske in seiner lokalen Netzwerkkonfiguration jeweils mit seiner eigenen IP-Adresse und der IP-Adresse von Endgerät Y die Subnetznummer des eigenen Netzes und die Subnetznummer von Endgerät Y. Sind die Subnetznummern identisch, befinden sich beide Kommunikationspartner im gleichen logischen Subnetz. Ein logisches Subnetz deckt mindestens ein physisches Netz ab und kann immer nur mit einer Schnittstelle eines Routers verbunden sein. Sind die Subnetznummern unterschiedlich, befinden sich Sender und Empfänger wie auch im Beispiel in Abbildung 7.19, in verschiedenen physischen Netzen.

Befinden sich zwei Kommunikationspartner im gleichen physischen und logischen Netz, kann der Sender via Adressauflösung mit ARP (siehe Abschnitt 6.10) die MAC-Adresse des Empfängers erfahren. Weil es sich hier aber um Kommunikation über Netzgrenzen hinweg handelt, ist das nicht ausreichend, denn ARP ist nur für die Auflösung der MAC-Adressen im lokalen physischen Netz zuständig. Der Grund dafür ist, dass ARP-Anfragen in Rahmen auf der Sicherungsschicht gesendet werden, bei denen das Datenfeld für die Zieladresse die Broadcast-Adresse enthält. Solche Rahmen werden von Bridges und Switches aber nicht weitergeleitet (siehe Abschnitt 6.2.1) und darum ist mit ARP keine netzübergreifende Adressauflösung möglich.

In Abbildung 7.19 ist zu sehen, dass sich die Endgeräte X und Y in zwei unterschiedlichen physischen und logischen Netzen befinden, die über einen Router oder Layer-3-Switch miteinander verbunden sind. Die beiden physischen Netze A und B können auf unterschiedlichen Vernetzungstechnologien basieren.

Um das auf der Vermittlungsschicht erzeugte IP-Paket an Endgerät Y, das sich ja bekannterweise in einem anderen Netz befindet, zu senden, muss Endgerät X im Rahmen auf der Sicherungsschicht seine eigene MAC-Adresse als Quelle und die MAC-Adresse des Routers als Ziel angeben. Endgerät X kennt die IP-Adresse des Routers, weil sie in der lokalen Netzwerkkonfiguration als *Default Gateway* eintragen ist und durch die Adressauflösung via ARP erfährt Endgerät X die MAC-Adresse des Routers.

Im Nutzdatenteil des Rahmens befindet sich das IP-Paket für Endgerät Y mit der IP-Adresse

address of terminal device X as source and the IP address of terminal device Y as destination.

The Router receives the IP packet and uses its local routing table, which contains all known logical networks, to identify the correct interface for forwarding the packet.

Since the Router is connected to the physical network by one of its interfaces, via which terminal device Y can also be reached, the Router determines the MAC address of terminal device Y via address resolution with ARP. The Router then inserts the IP packet to a frame where the MAC address of the Router is the source, and the MAC address of terminal device Y is the destination.

If the maximum packet length (*Maximum Transmission Unit*) of network B is smaller than the one of network A, it may be necessary for the Router to *fragment* the received packet into several smaller packets, depending on the size of the IP packet to be forwarded (see section 7.2.5).

The IP addresses of the sender (terminal device X) and the receiver (terminal device Y) in the IP packet are not modified during forwarding by the Routers.

von Endgerät X als Quelle und der IP-Adresse von Endgerät Y als Ziel.

Der Router empfängt das IP-Paket und ermittelt mit seiner lokalen Routing-Tabelle, die alle ihm bekannten logischen Netze enthält, die korrekte Schnittstelle für die Weiterleitung des Pakets.

Da der Router über eine seiner Schnittstellen mit dem physischen Netz verbunden ist, über das auch Endgerät Y erreichbar ist, ermittelt der Router die MAC-Adresse von Endgerät Y via Adressauflösung mit ARP. Daraufhin verpackt der Router das IP-Paket in einem Rahmen bei dem die MAC-Adresse des Routers als Quelle und die MAC-Adresse von Endgerät Y als Ziel angegeben ist.

Ist die maximale Paketlänge (*Maximum Transmission Unit*) von Netz B kleiner als das bei Netz A der Fall ist, kann es je abhängig von der Größe des weiterzuleitenden IP-Pakets nötig sein, dass der Router das empfangene Paket in mehrere kleinere Pakete *fragmentiert* (siehe Abschnitt 7.2.5).

Die IP-Adressen des Senders (Endgerät X) und des Empfängers (Endgerät Y) im IP-Paket werden bei der Weiterleitung durch die Router nicht verändert.

7.7

Network Address Translation

Almost no household or company and even very few educational and research institutions have enough public IPv4 addresses to equip all of network devices (e.g., PCs, workstations, servers, IP telephones, WLAN access points, WLAN repeaters, network printers, IP cameras, game consoles, smart TVs, file servers – network attached storage) with their IP addresses. For this reason, LANs usually use a private IPv4 address space (see section 7.2.3).

Since private IP addresses (seen from a global perspective) cannot be assigned to individual network devices, packets from networks with private IP address ranges must be forwarded to the Internet by Routers. In order for net-

Network Address Translation

Fast kein Haushalt oder Unternehmen und auch die allerwenigsten Bildungs- und Forschungseinrichtungen haben so viele öffentlich erreichbare IPv4-Adressen zur Verfügung, um alle ihre mit dem lokalen Netzwerk verbundenen Geräte (PCs, Workstations, Server, IP-Telefone, WLAN-Basisstationen – Access Points, WLAN-Repeater, Netzwerkdrucker, IP-Kameras, Spielkonsolen, Smart-TVs, Dateiserver – Network Attached Storage, etc.) mit eigenen Adressen zu versehen. Aus diesem Grund verwenden lokale Netze in der Regel einen privaten IPv4-Adressraum (siehe Abschnitt 7.2.3).

Da die privaten IP-Adressen (global gesehen) nicht einzelnen Netzwerkgeräten zugeordnet werden können, müssen Pakete aus Netzen mit privaten IP-Adressbereichen von Routern ins Internet weitergeleitet werden. Damit Netz-

work devices in private networks to communicate with network devices with globally accessible addresses, Routers must be able to identify themselves as the source of the IP packets they forward from directly connected private networks to the Internet, and at the same time deliver the incoming replies to the individual nodes in the directly connected private networks.

This technique is called *Network Address Translation* – NAT (RFC 2663 and RFC 3022), and it implements the automatic replacement of addresses in IP packets.

Figure 7.20 shows the functional principle of Network Address Translation. The clients X, Y, and Z are located in a network with a private IP address range. Only the Router has a globally accessible IP address. To the outside world, it does not appear as a Router, but like a network device with a single publicly registered IP address [23].

For example, if client X requests a web page from a server, the request (message 1) contains the IP address and port number of Client X as source addresses and the IP address and port number of the server as destination addresses. The Router, via which the local network (LAN) is connected to the global Internet (WAN), replaces the IP address and port number of the client with its IP address and port number in the forwarded request (message 2).

The Router stores the mappings between the ports of the Router and the corresponding network devices in the local network in a *NAT translation table*.

The reply from the server (message 3) is addressed to the Router. It replaces the address information according to the translation table and forwards the reply to client X (message 4).

With IPv6, a technology like NAT is unnecessary in principle, since the address space of IPv6 is large enough to assign globally accessible addresses to all network devices. Whether this is advisable for security reasons, however, is controversial. Although NAT was not developed as a dedicated security feature, in practice it helps to improve the network security by hiding the topology of the local network from the

werkgeräte in privaten Netzen trotzdem auch mit Netzwerkgeräten mit global erreichbaren Adressen kommunizieren können, müssen Router in der Lage sein, sich selbst als Quelle derjenigen IP-Pakete auszugeben, die sie aus direkt verbundenen privaten Netz in das Internet weiterleiten und gleichzeitig müssen Sie die eintreffenden Antworten den jeweiligen Teilnehmern in den direkt verbundenen privaten Netzen zustellen.

Dieses Verfahren heißt *Network Address Translation* – NAT (RFC 2663 und RFC 3022) und es realisiert die automatisierte Ersetzung von Adressen in IP-Paketen.

Abbildung 7.20 zeigt das Funktionsprinzip der Network Address Translation. Die Clients X, Y und Z befinden sich in einem Netz mit einem privaten IP-Adressbereich. Nur der Router hat eine global erreichbare IP-Adresse. Er wirkt für die Außenwelt nicht wie ein Router, sondern wie ein Netzwerkgerät mit einer einzelnen öffentlich registrierten IP-Adresse [23].

Fordert beispielsweise Client X eine Webseite von einem Server an, enthält die Anfrage (Nachricht 1) als Quelladressen die IP-Adresse und Portnummer von Client X und als Zieladressen die IP-Adresse und Portnummer des Servers. Der Router, über den das lokale Netz (LAN) mit dem globalen Internet (WAN) angebunden ist, ersetzt in der weitergeleiteten Anfrage (Nachricht 2) die IP-Adresse und Portnummer des Clients durch seine eigene IP-Adresse und Portnummer.

Die Zuordnungen zwischen den Ports des Routers und den zugehörigen Netzwerkgeräten im lokalen Netz speichert der Router in einer *NAT-Übersetzungstabelle* (*NAT Translation Table*).

Die Antwort des Servers (Nachricht 3) ist an den Router adressiert. Dieser ersetzt die Adressinformationen entsprechend der Übersetzungstabelle und leitet die Antwort an Client X weiter (Nachricht 4).

Bei IPv6 ist eine Technologie wie NAT prinzipiell unnötig, da der Adressraum von IPv6 groß genug ist, um allen Netzwerkgeräten global erreichbare Adressen zuzuweisen. Ob das aus Gründen der Sicherheit allerdings ratsam ist, ist umstritten. NAT wurde zwar nicht dediziert als Sicherheitsfeature entwickelt, verbessert aber in der Praxis die Netzwerksicherheit dadurch, dass es die Topologie des lokalen Netzes vor der

Figure 7.20: Functional Principle of Network Address Translation (NAT)

outside world. NAT in connection with IPv6 is called *IPv6-to-IPv6 Network Address Translation* (NAT66).

Außenwelt verbirgt. NAT im Zusammenhang mit IPv6 heißt *IPv6-to-IPv6 Network Address Translation* (NAT66).

7.8

Diagnostic and Error Messages via ICMP

The exchange of diagnostic and error messages is possible via the *Internet Control Message Protocol* (ICMP). All Routers and terminal devices can use ICMP. A typical situation for the application of ICMP is when the destination of an IP packet is unreachable because Time To Live (TTL) has expired.

Except for the echo function, an ICMP packet can never trigger another ICMP packet. If an ICMP packet cannot be delivered, no further action is taken. One application that sends ICMP packets is the `ping` program. ICMP defines various information messages that a Router can send in return.

Diagnose und Fehlermeldungen mit ICMP

Der Austausch von Diagnose- und Fehlermeldungen ist über das *Internet Control Message Protocol* (ICMP) möglich. Alle Router und Endgeräte können mit ICMP umgehen. Eine typische Situation, wo ICMP zum Einsatz kommt, ist wenn das Ziel eines IP-Pakets unerreichbar ist, weil die Time To Live (TTL) abgelaufen ist.

Mit Ausnahme der Echo-Funktion kann ein ICMP-Paket niemals ein anderes ICMP-Paket auslösen. Kann ein ICMP-Paket nicht zugestellt werden, wird nichts weiter unternommen. Eine Anwendung, die ICMP-Pakete versendet, ist das Programm `ping`. ICMP definiert verschiedene Informationsnachrichten, die ein Router zurücksenden kann.

Figure 7.21: Fields of ICMP Messages

ICMP messages are transmitted in the payload field of IPv4 packets. The header of the IPv4 packet has the value `1` in the field *Protocol ID*.

The field `Type` in the ICMP header specifies the message type. That is the class to which the ICMP message belongs. The field `Code` specifies the type of message within a message type (see Figure 7.21). Table 7.9 contains some combinations of message type and message code.

One application example for ICMP is the `traceroute` tool, which determines which Routers are used to forward IP packets to their destination. The sender sends an IP packet to the receiver with `TTL=1`. Router A receives the IP packet, sets `TTL=0`, discards the IP packet

ICMP-Nachrichten werden im Nutzdatenteil von IPv4-Paketen übertragen. Im Header des IPv4-Pakets steht im Datenfeld *Protokoll-ID* der Wert `1`.

Das Datenfeld `Typ` im ICMP-Header gibt den Nachrichtentyp an, also die Klasse zu der die ICMP-Nachricht gehört. Das Datenfeld `Code` spezifiziert die Art der Nachricht innerhalb eines Nachrichtentyps (siehe Abbildung 7.21). Tabelle 7.9 enthält einige Nachrichtentyp-Code-Kombinationen.

Ein Anwendungsbeispiel für ICMP ist das Werkzeug `traceroute`, das ermittelt, über welche Router IP-Pakete bis zum Ziel vermittelt werden. Der Sender schickt ein IP-Paket an den Empfänger mit `TTL=1`. Router A empfängt das IP-Paket, setzt `TTL=0`, verwirft das IP-Paket

Table 7.9: Selection of Combinations of Message Type and Message Code of ICMP

Type	Name of type	Code	Description
0	Echo reply	0	Echo reply (reply for `ping`)
3	Dest. unreachable	0	Destination network unreachable
		1	Destination host unreachable
		2	Destination protocol unreachable
		3	Destination port unreachable
		4	Fragmentation required but forbidden by the IP packet's flags
		13	Firewall at destination rejects the IP packet
4	Source quench	0	Receive buffer is full, IP packet discarded (congestion control)
8	Echo request	0	Echo request (`ping`)
11	Time exceeded	0	TTL (Time To Live) expired
17	Address mask req.	0	Request the number of bits in the subnet mask
18	Address mask reply	0	Reply to message type 17
		1	Time limit exceeded during defragmentation
30	Traceroute	0	Determine the route to the destination

Figure 7.22: Function Principle of `traceroute` via ICMP

and sends an ICMP message of the message type `11` and code `0` to the sender. The sender then sends an IP packet to the receiver with `TTL=2`. Router A forwards the IP packet, and the value of `TTL` is decremented. Router B receives the IP packet, sets `TTL=0`, discards the IP packet and sends an ICMP message of message type `11` and code `0` to the sender (see Figure 7.22).

Once the value of `TTL` is large enough for the packet to reach the receiver, it sends an ICMP message of the message type `3` and code `3` to the sender. In this way, the path from sender to receiver can be traced via ICMP.

und sendet eine ICMP-Nachricht vom Nachrichtentyp `11` und Code `0` an den Sender. Daraufhin schickt der Sender ein IP-Paket an den Empfänger mit `TTL=2`. Das IP-Paket wird von Router A weitergeleitet und dabei wird auch der Wert von `TTL` dekrementiert. Router B empfängt das IP-Paket, setzt `TTL=0`, verwirft das IP-Paket und sendet eine ICMP-Nachricht vom Nachrichtentyp `11` und Code `0` an den Sender (siehe Abbildung 7.22).

Sobald der Wert von `TTL` groß genug ist, dass das Paket den Empfänger erreicht, sendet dieser eine ICMP-Nachricht vom Nachrichtentyp `3` und Code `3` an den Sender. Auf dieses Art und Weise kann via ICMP der Weg vom Sender zum Empfänger nachvollzogen werden.

8

Transport Layer

The first three layers of the OSI reference model and the hybrid reference model contain technologies for connecting computers and exchanging data. The terminal devices can communicate with these, but protocols for interprocess communication are still missing. The provision of end-to-end protocols for interprocess communication is the task of the Transport Layer.

In this layer, the data of the Application Layer are packed into *Segments* at the sender, and the processes are addressed with port numbers. The Transport Layer at the receiver identifies the segments in the packets of the Network Layer.

Transportschicht

Die ersten drei Schichten des OSI-Referenzmodells und des hybriden Referenzmodells enthalten Technologien, um Computer zu verbinden und Daten auszutauschen. Mit diesen können die Endgeräte kommunizieren, aber es fehlen noch Protokolle zur Interprozesskommunikation. Die Bereitstellung der Ende-zu-Ende-Protokolle für Interprozesskommunikation ist die Aufgabe der Transportschicht.

In dieser Schicht werden beim Sender die Daten der Anwendungsschicht in *Segmente* verpackt und es findet die Adressierung der Prozesse mit Portnummern statt. Beim Empfänger erkennt die Transportschicht die Segmente in den Paketen der Vermittlungsschicht.

8.1

Characteristics of Transport Layer Protocols

The Network Layer protocol IP works connectionless. Each IP packet is routed to its destination independently of other packets. This method generates little overhead but leads to various problems from the Transport Layer point of view. IP packets may get lost or discarded because the TTL (Time To Live) has expired. IP packets often reach their destination in the wrong order. It is also possible that multiple copies of IP packets reach the destination. One of the reasons for this is that large networks are not static. Their infrastructure changes. Transmission media are exposed to external influences and can fail. Furthermore, the network utilization and thus the delay of the networks is fluctuating.

Eigenschaften von Transportprotokollen

Das Protokoll IP auf der Vermittlungsschicht arbeitet verbindungslos. Jedes IP-Paket wird unabhängig von anderen Paketen an sein Ziel vermittelt (*geroutet*). Dieses Vorgehen erzeugt wenig Overhead, führt aber aus Sicht der Transportschicht zu diversen Problemen. IP-Pakete gehen eventuell verloren oder werden verworfen, weil die TTL (Time To Live) abgelaufen ist. IP-Pakete erreichen ihr Ziel auch häufig in der falschen Reihenfolge. Es ist auch möglich, dass mehrere Kopien von IP-Paketen das Ziel erreichen. Gründe sind unter anderem, dass große Netze nicht statisch sind. Ihre Infrastruktur ändert sich. Übertragungsmedien sind äußeren Einflüssen ausgesetzt und können ausfallen. Auch

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_8

These problems are common in computer networks and, depending on the application, Transport Layer protocols have to compensate for these drawbacks. For many applications, the required characteristics of Transport Layer protocols are, among others:

- Guaranteed data transmission.
- Maintaining the correct order of the data.
- Supporting data transmissions of any size.
- The sender should be able to control the flow of data, i.e., the data rate.

This means that Transport Layer protocols are needed that translate the negative characteristics of the physical and logical networks into the (positive) characteristics that are expected from network applications of the Application Layer.

The two most frequently used Transport Layer protocols are UDP and TCP, and addressing is done in the Transport Layer with Sockets.

die Auslastung und damit die Verzögerung der Netze ist Schwankungen unterworfen.

Diese Probleme sind in Computernetzen alltäglich und je nach Anwendung müssen Transportprotokolle diese Nachteile ausgleichen. Gewünschte Eigenschaften von Transportprotokollen aus Sicht vieler Anwendungen sind unter anderem:

- Garantierte Datenübertragung.
- Einhaltung der korrekten Reihenfolge der Daten.
- Unterstützung beliebig großer Datenübertragungen.
- Der Sender soll in der Lage sein, den Datenfluss, also die Übertragungsrate zu kontrollieren.

Es sind also Transportprotokolle nötig, die die negativen Eigenschaften der physischen und logischen Netze in die (positiven) Eigenschaften umwandeln, die von den Netzwerkanwendungen der Anwendungsschicht erwartet werden.

Die beiden am häufigsten verwendeten Transportprotokolle sind UDP und TCP und die Adressierung erfolgt in der Transportschicht mit Sockets.

8.2

Addressing in the Transport Layer

Every application that uses TCP or UDP has a *port number*. This indicates which service is addressed on the terminal device. With TCP and UDP, the port number is 16 bits long and thus has a value between 0 and 65,535. In principle, the port numbers can be assigned arbitrarily. However, conventions exist which define the mapping of standard application to ports. Table 8.1 contains a selection of popular port numbers.

Different applications can use identical port numbers in the operating system at the same time if they communicate using different Transport Layer protocols. Also, some applications

Adressierung in der Transportschicht

Jede Anwendung, die TCP oder UDP nutzt, hat eine *Portnummer*. Diese gibt an, welcher Dienst auf dem Endgerät angesprochen wird. Bei TCP und UDP ist die Portnummer 16 Bits lang und liegt somit im Wertebereich 0 bis 65.535. Die Portnummern können im Prinzip beliebig vergeben werden. Es gibt aber Konventionen, welche Standardanwendungen welche Ports nutzen. Tabelle 8.1 enthält eine kleine Auswahl bekannter Portnummern.

Verschiedene Anwendungen können im Betriebssystem gleichzeitig identische Portnummern verwenden, wenn Sie über unterschiedliche Transportprotokolle kommunizieren. Zudem

Table 8.1: Selection of popular Port Numbers (*Well-known Ports*)

Port number	Service	Description
21	FTP	File transfer
22	SSH	Encrypted terminal emulation (secure shell)
23	Telnet	Terminal emulation for remote control of computers
25	SMTP	Email transfer
53	DNS	Resolution of domain names into IP addresses
67	DHCP	Assignment of the network configuration to clients
80	HTTP	Webserver
110	POP3	Client access to email server
143	IMAP	Client access to email server
443	HTTPS	Webserver (encrypted)
993	IMAPS	Client access to email server (encrypted)
995	POP3S	Client access to email server (encrypted)

implement communication via TCP and UDP using a single port number. An example of this is the Domain Name System – DNS (see section 9.1) [33].

The port numbers are subdivided into three groups. The ports from 0 to 1,023 are the *well-known ports*. These are assigned to applications and generally known. The ports from 1,024 to 49,151 are called *registered ports*. Application developers can register port numbers in this range for their applications. The port numbers from 49,152 to 65,535 are called *private ports*, they are not registered and can be used freely. The well-known ports and the registered ports are assigned by the Internet Assigned Numbers Authority (IANA).

In Linux/UNIX, the mapping of applications (services) to port numbers is specified in the file `/etc/services`. In the Windows operating system, for this purpose the following file exists:

gibt es Anwendungen, die Kommunikation via TCP und UDP über eine einzige Portnummer realisieren. Ein Beispiel hierfür ist das Domain Name System – DNS (siehe Abschnitt 9.1) [33].

Die Portnummern sind in drei Gruppen unterteilt. Die Ports 0 bis 1.023 sind die *Well-known Ports*. Diese sind Anwendungen fest zugeordnet und allgemein bekannt. Die Ports 1.024 bis 49.151 heißen *Registered Ports*. Anwendungsentwickler können sich Portnummern in diesem Bereich für eigene Anwendungen registrieren. Die Portnummern 49.152 bis 65.535 heißen *Private Ports*, sind nicht registriert und können frei verwendet werden. Die Well-known Ports und die Registered Ports werden durch die Internet Assigned Numbers Authority (IANA) vergeben.

Unter Linux/UNIX ist die Zuordnung von Anwendungen (Diensten) zu Portnummern in der Datei `/etc/services` festgelegt. Beim Betriebssystem Windows existiert für diesen Zweck diese Datei:

```
%WINDIR%\system32\drivers\etc\services
```

Sockets are the platform-independent, standardized interface between the implementation of the network protocols in the operating system and the applications. A socket consists of a port number and an IP address. *Datagram sockets* use the connectionless Transport Layer protocol UDP, and *Stream sockets* use the connection-oriented Transport Layer protocol TCP. The command `netstat` outputs an overview of the sockets.

Sockets sind die plattformunabhängige, standardisierte Schnittstelle zwischen der Implementierung der Netzwerkprotokolle im Betriebssystem und den Anwendungen. Ein Socket besteht aus einer Portnummer und einer IP-Adresse. *Datagram Sockets* verwendeten das verbindungslose Transportprotokoll UDP und *Stream Sockets* das verbindungsorientierte Transportprotokoll TCP. Eine Übersicht der Sockets ermöglicht das Kommando `netstat`.

8.3

User Datagram Protocol (UDP)

With UDP, data transmissions can take place without prior connection establishment. In contrast to TCP, UDP is only responsible for the addressing of the segments. The data transmission is not secure. The receiver does not acknowledge transmissions at the sender. With UDP, segments can get lost during transmission. Depending on the application, for example, real-time communication (video telephony), this is acceptable [18].

If a segment, i.e., image information and/or sound information, gets lost during communication via TCP, it is requested again and dropouts could occur therefore. To compensate for this, playback buffers are required. Especially with real-time communication, however, these buffers should be kept as small as possible because they lead to delays. If, on the other hand, UDP is used for real-time communication, only one image/audio segment is lost if a segment fails to arrive at the receiver.

The maximum size of a UDP segment is 65,535 bytes. This is because the field in the UDP header, which contains the segment length, is 16 bits long and the maximum number that can be represented with 16 bits is 65,535. Such large UDP segments, however, are transmitted in fragments by the IP.

User Datagram Protocol (UDP)

Bei UDP finden Datenübertragungen ohne vorherigen Verbindungsaufbau statt. Im Gegensatz zu TCP ist UDP ausschließlich für die Adressierung der Segmente zuständig. Es findet keine Sicherung der Datenübertragung statt. Übertragungen werden nicht vom Empfänger beim Sender bestätigt. Bei UDP können also Segmente bei der Übertragung verloren gehen. Abhängig von der Anwendung, zum Beispiel bei Echtzeitkommunikation (Videotelefonie), ist das auch akzeptabel [18].

Geht bei der Kommunikation via TCP ein Segment, also eine Bildinformation und/oder Toninformation verloren, wird es neu angefordert und es käme zu Aussetzern. Um das zu kompensieren, sind Wiedergabepuffer nötig. Speziell bei Echtzeitkommunikation versucht man aber die Puffer möglichst klein zu halten, weil diese zu Verzögerungen führen. Nutzt man hingegen UDP zur Echtzeitkommunikation, geht beim Verlust eines Segments nur ein Bild/Tonsegment verloren.

Die maximale Größe eines UDP-Segments ist 65.535 Bytes. Der Grund dafür ist, dass das Feld im UDP-Header, das die Segmentlänge enthält, 16 Bits lang ist und die maximal darstellbare Zahl mit 16 Bits ist 65.535. So große UDP-Segmente werden vom IP aber fragmentiert übertragen.

8.3.1

Format of UDP Segments

The UDP header consists of four 16 bit long fields (see Figure 8.1). The first field contains the port number of the sending process. This is necessary so that the receiver learns to whom it has to respond for the segment. The field can remain empty (value `0`) if no response is required. The second field contains the port number of the process that shall receive the segment. The last two fields contain the *length* of the entire segment (without pseudo-header) and a *checksum* over the entire segment (including pseudo-header).

Aufbau von UDP-Segmenten

Der UDP-Header besteht aus vier je 16 Bit langen Datenfeldern (siehe Abbildung 8.1). Das erste Datenfeld enthält die Portnummer des sendenden Prozesses. Diese ist nötig, damit der Empfänger weiß, wem er auf das Segment antworten soll. Das Datenfeld kann frei bleiben (Wert `0`), wenn keine Antwort erforderlich ist. Das zweite Feld enthält die Portnummer des Prozesses, der das Segment empfangen soll. Die letzten beiden Datenfelder enthalten die *Länge* des vollständigen Segments (ohne Pseudo-Header)

und eine *Prüfsumme* über das vollständige Segment (inklusive Pseudo-Header).

Figure 8.1: Fields of a UDP Segment including the Pseudo-Header

A pseudo-header is created (see Figure 8.1), which contains the IP addresses of the sender and the destination as well as information of the Network Layer. The protocol ID of UDP is 17. The fields of the pseudo-header are not transferred but are used for the calculation of the checksum.

Es wird ein Pseudo-Header erzeugt (siehe Abbildung 8.1), der mit den IP-Adressen von Sender und Ziel auch Informationen der Vermittlungsschicht enthält. Die Protokoll-ID von UDP ist die 17. Die Felder des Pseudo-Headers werden nicht übertragen, gehen aber in die Berechnung der Prüfsumme mit ein.

8.4

Transmission Control Protocol (TCP)

The connection-oriented TCP extends the Network Layer protocol IP with the reliability that is desired or necessary for many applications. TCP guarantees that segments reach their destination fully intact and in the correct order. The sender sends lost or unacknowledged TCP segments again.

A TCP connection is opened and closed in the same way as a file. As with a file, the position in the data stream must be precisely specified. The position is specified. This is done with sequence numbers. With these unique sequence numbers, the receiver can correct the order of the segments and sort out duplicate segments. The length of a segment is indicated in the IP header. This way, gaps in the data stream

Transmission Control Protocol (TCP)

Das verbindungsorientierte TCP erweitert das Vermittlungsprotokoll IP um die Zuverlässigkeit, die für viele Anwendungen gewünscht oder notwendig ist. Bei TCP ist garantiert, dass Segmente vollständig und in der korrekten Reihenfolge ihr Ziel erreichen. Verlorene oder unbestätigte TCP-Segmente sendet der Sender erneut.

Eine TCP-Verbindung wird wie eine Datei geöffnet und geschlossen. Genau wie bei einer Datei ist die Position im Datenstrom exakt angeben. Die Positionsangabe geschieht mit Sequenznummern (Folgenummern). Mit den eindeutigen Sequenznummern kann der Empfänger die Reihenfolge der Segmente korrigieren und doppelt angekommene Segmente aussortieren. Die Länge eines Segments ist aus dem IP-Header bekannt.

are detected, and the receiver can request lost segments again.

TCP treats payload as an ordered data stream. The sequence numbers generate indices for the stream of bytes sent. The sequence number of a segment is the position of the first byte of the segment in the byte stream. In the example (see Figure 8.2), the sender splits a stream with 5,000 bytes of Application Layer data into segments using a MSS of 1,460 bytes.

For Ethernet, the Maximum Transmission Unit (MTU), i.e., the maximum amount of payload of a frame on the Data Link Layer, is 1,500 bytes.

The maximum segment size (MSS), i.e., the maximum amount of Application Layer data that a segment of the Transport Layer can contain as payload, results from the MTU minus the additional overhead for the used protocols of the Network Layer and Transport Layer.

So werden Lücken im Datenstrom entdeckt und der Empfänger kann verlorene Segmente neu anfordern.

TCP sieht Nutzdaten als geordneten Datenstrom. Die Sequenznummern nummerieren den Strom der gesendeten Bytes. Die Sequenznummer eines Segments ist die Position des ersten Bytes des Segments im Bytestrom. Im Beispiel (siehe Abbildung 8.2) unterteilt der Sender einen Strom mit 5.000 Bytes Anwendungsdaten in Segmente bei einer MSS von 1.460 Bytes.

Bei Ethernet ist die Maximum Transmission Unit (MTU), also die maximale Menge an Nutzdaten eines Rahmens auf der Sicherungsschicht 1.500 Bytes.

Die Maximum Segment Size (MSS), also die maximale Menge an Anwendungsdaten, die ein Segment der Transportschicht als Nutzdaten enthalten kann, ergibt sich aus der MTU abzüglich des Mehraufwands (Protokoll-Overhead) für die verwendeten Protokolle der Vermittlungs- und Transportschicht.

Figure 8.2: Subdivision of a Data Stream from the Application Layer into TCP Segments with an MSS of 1,460 Bytes

8.4.1

Format of TCP Segments

A TCP segment can contain a maximum payload of 64 kB (Application Layer data). Smaller segments are common ($\leq$ 1,500 bytes for Ethernet or approximately 4 kB for Token Ring). In addition to the payload, TCP segments also have a header, which is much more complex than the header of UDP segments (see Figure 8.3). However, the TCP header (without the Options field) is only 20 bytes long. Together with the header of IPv4, which is also only 20 bytes long without the Options field, the protocol overhead

Aufbau von TCP-Segmenten

Ein TCP-Segment kann maximal 64 kB Nutzdaten (Daten der Anwendungsschicht) enthalten. Üblich sind kleinere Segmente ($\leq$ 1.500 Bytes bei Ethernet oder ca. 4 kB bei Token Ring). Neben den Nutzdaten existiert auch bei TCP-Segmenten ein Header. Dieser ist im Gegensatz zum Header von UDP-Segmenten deutlich komplexer (siehe Abbildung 8.3). Dennoch umfasst der TCP-Header nur 20 Byte (ohne das Optionsfeld). Zusammen mit dem Header von IPv4, der ohne das Optionsfeld auch nur 20 Bytes umfasst, ist bei einer IP-Paketgröße von mehreren

caused by the headers is low, given an IP packet with several kB in size.

kB der entstehende Protokoll-Overhead durch die Header gering.

Figure 8.3: Fields of a TCP Segment including the Pseudo-Header

One field contains the port number of the sending process and another field the port number of the process which is intended to receive the segment. *Seq number* contains the sequence number of the current segment and *Ack number* the sequence number of the next expected segment.

The field *Length* is 4 bits long and contains the length of the TCP header in 32-bit words so that the receiver understands where the payload begins in the TCP segment. This field is necessary because the field *Options and padding* can have a variable length (a multiple of 32 bits). The following field is 6 bits long and is not used. It has the value zero. This is followed by six fields, each 1 bit in size, which are required for the connection establishment, data exchange and connection termination. In the following, the functions of these fields are described in such a way that they have the value 1, i.e., they are *set*.

- *URG* (Urgent) indicates that the data from the header should be processed imme-

Je ein Datenfeld enthält die Portnummer des sendenden Prozesses und die Portnummer des Prozesses, der das Segment empfangen soll. *Seq-Nummer* enthält die Sequenznummer des aktuellen Segments und *Ack-Nummer* die Sequenznummer des nächsten erwarteten Segments.

Das Datenfeld *Länge* ist 4 Bits groß und enthält die Länge des TCP-Headers in 32-Bit-Worten, damit der Empfänger weiß, wo die Nutzdaten im TCP-Segment anfangen. Dieses Feld ist nötig, weil das Datenfeld *Optionen und Füllbits* eine variable Länge (Vielfaches von 32 Bits) haben kann. Das folgende Datenfeld ist 6 Bits groß wird nicht verwendet. Es hat den Wert Null. Anschließend folgen sechs je 1 Bit große Datenfelder, die für den Verbindungsaufbau, Datenaustausch und Verbindungsabbau benötigt werden. Im Folgenden sind die Funktionen dieser Datenfelder jeweils so beschrieben, dass sie den Wert 1 haben, also *gesetzt* sind.

- *URG* (Urgent) zeigt an, dass die Daten ab dem Header sofort vom Empfänger abgear-

diately by the receiver. This is equivalent to a software interrupt, because it interrupts the processing of a possibly existing other segment.

- *ACK* (Acknowledge) indicates that the acknowledgment number in the field *Ack number* is valid. It is therefore used to acknowledge the reception of segments.

- *PSH* (Push) instructs the protocol stack of the operating system at the sender to transmit the segment immediately and not wait until the send buffer is full. It also instructs the protocol stack of the operating system at the receiver to immediately pass the segment to the Application Layer and also to bypass the receive buffer.

- *RST* (Reset) is a request to reset (abort) the connection.

- *SYN* (Synchronize) instructs the synchronization of the sequence numbers. This initiates the connection establishment.

- *FIN* (Finish) instructs the termination of the connection and indicates that the sender will no longer send payload.

The field *Receive window* contains the number of free bytes in the receive window of the sender and is used for flow control. As with UDP, there is also a pseudo-header for each TCP segment, which is not transmitted, but whose fields, including the regular TCP header and payload, are used for the checksum calculation. The protocol ID of TCP is 6.

If the 1 bit long field *URG* has the value `1`, the content of the field *Urgent pointer* is interpreted. If this field is added to the content of the field *Seq number*, the result is the position of the first byte after the urgent data in the data stream. The urgent data itself, which the receiver is

beitet werden sollen. Das entspricht einem Softwareinterrupt, weil es die Abarbeitung eines eventuell vorhandenen anderen Segments unterbricht.

- *ACK* (Acknowledge) gibt an, dass die Bestätigungsnummer im Datenfeld *Ack-Nummer* gültig ist. Es wird also verwendet, um den Empfang von Segmenten zu bestätigen.

- *PSH* (Push) weist den Protokollstack des Betriebssystems beim Sender an, das Segment sofort zu übertragen und nicht zu warten, bis der Sendepuffer voll ist. Zudem weist es den Protokollstack des Betriebssystems beim Empfänger an, das Segment sofort an die Anwendungsschicht weiterzureichen und den Empfangspuffer ebenfalls zu übergehen.

- *RST* (Reset) ist eine Anforderung die Verbindung zurückzusetzen (abzubrechen).

- *SYN* (Synchronize) weist die Synchronisation der Sequenznummern an. Das initiiert den Verbindungsaufbau.

- *FIN* (Finish) weist den Verbindungsabbau an und gibt an, dass der Sender keine Nutzdaten mehr schicken wird.

Das Datenfeld *Empfangsfenster* enthält die Anzahl freier Bytes im Empfangsfensters des Senders zur Flusskontrolle. Genau wie bei UDP existiert auch für jedes TCP-Segment ein Pseudo-Header, der nicht übertragen wird, aber dessen Datenfelder inklusive regulärem TCP-Header und Nutzdaten in die Berechnung der Prüfsumme mit eingehen. Die Protokoll-ID von TCP ist die 6.

Hat das 1 Bit große Datenfeld *URG* den Wert `1`, wird der Wert des Feldes *Urgent-Zeiger* interpretiert. Dieses Datenfeld addiert mit dem Inhalt von *Seq-Nummer* ergibt die Position des ersten Bytes nach den Urgent-Daten im Datenstrom. Die Urgent-Daten selbst, die der Emp-

intended to process immediately, begins directly after the header.

The field *Options and padding* may contain additional information that cannot be provided by the other fields of the TCP header. One example is the specification of the maximum segment size. The size of the field must be a multiple of 32 bits, and it is filled with zero bits if necessary.

fänger unverzüglich abarbeiten soll, beginnen direkt nach dem Header.

Das Datenfeld *Optionen und Füllbits* kann Zusatzinformationen enthalten, die nicht von den übrigen Feldern im TCP-Header erbracht werden können. Ein Beispiel ist die Angabe der maximalen Segmentgröße. Das Datenfeld muss ein Vielfaches von 32 Bits groß sein und wird, wenn nötig, mit Null-Bits aufgefüllt.

8.4.2

Functioning of TCP

Each transmitted TCP segment has a unique sequence number, which indicates the position of its first byte in the byte stream of the connection. Using this number, the order of the segments can be corrected, and duplicated segments can be sorted out. The length of a segment is specified in the IP header. This allows to detect gaps in the data stream, and the receiver can request lost segments again.

Arbeitsweise von TCP

Jedes gesendete TCP-Segment hat eine eindeutige Sequenznummer, welche die Position seines ersten Bytes im Byte-Strom der Verbindung angibt. Anhand dieser Nummer kann die Reihenfolge der Segmente korrigiert und doppelt angekommene Segmente können aussortiert werden. Die Länge eines Segments ist aus dem IP-Header bekannt. So können Lücken im Datenstrom entdeckt werden, und der Empfänger kann verlorengegangene Segmente neu anfordern.

TCP Connection Establishment with Three-Way Handshake

When establishing a TCP connection, both communication partners exchange control information in three steps (see Figure 8.4 and Figure 8.5). This ensures that the respective partner exists and can accept data.

1. The client sends a segment with the request to synchronize the sequence numbers. This segment with the SYN bit set announces the initial sequence number of the client, which it randomly determines.

2. The server responds with a segment in which it acknowledges the reception of the first segment with the ACK bit set and in turn requests synchronization of the sequence numbers with the SYN bit set. The server also specifies its randomly determined initial sequence number for transmissions in the opposite direction.

Verbindungsaufbau via Dreiwege-Handshake

Beim Öffnen einer TCP-Verbindung tauschen beide Kommunikationspartner in drei Schritten Kontrollinformationen aus (siehe Abbildung 8.4 und Abbildung 8.5). So ist sichergestellt, dass der jeweilige Partner existiert und Daten annehmen kann.

1. Der Client schickt ein Segment mit der Aufforderung, die Sequenznummern zu synchronisieren. Dieses Segment mit gesetztem SYN-Bit gibt die Anfangs-Sequenznummer des Clients bekannt, die dieser zufällig bestimmt.

2. Der Server antwortet mit einem Segment, in dem er den Empfang des ersten Segments mit gesetztem ACK-Bit bestätigt und seinerseits mit gesetztem SYN-Bit zur Synchronisation der Sequenznummern auffordert. Auch der Server gibt seine zufällig bestimmte Anfangs-Sequenznummer für Übertragungen in die Gegenrichtung an.

3. The client acknowledges the reception with a segment with the ACK bit set. From this point on, data can be exchanged via the connection.

This type of exchange of control information, where each side must confirm the action of the other side before it takes effect, is called *three-way handshake*.

3. Der Client bestätigt den Empfang mit einem Segment mit gesetztem ACK-Bit. Ab diesem Zeitpunkt können Daten über die Verbindung ausgetauscht werden.

Diese Art des Austausches von Kontrollinformationen, bei der jede Seite die Aktion der Gegenseite bestätigen muss, ehe sie wirksam werden, heißt *Dreiwege-Handshake*.

Figure 8.4: Three-Way Handshake in TCP

Data Transmission

To demonstrate the process of data transmission with TCP, particular values are required for the fields *Seq number* (sequence number of the current segment) and *Ack number* (sequence number of the next expected segment). In the following example, the sequence number `x` of the client has the value 100 at the beginning of the three-way handshake, and the sequence number `y` of the server has the value 500. After completion of the three-way handshake and at the beginning of the data transmission, `x` and `y` have the values 101 and 501.

In the presented example (see Figure 8.6), the client first transmits 1,000 bytes of payload to the server. The server acknowledges the received payload with the ACK bit set and requests the sequence number of the next segment with the Ack number 1,101. In the same segment, the

Datenübertragung

Um den Ablauf der Datenübertragung bei TCP zu demonstrieren, sind für die *Seq-Nummer* (Sequenznummer des aktuellen Segments) und die *Ack-Nummer* (Sequenznummer des nächsten erwarteten Segments) konkrete Werte nötig. Im folgenden Beispiel hat die Sequenznummer `x` des Clients zu Beginn des Dreiwege-Handshake den Wert 100 und die Sequenznummer `y` des Servers den Wert 500. Nach Abschluss des Dreiwege-Handshake und zu Beginn der Datenübertragung haben `x` und `y` die Werte 101 und 501.

Im konkreten Beispiel (siehe Abbildung 8.6) überträgt der Client zuerst 1.000 Bytes Nutzdaten an den Server. Der Server bestätigt den Empfang der Nutzdaten mit gesetztem ACK-Bit und fordert mit der Ack-Nummer 1.101 die Sequenznummer des nächsten Segments an. Im

Figure 8.5: TCP – Simplified Finite State Machine [17, 33]

server transfers 400 bytes of payload to the client. The client then transmits another 1,000 bytes of payload, acknowledges the received payload with the ACK bit set and requests the sequence number of the next segment with the Ack number 901. Finally, the server acknowledges the received payload again with the ACK bit set and requests the sequence number of the next segment with the Ack number 2,101.

gleichen Segment überträgt der Server 400 Bytes Nutzdaten an den Client. Daraufhin überträgt der Client weitere 1.000 Bytes Nutzdaten, bestätigt den Empfang der Nutzdaten mit gesetztem ACK-Bit und fordert mit der Ack-Nummer 901 die Sequenznummer des nächsten Segments an. Abschließend bestätigt der Server wieder den Empfang der Nutzdaten mit gesetztem ACK-Bit und fordert mit der Ack-Nummer 2.101 die Sequenznummer des nächsten Segments an.

Figure 8.6: Data Transmission in TCP

Connection Termination

The termination of a connection is similar to the establishment of a connection via the three-way handshake. However, instead of the SYN bit, the FIN bit is used, which indicates that no more payload has to be expected from the sender. The connection is terminated in four steps (see Figure 8.5 and Figure 8.7):

1. The client sends its request to terminate the connection with a segment in which the FIN bit is set.

2. The server acknowledges the reception of the request to terminate the connection with a segment in which the ACK bit is set.

Verbindungsabbau

Der Verbindungsabbau ist dem Verbindungsaufbau via Dreiwege-Handshake ähnlich. Statt des SYN-Bit kommt aber das FIN-Bit zum Einsatz, das anzeigt, dass keine Nutzdaten mehr vom Sender kommen. Der Verbindungsabbau erfolgt in vier Schritten (siehe Abbildung 8.5 und Abbildung 8.7):

1. Der Client schickt seinen Abbauwunsch mit einem Segment, in dem das FIN-Bit gesetzt ist.

2. Der Server bestätigt den Empfang des Abbauwunschs mit einem Segment, in dem das ACK-Bit gesetzt ist.

3. The server now sends its request to terminate with a segment in which the FIN bit is set.

4. The client acknowledges the reception of the termination request with a segment in which the ACK bit is set, which completes the connection termination process.

3. Der Server sendet nun selbst seinen Abbauwunsch mit einem Segment, in dem das FIN-Bit gesetzt ist.

4. Der Client bestätigt den Empfang des Abbauwunschs mit einem Segment mit gesetztem ACK-Bit, was den Verbindungsabbau abschließt.

Figure 8.7: Connection Termination in TCP

The connection termination process described here (see Figure 8.7) is initiated by the client. However, it is also possible for the server to initiate the disconnection. In this case, the procedure is the same as the one that is described here.

Der hier beschriebene und in Abbildung 8.7 gezeigte Ablauf des Verbindungsabbaus wird vom Client initiiert. Es ist aber auch möglich, dass der Server den Verbindungsabbau initiiert. Der Ablauf ist in diesem Fall analog zum hier beschriebenen.

8.4.3

Flow Control

When using TCP, via flow control, the receiver dynamically controls the transmission speed of the sender and this way ensures the completeness of the transmission. Receivers with a low

Flusskontrolle

Via Flusskontrolle steuert der Empfänger bei TCP die Sendegeschwindigkeit des Senders dynamisch und stellt so und die Vollständigkeit der Datenübertragung sicher. Langsame Emp-

performance should not be flooded with data, because this would cause the loss of data. Data that is lost during transmission is resent. Underlying mechanisms for the implementation of flow control are *Acknowledgments*, ACK) as feedback or receipt and time limits (*Timeouts*). The way of calculating timeouts for TCP is described in RFC 2988 (obsolete) and RFC 6298.

fänger sollen nicht mit Daten überschüttet werden, denn dadurch würden Daten verloren gehen. Während der Übertragung verlorene Daten werden erneut gesendet. Grundlegende Mechanismen zur Realisierung der Flusskontrolle sind Bestätigungen (*Acknowledgments*, ACK) als Feedback bzw. Quittung und Zeitschranken (*Timeouts*). Die Art und Weise der Berechnung der Zeitschranken bei TCP ist in RFC 2988 (veraltet) und RFC 6298 beschrieben.

Sliding Window

A *Sliding-Window* allows the sender to transmit a given number of segments before an acknowledgment is expected. This way, line and receiver capacity are utilized more efficiently in comparison to the case when the sender waits for an acknowledgment after each transmission of a segment.

When an acknowledgment is received, the transmit window is shifted, and the sender is allowed to transmit further segments. The receiver can also acknowledge several segments at once with *cumulative acknowledgments*.

If a timeout occurs, the sender sends all segments in the window again. It, therefore, resends everything from the last unconfirmed sequence numbers. The flow control mode of operation via sliding windows is illustrated in Figures 8.8 and 8.9.

Schiebefenster

Ein Schiebefenster (*Sliding-Window*) ermöglicht dem Sender die Übertragung einer bestimmten Menge Segmente, bevor eine Bestätigung (Quittung) erwartet wird. So wird die Leitungs- und Empfangskapazität besser ausgelastet, als wenn nach jedem dem Senden eines Segments der Sender auf eine Bestätigung warten würde.

Beim Eintreffen einer Bestätigung wird das Sendefenster verschoben und der Sender kann weitere Segmente aussenden. Der Empfänger kann mit *kumulativen Acknowledgments* auch mehrere Segmente auf einmal bestätigen.

Beim Timeout übermittelt der Sender alle Segmente im Fenster neu. Er sendet also alles ab der letzten unbestätigten Sequenznummer erneut. Die Arbeitsweise der Flusskontrolle via Schiebefenster verdeutlichen die Abbildung 8.8 und 8.9.

Figure 8.8: Sliding Window (Receive Window) at the Receiver

The receive buffer of the receiver contains data for the Application Layer, that is in the correct order, but has not yet been read, and

Der Empfangspuffer des Empfängers enthält Daten für die Anwendungsschicht, die in der korrekten Reihenfolge vorliegen, aber noch nicht

Figure 8.9: Sliding Window (Transmit Window) at the Sender

that has arrived out of sequence. The receiver informs the sender in each segment about the size of its receive window (*Advertised Receive Window*). This avoids buffer overflows at the receiver.

The transmit buffer of the sender contains data of the Application Layer that has already been sent but not yet acknowledged as well as data that is ready to be sent, but has not been sent until now. The receiver determines the size of the transmit window because it corresponds to the size of the receive window. Figure 8.10 shows how sender and receiver control the transmit window with their segments and thus the behavior of the sender.

Each segment to be sent contains the sequence number of the current segment in the field *Seq number*. Each received segment contains the sequence number of the next expected segment in the field *Ack number* and shifts the send window. Also, each segment that is received by the sender contains the number of bytes in the field *Receive window* that the receiver can receive. This value determines the size of the transmit window.

If the receiver informs the sender that its receive window has a size of 0 bytes, this indicates that the receive buffer is full. In this case, the sender is blocked until it is informed by the receiver that there is free storage capacity in the receive window. As soon as capacity becomes available in the receive window, the receiver sends a *window update* to the sender.

gelesen wurden und die außer der Reihe angekommen sind. Der Empfänger teilt dem Sender mit jedem Segment mit, wie groß sein Empfangsfenster (*Advertised Receive Window*) ist. Das vermeidet Pufferüberläufe beim Empfänger.

Der Sendepuffer des Senders enthält Daten der Anwendungsschicht, die bereits gesendet, aber noch nicht bestätigt wurden und die bereits vorliegen, aber noch nicht gesendet wurden. Die Größe des Sendefensters legt der Empfänger fest, denn sie entspricht der Größe des Empfangsfensters. Abbildung 8.10 zeigt, wie Sender und Empfänger mit ihren Segmenten das Sendefenster und damit das Sendeverhalten des Senders steuern.

Jedes zu sendende Segment enthält mit dem Inhalt des Datenfelds *Seq-Nummer* die Sequenznummer des aktuellen Segments. Jedes empfangene Segment enthält mit dem Datenfeld *Ack-Nummer* die Sequenznummer des nächsten erwarteten Segments und verschiebt das Sendefenster. Zudem enthält jedes vom Sender empfangene Segment im Datenfeld *Empfangsfenster* die Angabe, wie viele Bytes der Empfänger empfangen kann. Dieser Wert legt die Größe des Sendefensters fest.

Teilt der Empfänger dem Sender mit, dass sein Empfangsfenster 0 Bytes groß ist, heißt das, dass der Empfangspuffer komplett gefüllt ist. In diesem Fall ist der Sender blockiert, bis er vom Empfänger erfährt, dass im Empfangsfenster freier Speicher ist. Sobald Kapazität im Empfangsfenster frei wird, schickt der Empfänger an den Sender eine *Fensteraktualisierung*.

Figure 8.10: The Principle of Flow Control in TCP – Sliding Window (Receive Window) at the Receiver

Silly Window Syndrome

Window updates immediately inform the sender when a previously overloaded receiver has free storage capacity in the receive window again. However, the risk of the *Silly Window Syndrome* exists, where many small segments are sent, which increases protocol overhead. In this scenario, the receiver sends a segment with the size of the free receive buffer as soon as a few bytes (e.g., 1 byte) have been read from the receive buffer. The sender then sends a segment with this 1 byte of payload. One consequence of this scenario is a protocol overhead of at least 40 bytes for the TCP/IP headers of the segment containing the payload, plus 40 bytes for the segment containing the acknowledgment. Eventually, another 40 bytes must be added, just for a segment containing a window update.

The solution to avoid the Silly Window Syndrome is that the receiver does not notify the transmitter about a free receive window until at least 25 % of the receive buffer is free, or until a segment of the size MSS can be received. This procedure is called *Silly Window Syndrome Avoidance*.

Silly Window Syndrom

Durch Fensteraktualisierungen erfährt der Sender zeitnah, wenn ein zuvor überlasteter Empfänger wieder freie Empfangskapazität hat. Es besteht dabei aber auch die Gefahr des *Silly Window Syndroms*, bei dem sehr viele kleine Segmente geschickt werden, was den Protokoll-Overhead vergrößert. Dabei sendet der Empfänger ein Segment mit der Größe des freien Empfangspuffers, sobald wenige Bytes (zum Beispiel 1 Byte) aus dem Empfangspuffer gelesen wurden. Der Sender sendet daraufhin ein Segment mit lediglich 1 Byte Nutzdaten. Ein Ergebnis dieses Szenarios ist ein Protokoll-Overhead von mindestens 40 Bytes für die TCP/IP-Header des Segments mit den Nutzdaten sowie zuzüglich 40 Bytes für das Segment mit der Bestätigung. Eventuell kommen noch weitere 40 Bytes hinzu, nur für ein Segment mit einer Fensteraktualisierung.

Der Lösungsansatz zur Vermeidung des Silly Window Syndrom ist, das der Empfänger den Sender erst dann über freie Empfangskapazität benachrichtigt, wenn der Empfangspuffer mindestens zu 25 % leer ist oder ein Segment mit der Größe MSS empfangen werden kann. Dieses Vorgehen heißt *Silly Window Syndrom Avoidance*.

8.4.4

Congestion Control

Two possible causes of congestion in computer networks exist:

- Insufficient receiver capacity
- Insufficient network capacity

Congestion at the receiver occurs when the receiver cannot process the received frames/packets/segments fast enough and the receive buffer therefore no longer has any free storage capacity. This problem is already solved by *flow control* (see section 8.4.3). If the receiver specifies the window size according to its receive buffer and the sender complies with it, no buffer overflow can occur at the receiver.

If the utilization (load) of a computer network exceeds its capacity, congestion occurs. In-

Überlastkontrolle

In Computernetzen gibt es zwei mögliche Ursachen für Überlastungen:

- Empfängerkapazität
- Netzkapazität

Zur Überlast beim Empfänger kommt es, wenn dieser die empfangen Rahmen/Pakete/Segmente nicht schnell genug verarbeiten kann und darum der Empfangspuffer keine freie Speicherkapazität mehr hat. Dieses Problem ist bereits durch die *Flusskontrolle* (siehe Abschnitt 8.4.3) gelöst. Spezifiziert der Empfänger die Fenstergröße anhand seines Empfangspuffers und hält sich der Sender daran, kann bei der Empfängerseite kein Pufferüberlauf auftreten.

Wird ein Netzwerk über seine Kapazität hinaus beansprucht, kommt es zu Überlastungen.

dications of network congestion include packet losses due to buffer overflows in Routers, long waiting times caused by full queues in Routers, and frequent retransmissions due to timeout or packet/segment loss. The only useful reaction to network congestion is to reduce the data rate. TCP attempts to avoid congestion by dynamically adapting the window size (*dynamic sliding window*).

Because of the heterogeneity of both potential causes of congestion, there is no single solution for both causes. Thus, both causes are addressed separately.

Congestion Window

In addition to the *Advertised Receive Window*, the sender maintains the *Congestion Window*.

The advertised receive window avoids congestion at the receiver and is offered by the receiver (*advertised*). The congestion window avoids the congestion of the network and is determined by the sender itself. The minimum of both windows is the maximum number of bytes the sender can transmit. For example, if the receive window of the receiver has a free storage capacity of 20 kB, but the sender recognizes that network congestion occurs when more than 12 kB are sent, it will send only 12 kB.

The sender knows precisely at any time how large the receive window is, because the receiver announces this to the sender with every segment. In contrast, determining the optimal size of the congestion window is a challenge for the sender, because it can never be sure about the capacity of the network. Also, the capacity of the network is not static. It depends, among other things, on the current load and malfunctions. As a workaround, the sender must incrementally try to identify the network's maximum transmission capacity.

Setting the Size of the Congestion Window

During connection establishment, the sender initializes the congestion window with the maximum segment size (MSS). Initially, it sends a segment of the size MSS. If the reception of

Anzeichen für Überlastungen der Netzkapazität sind Paketverluste durch Pufferüberläufe in Routern, lange Wartezeiten durch volle Warteschlangen in Routern und häufige Übertragungswiederholungen wegen Timeout oder Paket-/Segmentverlust. Die einzig hilfreiche Reaktion bei Überlastungen der Netze ist eine Reduzierung der Datenrate. TCP versucht Überlastungen durch dynamische Veränderungen der Fenstergröße (*dynamisches Sliding Window*) zu vermeiden.

Wegen der Verschiedenheit beider möglicher Ursachen für Überlastungen existiert nicht eine Lösung für beide Ursachen. Beide Ursachen werden getrennt angegangen.

Überlastungsfenster

Zusätzlich zum Empfangsfenster (*Advertised Receive Window*) verwaltet der Sender das Überlastungsfenster (*Congestion Window*).

Das Empfangsfenster vermeidet Überlast beim Empfänger und wird vom Empfänger angeboten (*advertised*). Das Überlastungsfenster vermeidet die Überlastung des Netzes und wird vom Sender selbst festgelegt. Das Minimum beider Fenster ist die maximale Anzahl Bytes, die der Sender übertragen kann. Kann der Empfänger zum Beispiel gemäß seinem Empfangsfenster 20 kB empfangen, aber der Sender erkennt, dass bei mehr als 12 kB das Netz verstopft, dann sendet er nur 12 kB.

Der Sender kann jederzeit genau sagen, wie groß das Empfangsfenster ist, denn der Empfänger teilt es ihm mit jedem Segment mit. Im Gegensatz dazu ist die Festlegung der optimalen Größe des Überlastungsfensters eine Herausforderung für den Sender, denn er weiß zu keiner Zeit sicher, wie leistungsfähig das Netz ist. Zudem ist die Leistungsfähigkeit der Netze nicht statisch. Sie hängt u. a. von der aktuellen Auslastung und von Netzstörungen ab. Als Lösungsweg muss sich der Sender an das Maximum dessen, was das Netzwerk übertragen kann, herantasten.

Größe des Überlastungsfensters festlegen

Beim Verbindungsaufbau initialisiert der Sender das Überlastungsfenster auf die maximale Segmentgröße (MSS). Dabei sendet er zuerst ein Segment mit der Größe MSS. Wird der Emp-

the segment is acknowledged before the timeout occurs, the congestion window is doubled (see Figure 8.11). After that, two segments of the size MSS are sent. If the reception of both segments is acknowledged before the timeout occurs, the congestion window is doubled again, and so on.

fang des Segments vor dem Timeout bestätigt, wird das Überlastungsfenster verdoppelt (siehe Abbildung 8.11). Daraufhin werden zwei Segmente mit der Größe MSS gesendet. Wird der Empfang beider Segmente vor dem Timeout bestätigt, wird das Überlastungsfenster erneut verdoppelt, usw.

Figure 8.11: Setting the Size of the Congestion Window during Connection Establishment

The congestion window grows exponentially until the receive window size that has been specified by the receiver is reached, or the *Threshold* value is reached, or a timeout occurs. The exponential growth phase is called *slow start* (RFC 2581) because of the sender's low transmission rate at the beginning.

If the congestion window has reached the size of the receive window, it does not grow any further (see Figure 8.12). At the beginning of the transmission, the threshold value is 2^{16} bytes = 64 kB, so it does not play a significant role at the beginning. The maximum size of the receive window is $2^{16}-1$ bytes. This value is determined by the size of the field *Window size* in the TCP header.

If a timeout occurs, the threshold value is reduced to half the congestion window's size, and the size of the congestion window is reduced to the size 1 MSS (see Figure 8.12). After that, again, the slow start phase follows. If the threshold value is reached, the congestion window grows in a linear way until the size of the

Das Überlastungsfenster wächst exponentiell bis das vom Empfänger festgelegte Empfangsfenster erreicht ist oder der Schwellenwert (*Threshold*) erreicht ist oder es zum Timeout kommt. Die exponentielle Wachstumsphase heißt, wegen der niedrigen Senderate des Sender am Anfang, *Slow Start* (RFC 2581).

Hat das Überlastungsfenster die Größe des Empfangsfensters erreicht, wächst es nicht weiter (siehe Abbildung 8.12). Der Schwellenwert ist am Anfang der Übertragung 2^{16} Bytes = 64 kB, damit er zu Beginn keine Rolle spielt. Das Empfangsfenster ist maximal $2^{16} - 1$ Bytes groß. Dieser Wert ist durch die Größe des Datenfelds *Empfangsfenster* im TCP-Header festgelegt.

Kommt es zum Timeout, wird der Schwellenwert auf die Hälfte des Überlastungsfensters herabgesetzt und das Überlastungsfenster auf die Größe 1 MSS reduziert (siehe Abbildung 8.12). Daraufhin folgt erneut die Phase Slow Start. Wird der Schwellenwert erreicht, wächst das Überlastungsfenster linear bis das vom Empfän-

Figure 8.12: Setting the Size of the Congestion Window

receive window is reached, which is determined by the receiver, or a timeout occurs. The linear growth phase is called *congestion avoidance*. (RFC 2581).

Fast Retransmit

A timeout can have several causes. In addition to the network congestion and the resulting delay, a loss of the transmission or a loss of the acknowledgment (ACK) may also occur.

Not only delays due to congestion, but also each loss event will reduce the congestion window to size 1 MSS. The procedure described so far corresponds to the obsolete algorithm for avoiding congestion, TCP *Tahoe* (1988). More modern TCP versions distinguish between an expired timeout caused by congestion of the network and multiple arrivals of acknowledgments caused by a loss event.

If a segment gets lost, a gap occurs in the data stream at the receiver. The receiver sends for each additional segment that is received after this gap an ACK for the segment before the

ger festgelegte Empfangsfenster erreicht ist oder es zum Timeout kommt. Die Phase des linearen Wachstums heißt *Congestion Avoidance* (RFC 2581).

Fast Retransmit

Ein Timeout kann verschiedene Gründe haben. Außer der Überlast des Netzes und der daraus resultierenden Verzögerung, kann auch ein Verlust der Sendung oder ein Verlust der Bestätigung (ACK) vorliegen.

Nicht nur Verzögerungen durch Überlast, sondern auch jedes Verlustereignis reduziert das Überlastungsfenster auf 1 MSS. Das bislang beschriebe Vorgehen entspricht dem veralteten Algorithmus zur Vermeidung von Überlast TCP *Tahoe* (1988). Modernere TCP-Versionen unterscheiden zwischen einem Timeout wegen Netzüberlast und dem mehrfachem Eintreffen von Bestätigungen wegen einem Verlustereignis.

Geht ein Segment verloren, entsteht im Datenstrom beim Empfänger eine Lücke. Der Empfänger sendet bei jedem weiteren nach dieser Lücke empfangenen Segment ein ACK für das

lost segment. If a segment gets lost, there is a gap in the data stream at the receiver. For each further segment that is received after this gap, the receiver sends an ACK for the segment before the lost segment. When a segment is lost, a reduction of the congestion window to size 1 MSS is not necessary, because the segment loss has not been caused by congestion in any case. The more modern algorithm to avoid congestion, TCP *Reno* (1990), retransmits the lost segment after receiving three duplicate ACKs (see Figure 8.13). This procedure is called *fast retransmit* (RFC 2581).

Segment vor dem verlorenen Segment. Beim Segmentverlust ist eine Reduzierung des Überlastungsfensters auf 1 MSS unnötig, denn für einen Segmentverlust ist nicht zwingend Überlastung verantwortlich. Der modernere Algorithmus zur Vermeidung von Überlast TCP *Reno* (1990) sendet nach dreimaligem Empfang eines doppelten ACK das verlorene Segment neu (siehe Abbildung 8.13). Dieses Vorgehen heißt *Fast Retransmit* (RFC 2581).

Figure 8.13: Sending a lost Segment again after receiving three duplicate ACKs

Fast Recovery

TCP *Reno* also avoids the slow start phase after receiving three duplicate ACKs. The congestion window is set to the threshold value immediately after receiving three duplicate ACKs. This method (see Figure 8.14) is called *Fast Recovery* (RFC 2581). The congestion window grows in a linear way. It grows each time a transmission is acknowledged, until the size of the receive win-

Fast Recovery

TCP *Reno* vermeidet auch die Phase Slow Start nach dreimaligem Empfang eines doppelten ACK. Das Überlastungsfenster wird nach dreimaligem Empfang eines doppelten ACK direkt auf den Schwellenwert gesetzt. Dieses Vorgehen (siehe Abbildung 8.14) heißt *Fast Recovery* (RFC 2581). Das Überlastungsfenster wächst mit jeder bestätigten Übertragung linear bis

dow, that is specified by the receiver, is reached, or until a timeout occurs.

das vom Empfänger festgelegte Empfangsfenster erreicht ist oder es zum Timeout kommt.

Figure 8.14: Fast Recovery in TCP *Reno* after the loss of a Segment

Avoiding Congestion

A more recent algorithm to avoid congestion, called TCP *Vegas* (1994), tries to avoid loss events and thus network congestion by observing the *round trip time* of the segments.

The round trip time (RTT) for TCP is the period from sending the segment (i.e., the time at which the segment is passed to the Network Layer) to receiving the acknowledgment (i.e., the time when the ACK is passed from the Network Layer to the Transport Layer).

If the RTT grows, TCP *Vegas* reduces the size of the congestion window in a linear way.

Überlast vermeiden

Ein aktuellerer Algorithmus zur Vermeidung von Überlast mit dem Namen TCP *Vegas* (1994) versucht durch das Beobachten der Rundlaufzeit (*Round Trip Time*) der Segmente bevorstehende Verlustereignisse und damit die Netzüberlast zu vermeiden.

Die Round Trip Time (RTT) bei TCP ist der Zeitraum vom Senden des Segments (also von der Übergabe des Segments an die Vermittlungsschicht) bis zum Empfang der Bestätigung (also bis zu dem Zeitpunkt, an dem das ACK von der Vermittlungsschicht an die Transportschicht übergeben wird).

Wächst die RTT, reduziert TCP *Vegas* die Größe des Überlastungsfensters linear.

Additive Increase/Multiplicative Decrease

TCP operates according to the concept of the rapid reduction of the congestion window after a timeout occurs or a loss event occurs, and it implements a slow (linear) growth of the congestion window. TCP tries to minimize network utilization and at the same time to quickly react to signs of congestion.

This concept is called *Additive Increase / Multiplicative Decrease* (AIMD). The reason for the aggressive reduction and conservative increase of the congestion window is that the consequences of a congestion window that is too large are worse than those of a window that is too small. If the window is too small, the available bandwidth remains unused. If, on the other hand, the window is too large, segments get lost and must be transferred again. This increases the congestion of the network even more.

The state of congestion must, therefore, be left as quick as possible. Therefore, the size of the congestion window is significantly reduced after a timeout has expired or a loss event did occur.

Additive Increase/Multiplicative Decrease

TCP arbeitet nach dem Konzept der raschen Reduzierung des Überlastungsfensters nach Timeouts oder Verlustereignissen und des langsamen (linearen) Anwachsens des Überlastungsfensters. So versucht TCP die Netzauslastung zu minimieren und dennoch schnell auf Überlastungsanzeichen zu reagieren.

Dieses Konzept heißt *Additive Increase / Multiplicative Decrease* (AIMD). Der Grund für die aggressive Senkung und konservative Erhöhung des Überlastungsfensters ist, dass die Folgen eines zu großen Überlastungsfensters schlimmer sind als die eines zu kleinen Fensters. Ist das Fenster zu klein, bleibt verfügbare Bandbreite ungenutzt. Ist im Gegensatz dazu das Fenster zu groß, gehen Segmente verloren und müssen erneut übertragen werden. Das führt zu einer weiteren Verschärfung der Netzüberlast.

Der Zustand der Überlastung muss also möglichst rasch verlassen werden. Darum wird die Größe des Überlastungsfensters nach einem Timeout oder Verlustereigniss deutlich reduziert.

Window Scale

In an Ethernet LAN, the maximum latency is usually 1 ms. If, however, sender and receiver communicate via broadband Internet access, latency depends on the geographical distance between the communication partners and the transmission media that are used. In any case, the latency is 10 to 100 times worse than in an Ethernet LAN.

To take advantage of the performance of broadband Internet access, the size of the receive window must grow, because the response time of the communication partners grows.

Window scaling using the TCP Window Scale Option (WSopt) according to RFC 1323 allows larger receive windows. This is an extension of the TCP header. It allows to specify a scaling factor up to a maximum of 2^{14} by which the receive window size is multiplied. The maximum receive window size is, therefore, $2^{14} \times 2^{16} = 2^{30} = 1,073,725,440$ bytes, i.e., 1 GB.

Fensterskalierung

Bei der Kommunikation in einem lokalen Netzwerk mit Ethernet liegt die Latenz meist bei maximal 1 ms. Kommunizieren Sender und Empfänger jedoch über Breitbandinternetzugänge, hängt die Latenz von der geographischen Entfernung zwischen den Kommunikationspartnern und den dazwischenliegenden Übertragungsmedien ab. In jedem Fall ist die Latenz um Faktor 10 bis 100 schlechter als in einem lokalen Ethernet.

Um die Leistungsfähigkeit von Breitbandinternetzugängen auszunutzen, muss auch die Größe des Empfangsfensters wachsen, einfach weil die Reaktionsdauer der Kommunikationspartner zunimmt.

Via Fensterskalierung mit der TCP Window Scale Option (WSopt) gemäß RFC 1323 sind auch größere Empfangsfenster möglich. Es handelt sich um eine Erweiterung des TCP-Headers. Diese erlaubt es einen Skalierungsfaktor bis maximal 2^{14} anzugeben, mit dem das Empfangsfenster multipliziert wird. Die maximale Empfangsfenstergröße ist somit $2^{14} * 2^{16} = 2^{30} =$ 1.073.725.440 Bytes, also 1 GB.

The precondition for using window scaling is that it is implemented by both communication partners.

Die Voraussetzung für Fensterskalierung ist, dass beide Kommunikationspartner dieses unterstützen.

9

Application Layer

The Application Layer is the top layer of the OSI reference model and the hybrid reference model. It contains the application protocols and services based on them, including data transfer, synchronization and remote control of computers and name resolution.

This chapter covers some of the most common Application Layer protocols.

Anwendungsschicht

Die Anwendungsschicht ist die letzte Schicht des OSI-Referenzmodells und des hybriden Referenzmodells. Sie enthält die Anwendungsprotokolle und darauf aufbauende Dienste unter anderem zur Datenübertragung, Synchronisierung und Fernsteuerung von Rechnern und Namensauflösung.

Dieses Kapitel behandelt einige der bekanntesten Anwendungsprotokolle.

9.1

Domain Name System (DNS)

The DNS (RFC 1034 and 1035) is a protocol for the name resolution of domain names to IP addresses. DNS replaced the local domain name tables in the file `/etc/hosts`, which until then had been used for managing name/address mappings.

DNS uses a hierarchical namespace. The information with the mapping information is split into separate parts and distributed to *name servers* across the Internet.

The *domain namespace* has a tree-like structure. The leaves and nodes are called *labels*, and each subtree is a *domain*. A full domain name consists of the concatenation of all labels of a path. Labels are alphanumerical strings. The only special character that is allowed is a hyphen. Labels are 1 to 63 characters long. They must begin with a letter and must not begin or end with a hyphen. Each label ends with a period.

Domain names end with a period. This period is usually omitted, but formally belongs to a full domain name (*Fully Qualified Domain-Name*,

Domain Name System (DNS)

Das DNS (RFC 1034 und 1035) ist ein Protokoll zur Namensauflösung von Domainnamen zu IP-Adressen. DNS löste die lokalen Namenstabellen in der Datei `/etc/hosts` ab, die bis dahin für die Verwaltung der Namen/Adressen-Zuordnungen zuständig waren.

DNS basiert auf einem hierarchischen Namensraum. Die Information mit den Zuordnungen sind in separate Teile gegliedert und im gesamten Internet auf *Nameservern* verteilt.

Der *Domain-Namensraum* hat eine baumförmige Struktur. Die Blätter und Knoten heißen *Labels* und jeder Unterbaum ist eine *Domäne*. Ein vollständiger Domainname besteht aus der Verkettung aller Labels eines Pfades. Label sind alphanumerische Zeichenketten. Als einziges Sonderzeichen ist der Bindestrich erlaubt. Labels sind 1 bis 63 Zeichen lang. Sie müssen mit einem Buchstaben beginnen und dürfen nicht mit einem Bindestrich anfangen oder enden. Jedes Label endet mit einem Punkt.

Domainnamen werden mit einem Punkt abgeschlossen. Dieser Punkt wird meist weggelassen, gehört rein formal aber zu einem vollständigen Domainnamen – *Fully Qualified Domain-Name*

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_9

FQDN). For example, a full domain name is `fbi.h-da.de.`

(FQDN) dazu. Ein vollständiger Domainname ist also zum Beispiel `fbi.h-da.de.`

Figure 9.1: Excerpt from the Domain Namespace

9.1.1

Functioning of TCP

Domain names are resolved from right to left (see Figure 9.1). The further to the right a label is, the higher its position in the tree. The first level below the root is called the *top-level domain* (TLD). The DNS objects of a domain (for example, the hostnames) are stored as a set of *resource records* (RR) in a zone file that exists on one or more name servers. The zone file is often merely called *zone*.

The 13 root name servers (`A` to `M`) publish the *root zone* of the DNS. The domain names of the root name servers have the form `letter.root-servers.net` The root zone contains around 3,000 entries and is the root of the DNS. It contains the names and IP addresses of the name servers, which are responsible for the TLDs. Some root servers do not consist of one, but of multiple physical servers, which are combined to a logical server. These computers are located at different places around the world and can be accessed via Anycast (see section 7.2) using the same IP address (see Table 9.1).

DNS requests are usually sent via the connectionless Transport Layer protocol UDP to port 53 to the name server. Name servers send replies with a size up to 512 bytes via UDP to

Arbeitsweise des DNS

Domainnamen werden von rechts nach links aufgelöst (siehe Abbildung 9.1). Je weiter rechts ein Label steht, umso höher steht es im Baum. Die erste Ebene unterhalb der Wurzel heißt *Top-Level-Domain* (TLD). Die DNS-Objekte einer Domäne (zum Beispiel die Rechnernamen) sind als Satz von *Resource Records* (RR) in einer Zonendatei gespeichert, die auf einem oder mehreren Nameservern vorhanden ist. Die Zonendatei heißt häufig einfach *Zone*.

Die 13 Root-Nameserver (`A` bis `M`) publizieren die *Root-Zone* des DNS. Die Domainnamen der Root-Nameserver haben die Form `buchstabe.root-servers.net`. Die Root-Zone enthält ca. 3.000 Einträge und ist die Wurzel des DNS. Sie enthält die Namen und IP-Adressen der für die TLDs zuständigen Nameserver. Einige Root-Server bestehen nicht aus einem, sondern mehreren physischen Servern, die zu einem logischen Server zusammengeschlossen sind. Diese Rechner befinden sich an verschiedenen Standorten weltweit und sind via Anycast (siehe Abschnitt 7.2) über dieselbe IP-Adresse erreichbar (siehe Tabelle 9.1).

DNS-Anfragen werden üblicherweise über das verbindungslose Transportprotokoll UDP an Port 53 zum Namensserver gesendet. Antworten bis zu einer Größe von 512 Bytes senden

Table 9.1: Root Name Servers of the DNS [25]

Name	IPv4 address	IPv6 address	Location
A	198.41.0.4	2001:503:ba3e::2:30	distributed (Anycast)
B	199.9.14.201	2001:500:200::b	distributed (Anycast)
C	192.33.4.12	2001:500:2::c	distributed (Anycast)
D	199.7.91.13	2001:500:2d::d	distributed (Anycast)
E	192.203.230.10	2001:500:a8::e	distributed (Anycast)
F	192.5.5.241	2001:500:2f::f	distributed (Anycast)
G	192.112.36.4	2001:500:12::d0d	distributed (Anycast)
H	198.97.190.53	2001:500:1::53	Maryland and California, USA
I	192.36.148.17	2001:7fe::53	distributed (Anycast)
J	192.58.128.30	2001:503:c27::2:30	distributed (Anycast)
K	193.0.14.129	2001:7fd::1	distributed (Anycast)
L	199.7.83.42	2001:500:9f::42	distributed (Anycast)
M	202.12.27.33	2001:dc3::35	distributed (Anycast)

the clients. Name servers send longer replies via the Transport Layer protocol TCP (RFC 7766).

Namensserver via UDP zu den Clients. Größere Antworten senden Namensserver über das Transportprotokoll TCP (RFC 7766).

9.1.2

Example of a Domain Name Resolution

The following scenario of a name resolution of the domain name `fbi.h-da.de.` demonstrates how DNS works (see Figure 9.2).

1. As the first step, the operating system of the requesting terminal device checks whether the IP address for the domain name `fbi.h-da.de.` is mentioned in the local file `/etc/hosts`. If this is not the case, the terminal device queries the DNS server, which is either permanently registered or automatically assigned via DHCP.

2. If the DNS server of the terminal device has buffered an IP address for the requested domain name, it sends the address to the terminal device. Otherwise, it requests the IP address for the domain name `fbi.h-da.de.` from one of the 13 root name servers.

3. The root name server sends the domain names and IP addresses of the name

Auflösung eines Domainnamens

Das folgende Szenario einer Namensauflösung des Domainnamens `fbi.h-da.de.` demonstriert die Arbeitsweise des DNS (siehe Abbildung 9.2).

1. Im ersten Schritt prüft das Betriebssystem des anfragenden Endgeräts, ob in der lokalen Datei `/etc/hosts` die IP-Adresse für den Domainnamen `fbi.h-da.de.` hinterlegt ist. Ist das nicht der Fall, fragt das Endgerät bei dem DNS-Server nach, der entweder fest eingetragen ist oder per DHCP automatisch zugewiesen wurde.

2. Hat der DNS-Server des Endgeräts eine IP-Adresse für den angefragten Domainnamen zwischengespeichert, sendet er die Adresse an das Endgerät. Andernfalls fragt er einen der 13 Root-Nameserver nach der IP-Adresse für den Domainnamen `fbi.h-da.de.`

3. Der Root-Nameserver sendet die Domainnamen und IP-Adressen der Nameserver

servers for the zone `de.` to the DNS server of the terminal device.

4. The DNS server of the terminal device queries a name server of the zone `de.` for the IP address of the domain name `fbi.h-da.de.`

5. The name server of the zone `de.` sends the domain names and IP addresses of the name servers of the zone `h-da.de.` to the DNS server of the terminal device.

6. The DNS server of the terminal device requests the IP address for the domain name `fbi.h-da.de.` at a name server of the `h-da.de.` zone.

7. The name server of the zone `h-da.de.` sends the IP address for the domain name `fbi.h-da.de.` to the DNS server of the terminal device.

8. The DNS server of the terminal device sends the IP address for the domain name `fbi.h-da.de.` to the terminal device.

für die Zone `de.` zum DNS-Server des Endgeräts.

4. Der DNS-Server des Endgeräts fragt einen Nameserver für die Zone `de.` nach der IP-Adresse für den Domainnamen `fbi.h-da.de.`.

5. Der Nameserver der Zone `de.` sendet die Domainnamen und IP-Adressen der Nameserver für die Zone `h-da.de.` zum DNS-Server des Endgeräts.

6. Der DNS-Server des Endgeräts fragt einen Nameserver für die Zone `h-da.de.` nach der IP-Adresse für den Domainnamen `fbi.h-da.de.`

7. Der Nameserver der Zone `h-da.de.` sendet die IP-Adresse für den Domainnamen `fbi.h-da.de.` zum DNS-Server des Endgeräts.

8. Der DNS-Server des Endgeräts sendet die IP-Adresse für den Domainnamen `fbi.h-da.de.` zum Endgerät.

Figure 9.2: Resolution of the Domain Name `fbi.h-da.de`

9.2

Dynamic Host Configuration Protocol (DHCP)

The protocol *Dynamic Host Configuration Protocol* (DHCP), which is specified in RFC 2131, is used to assign the network configuration (e.g., IP address, netmask, default gateway, name server) to network devices by using a *DHCP client* through a *DHCP server*. DHCP uses UDP and the ports 67 (Server or Relay Agent) and 68 (Client).

With DHCP, network settings are automatically assigned in a LAN. Especially with mobile devices, it is not useful to assign fixed IP addresses, as otherwise the network settings would have to be configured on all clients when the network topology changes. With DHCP, only the configuration of the DHCP server has to be adjusted. The DHCP server has a pool of IP addresses and distributes them to requesting clients.

For a DHCP client to be able to use a DHCP server, both must be in the same logical network. The reason for this restriction is that DHCP uses broadcasts and Routers do not forward broadcasts. If the DHCP server is located in a different logical network, a *DHCP relay* must forward the requests to the DHCP server.

Dynamic Host Configuration Protocol (DHCP)

Das in RFC 2131 spezifizierte Protokoll *Dynamic Host Configuration Protocol* (DHCP) ermöglicht die Zuweisung der Netzwerkkonfiguration (IP-Adresse, Netzmaske, Default-Gateway, Nameserver, usw.) an Netzwerkgeräte mit Hilfe eines *DHCP-Clients* durch einen *DHCP-Server*. DHCP verwendet UDP und die Ports 67 (Server oder Relay-Agent) und 68 (Client).

Mit DHCP werden Netzwerkeinstellungen automatisiert im LAN vergeben. Speziell bei mobilen Geräten ist es nicht sinnvoll, feste IP-Adressen zu vergeben, da man ansonsten bei Änderungen an der Topologie des Netzes auf allen Clients die Netzwerkeinstellungen anpassen müsste. Bei DHCP muss nur die Konfiguration des DHCP-Servers angepasst werden. Der DHCP-Server verfügt über einen Pool an IP-Adressen und verteilt diese an anfragende Clients.

Damit ein DHCP-Client einen DHCP-Server nutzen kann, müssen sich beide im selben logischen Netz befinden. Der Grund für diese Einschränkung ist, dass DHCP Broadcasts verwendet und Router keine Broadcasts weiterleiten. Liegt der DHCP-Server in einem anderen logischen Netz, muss ein *DHCP-Relay* die Anfragen an den DHCP-Server weiterleiten.

9.2.1

Functioning of DHCP

A DHCP client that has no IP address sends a request (*DHCP Discover*) via broadcast to the DHCP servers in reach. Multiple DHCP servers might exist in one subnet. The sender IP address of the broadcast is 0.0.0.0, the destination address is 255.255.255.255. Each server in reach, with free IP addresses in its pool, replies to the request with an address offer (*DHCP Offer*). The address offer is also sent as a broadcast with the destination address 255.255.255.255.

The client accepts an address offer by sending a request (*DHCP Request*) to the network via broadcast. The message contains the ID of the

Arbeitsweise von DHCP

Ein DHCP-Client ohne IP-Adresse sendet via Broadcast eine Anfrage (*DHCP-Discover*) an die erreichbaren DHCP-Server. Es kann durchaus mehrere DHCP-Server in einem Subnetz geben. Die Absender-IP-Adresse des Broadcast ist 0.0.0.0 und die Zieladresse ist 255.255.255.255. Jeder erreichbare Server mit freien IP-Adressen in seinem Pool antwortet auf die Anfrage mit einem Adressangebot (*DHCP-Offer*). Auch das Adressangebot wird als Broadcast mit der Zieladresse 255.255.255.255 gesendet.

Der Client nimmt ein Adressangebot an, indem er eine Anfrage (*DHCP-Request*) via Broadcast ins Netzwerk schickt. Die Nachricht enthält

desired DHCP server. Any other DHCP servers that may be present, interpret the message as a rejection for their address offers. The server with the matching ID replies with *DHCP Ack* and marks the address in its address pool as assigned.

If a server has assigned an IP address and confirmed this with *DHCP Ack*, it creates a *lease* record for the address in its database. If all addresses are assigned (leased), no other clients can be supplied with IP addresses. In this case, the server rejects any incoming requests with *DHCP Nak*.

Before the client makes use of an assigned address, it checks whether another network device is accidentally already using that address. To achieve this, an ARP request is used (see section 6.10). If another network device responds to the ARP request, the client rejects the address that the server offers with *DHCP Decline*.

Each address has an expiration date (*lease time*). This is transmitted to the client together with the acknowledgment (*DHCP Ack*). Active clients periodically extend the *lease* time after this half lease time has expired. This is done with another *DHCP Request*. This is sent directly to the server via unicast and not via broadcast. The server replies with a new acknowledgment (*DHCP Ack*) that contains the same information as before as well as a new expiration date. Thus, the lease of the assigned address is prolonged. It is also possible that the server rejects the extension of the address with *DHCP Nak*.

If the server does not reply, for example, because it no longer exists, the client can continue to use its address. After 7/8 of the lease time (87,5 %), the client tries to get an extension of the address from any DHCP server by sending *DHCP Request* as a broadcast to all possibly existing servers in the subnet. The original DHCP server or a new one may acknowledge the address with *DHCP-Ack* or refuse it via *DHCP Nak*.

To return an address before the lease time has expired, a client sends *DHCP Release* to the server.

If the lease time has expired, the server can reassign the address for requests. The client

die ID des gewünschten DHCP-Servers. Eventuell vorhandene weitere DHCP-Server erkennen in der Nachricht die Absage für ihre Adressangebote. Der Server mit der passenden ID antwortet mit *DHCP-Ack* und markiert die Adresse in seinem Adresspool als vergeben.

Hat ein Server eine IP-Adresse vergeben und dies mit *DHCP-Ack* bestätigt, trägt er in seiner Datenbank bei der Adresse ein *Lease* ein. Sind alle Adressen vergeben (verliehen), können keine weiteren Clients mit IP-Adressen versorgt werden. Eintreffende Anfragen lehnt der Server dann mit *DHCP-Nak* ab.

Bevor der Client eine zugewiesene Adresse verwendet, prüft er mit einer ARP-Anforderung (siehe Abschnitt 6.10), ob nicht versehentlich ein anderes Netzwerkgerät diese Adresse bereits verwendet. Antwortet ein anderes Netzwerkgerät auf die ARP-Anforderung, weist der Client die vom Server vorgeschlagene Adresse mit *DHCP-Decline* zurück.

Jede Adresse besitzt ein Verfallsdatum (*Lease Time*). Dieses wird mit der Bestätigung (*DHCP-Ack*) an den Client übermittelt. Aktive Clients verlängern den *Lease* regelmäßig nach der Hälfte des Verfallsdatums mit einem erneuten *DHCP-Request*. Dieser wird direkt via Unicast an den Server gesendet und nicht per Broadcast. Der Server antwortet mit einer erneuten Bestätigung (*DHCP-Ack*) mit den identischen Daten wie vorher und einem neuen Verfallsdatum. Damit gilt die vergebene Adresse als verlängert. Es ist auch möglich, dass der Server die Verlängerung mit *DHCP-Nak* ablehnt.

Bleibt eine Antwortet des Servers aus, weil er beispielsweise nicht mehr existiert, kann der Client die Adresse weiter verwenden. Nach 7/8 der Lease-Zeit (87,5 %) versucht der Client eine Verlängerung der Adresse von irgendeinem DHCP-Server zu erhalten indem er *DHCP-Request* als Broadcast an alle eventuell vorhandenen Server im Subnetz schickt. Der ursprüngliche oder ein neuer DHCP-Server kann die Adresse mit *DHCP-Ack* bestätigen oder via *DHCP-Nak* ablehnen.

Um eine Adresse vor dem Ablauf des Verfallsdatums zurückzugeben, sendet ein Client an den Server *DHCP-Release*.

Ist das Verfallsdatum abgelaufen, kann der Server die Adresse bei Anfragen neu vergeben.

Figure 9.3: State Diagram of DHCP Clients

must then restart the initial address allocation process via *DHCP Discover*.

The state diagram in Figure 9.3 contains the possible states which the DHCP client of a network device can adopt, as well as the events and DHCP messages that cause a state transition.

Der Client muss dann via *DHCP-Discover* erneut den Vorgang der initialen Adresszuweisung starten.

Das Zustandsdiagramm in Abbildung 9.3 enthält die möglichen Zustände, in denen sich der DHCP-Client eines Netzwerkgeräts befinden kann und die Ereignisse und DHCP-Nachrichten, die dazu führen, dass es zum Zustandsübergang kommt.

9.2.2

Structure of DHCP Messages

Figure 9.4 shows the structure of DHCP messages. The field *Operation* defines the meaning of the DHCP message. For a *request* from a client, the field has the value `1`. If the message contains a *reply* from a server, it has the value `2`.

In the field *Type*, the networking technology is indicated. The value `1` represents Ethernet. The field *Length* defines the length of the physical

Aufbau von DHCP-Nachrichten

Abbildung 9.4 zeigt den Aufbau von DHCP-Nachrichten. Im Datenfeld *Operation* ist festgelegt, um was für eine DHCP-Nachricht es sich handelt. Bei einer Anforderung (*Request*) eines Clients hat das Datenfeld den Wert `1`. Enthält die Nachricht eine Antwort (*Reply*) eines Servers, hat es den Wert `2`.

Im Datenfeld *Netztyp* ist die Vernetzungstechnologie angegeben. Der Wert `1` steht für Ethernet. Das Datenfeld *Länge* definiert die Länge

Figure 9.4: Structure of DHCP Messages

network address in bytes. The *Seconds* field contains the seconds since the start of the DHCP client. *Hops* is an optional field and specifies the number of DHCP relays on the route. *Filename* is also an optional field. It contains the name of a file that the client can access via the Trivial File Transfer Protocol (TFTP). This allows a terminal device to boot over the network.

der physischen Netzadresse in Bytes. *Sekunden* enthält die Sekunden seit dem Start des DHCP-Clients. *Hops* ist ein optionales Datenfeld und gibt die Anzahl der DHCP-Relays auf dem Pfad an. Auch das Datenfeld *Dateiname* ist optional und enthält den Namen einer Datei, die sich der Client via Trivial File Transfer Protocol (TFTP) holen soll. Damit kann ein Endgerät über das Netzwerk booten.

9.2.3

DHCP Relay

Routers separate broadcast domains, i.e., they do not forward broadcasts. However, DHCP requests are sent via broadcast. If DHCP is provided with a single DHCP server for multiple subnets, the Routers must be configured in a way, so that they forward DHCP messages.

A Router that is configured as a DHCP relay sends DHCP broadcast requests directly to the DHCP server via unicast. The DHCP server knows the IP address of the Router from the field *IP address (relay)* in the DHCP message.

DHCP-Relay

Router trennen Broadcast-Domänen, lassen also Broadcasts nicht durch. DHCP-Anfragen werden aber via Broadcast gesendet. Soll DHCP mit einem DHCP-Server für mehrere Subnetze angeboten werden, müssen die Router so konfiguriert sein, dass sie DHCP-Nachrichten weiterleiten.

Ein Router, der als DHCP-Relay konfiguriert ist, sendet DHCP-Broadcast-Anfragen direkt via Unicast an den DHCP-Server. Der DHCP-Server kennt die IP-Adresse des Routers aus dem Datenfeld *IP des Relays* in der DHCP-Nachricht.

Therefore, the DHCP server also learns from which subnet the DHCP request originates. The DHCP server also sends its address offer via unicast back to the Router, which forwards the message to the network from which the request came.

Daher weiß der DHCP-Server auch, aus welchem Subnetz die DHCP-Anfrage kommt. Der DHCP-Server sendet sein Adressangebot auch via Unicast an den Router zurück, der die Nachricht in das Netz weiterleitet, aus dem die Anfrage gekommen ist.

9.3

Telecommunication Network (Telnet)

Telnet (RFC 854) is a protocol for the remote control of computers. It implements character-oriented data exchange via TCP and uses port 23 by default. Telnet is only suitable for applications without a graphical user interface, and it does not implement encryption. Even passwords are sent in plain text.

In contrast to Telnet, the protocol *Secure Shell* – SSH (RFC 4251) implements an encrypted and thus more secure remote control of computers. It is the secure alternative to Telnet and uses TCP and port 22 by default.

Due to the lack of encryption, Telnet is too insecure for remote work by today's standards. However, the protocol is often used for troubleshooting other services, such as web servers, FTP servers, or SMTP servers, and for database administration as well as in LANs. Telnet clients can connect to any port number. This enables the administrator to send commands to web servers, FTP servers or SMTP servers via a Telnet client and to monitor their reaction in an unaltered way.

Telnet is based on the standard *Network Virtual Terminal* (NVT), which is a vendor-independent interface. NVT is a conversion concept for different data formats that is supported by all Telnet implementations on all hardware platforms. A NVT consists of an input device and an output device that can both generate and display only a specific set of characters. Telnet clients convert the keystrokes and control characters into the NVT format and send this data to a Telnet server, which in turn decodes and forwards them.

Telecommunication Network (Telnet)

Telnet (RFC 854) ist ein Protokoll zur Fernsteuerung von Rechnern. Es ermöglicht zeichenorientierten Datenaustausch über TCP und verwendet standardmäßig Port 23. Telnet eignet sich nur für Anwendungen ohne grafische Benutzeroberfläche und bietet keine Verschlüsselung. Auch die Passwörter werden im Klartext versendet.

Im Gegensatz zu Telnet, ermöglicht das Protokoll *Secure Shell* – SSH (RFC 4251) eine verschlüsselte und damit sichere Fernsteuerung von Rechnern. Es ist die sichere Alternative zu Telnet und verwendet TCP und standardmäßig Port 22.

Telnet ist wegen der fehlenden Verschlüsselung heute zu unsicher für entferntes Arbeiten. Das Protokoll wird aber häufig zur Fehlersuche bei anderen Diensten, zum Beispiel Web-Servern, FTP-Servern oder SMTP-Servern, und zur Administration von Datenbanken sowie in LANs eingesetzt. Telnet-Clients können sich mit beliebigen Portnummern verbinden. Das ermöglicht dem Administrator, über einen Telnet-Client, Kommandos an Web-Server, FTP-Server oder SMTP-Server zu senden und unverfälscht deren Reaktion zu beobachten.

Telnet basiert auf dem Standard *Network Virtual Terminal* (NVT), was soviel bedeutet wie virtuelles Netzwerkterminal und im Prinzip eine herstellerunabhängige Schnittstelle ist. NVT ist ein Konvertierungskonzept für unterschiedliche Datenformate, das von allen Telnet-Implementierungen auf allen Hardwareplattformen unterstützt wird. Ein NVT besteht aus einem Eingabegerät und einem Ausgabegerät, die jeweils nur bestimmte Zeichen erzeugen bzw. anzeigen können. Telnet-Clients konvertieren die Tasteneingaben und Kontrollanweisungen in das

NVT-Format und übertragen diese Daten an den Telnet-Server, der sie wiederum dekodiert und weiterreicht.

Table 9.2: Control Characters of Telnet

Name	Code	Description
NULL	`NUL`	No operation
Line Feed	`LF`	Moves the cursor to the next line and keeps the column
Carriage Return	`CR`	Moves the cursor to the first column of the current line
BELL	`BEL`	Produces an audible or visible signal
Back Space	`BS`	Moves the cursor one position back
Horizontal Tab	`HT`	Moves the cursor to next horizontal tab stop
Vertical Tab	`VT`	Moves the cursor to the next vertical tab stop
Form Feed	`FF`	Moves the cursor to the first column of the first line and clears the terminal

NVT operates with information units that have a size of 8 bits (1 byte) each, and it uses the 7-bit character encoding US-ASCII. The most significant bit of each character is padded with a zero-bit to achieve 8 bits.

Table 9.2 contains the control statements of NVT. Each Telnet client and server implement the first three control characters. The remaining five control characters are optional.

NVT arbeitet mit Informationseinheiten von je 8 Bits (1 Byte) und verwendet die 7-Bit-Zeichenkodierung US-ASCII. Das höchstwertige Bit jedes Zeichens wird mit einer Null aufgefüllt, um auf 8 Bits zu kommen.

Tabelle 9.2 enthält die Kontrollanweisungen von NVT. Die ersten drei Kontrollzeichen versteht jeder Telnet-Client und -Server. Die übrigen fünf Kontrollzeichen sind optional.

9.4

Hypertext Transfer Protocol (HTTP)

HTTP is a stateless protocol for the transmission of data. Stateless means that every HTTP message contains all the information that is required to understand the message. The server does not maintain any information about the state or session for the client, and each request is a transaction that is independent of other requests.

The primary purpose of HTTP is to load web pages from the World Wide Web (WWW) to a browser. For communication, HTTP needs a reliable Transport Layer protocol. In almost all cases, TCP is used.

Each HTTP message consists of a *message header* (*HTTP header*) and the message *body*. The header contains information about the en-

Hypertext Transfer Protocol (HTTP)

HTTP ist ein zustandsloses Protokoll zur Übertragung von Daten. Zustandslos bedeutet, dass jede HTTP-Nachricht alle nötigen Informationen enthält, um die Nachricht zu verstehen. Der Server hält keine Zustands- bzw. Sitzungsinformation über den Client vor, und jede Anfrage ist eine von anderen Anfragen unabhängige Transaktion.

Der Haupteinsatzzweck von HTTP ist es, Webseiten aus dem World Wide Web (WWW) in einen Browser laden. Zur Kommunikation ist HTTP auf ein zuverlässiges Transportprotokoll angewiesen. In den allermeisten Fällen wird TCP verwendet.

Jede HTTP-Nachricht besteht aus einem *Nachrichtenkopf* (*HTTP-Header*) und dem Nachrichtenkörper (*Body*). Der Nachrichtenkopf

coding, desired language, browser and content type. The body contains the payload, such as the HTML source code of a web page.

If a URL (for example `http://www.informatik.hs-mannheim.de/~baun/index.html`) is accessed via HTTP, a request for the resource `/~baun/index.html` is sent to the server with the hostname `www.informatik.hs-mannheim.de`. First, the hostname is resolved to an IP address using DNS. The following HTTP-`GET` request is sent via TCP to port 80, on which the web server usually operates:

enthält unter anderem Informationen zu Kodierung, gewünschter Sprache, Browser und Inhaltstyp. Der Nachrichtenkörper enthält die Nutzdaten, wie den HTML-Quelltext einer Webseite.

Wird via HTTP auf eine URL (zum Beispiel `http://www.informatik.hs-mannheim.de/~baun/index.html`) zugegriffen, wird an den Rechner mit dem Hostnamen `www.informatik.hs-mannheim.de` eine Anfrage für die Ressource `/~baun/index.html` gesendet. Zuerst wird der Hostname via DNS in eine IP-Adresse aufgelöst. Über TCP wird zu Port 80, auf dem der Web-Server üblicherweise arbeitet, folgende HTTP-`GET`-Anforderung gesendet:

```
GET /~baun/index.html HTTP/1.1
Host: www.informatik.hs-mannheim.de
```

The presented request is shortened, because a browser sends further information besides the HTTP `GET` request, such as information about itself and the language of the client. However, the given HTTP-`GET` request is sufficient.

Die dargestellte Anforderung ist verkürzt, da ein Browser außer der HTTP-`GET`-Anforderung noch weitere Informationen sendet, wie zum Beispiel Informationen über sich selbst und die Sprache des Clients. Die hier angegebene HTTP-`GET`-Anforderung genügt aber völlig.

The header of an HTTP message is separated from the message body by a line feed (`LF`) and a carriage return (`CR`). In this example, however, the HTTP request does not have a message body. The HTTP response of the web server consists of a message header and the message body with the actual message. In this scenario, the message body contains the content of the requested file `index.html`.

Der Nachrichtenkopf einer HTTP-Nachricht wird mit einem Line Feed (`LF`) und einem Carriage Return (`CR`) vom Nachrichtenkörper abgegrenzt. Im Beispiel hat die HTTP-Anforderung aber keinen Nachrichtenkörper. Die HTTP-Antwort des Web-Servers besteht aus einem Nachrichtenkopf und dem Nachrichtenkörper mit der eigentlichen Nachricht. In diesem Fall enthält der Nachrichtenkörper den Inhalt der angeforderten Datei `index.html`.

```
HTTP/1.1 200 OK
Date: Sun, 04 Sep 2011 15:19:13 GMT
Server: Apache/2.2.17 (Fedora)
Last-Modified: Mon, 22 Aug 2011 12:37:04 GMT
ETag: "101ec1-2157-4ab17561a3c00"
Accept-Ranges: bytes
Content-Length: 8535
Keep-Alive: timeout=13, max=499
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
    Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
...
</html>
```

Each HTTP response contains a *status code*, consisting of three digits and a string that describes the reason for the response. Table 9.3 contains some common state codes of HTTP.

Jede HTTP-Antwort enthält einen *Statuscode*, der aus drei Ziffern besteht, und eine Textkette, die den Grund für die Antwort beschreibt. Tabelle 9.3 enthält einige bekannte Statuscodes von HTTP.

Table 9.3: Status Codes of HTTP Servers

Code	Meaning	Description
200	`OK`	Request processed successfully. The Reply has a payload
202	`Accepted`	Request accepted but is executed later
204	`No Content`	Request executed successfully. The Reply has no payload
301	`Moved Permanently`	Resource moved. The old address is no longer valid
307	`Temporary Redirect`	Resource moved. The old address remains valid
400	`Bad Request`	Request has an incorrect syntax
401	`Unauthorized`	Request cannot be executed without authentication
403	`Forbidden`	Request not executed. Lack of privileges of the client
404	`Not Found`	Resource not found by the server
500	`Int. Server Error`	Unexpected server error

There are three protocol versions (`HTTP/1.0`, `HTTP/1.1` and `HTTP/2`). When communicating via `HTTP/1.0` (RFC 1945), a new TCP connection is established before each request to the server and closed by default by the server after the response has been transmitted (see Figure 9.5). If an HTML document contains references to 10 images, for example, 11 TCP connections are required for their transmission to the client.

Mit `HTTP/1.0`, `HTTP/1.1` und `HTTP/2` existieren drei Protokollversionen. Bei der Kommunikation via `HTTP/1.0` (RFC 1945) wird vor jeder Anfrage an den Server eine neue TCP-Verbindung aufgebaut und nach der Übertragung der Antwort standardmäßig vom Server wieder geschlossen (siehe Abbildung 9.5). Enthält ein HTML-Dokument Referenzen auf zum Beispiel 10 Bilder, sind also 11 TCP-Verbindungen zur Übertragung an den Client nötig.

Figure 9.5: Difference between HTTP/1.0 and HTTP/1.1

With `HTTP/1.1` (RFC 2616), the connection is not terminated by default. So the connection

Bei `HTTP/1.1` (RFC 2616) wird standardmäßig kein Verbindungsabbau durchgeführt. So

can be used for multiple transmissions. For the transfer of an HTML document with 10 images, only a single TCP connection is required. This accelerates the download of the document. Furthermore, `HTTP/1.1` allows to continue aborted transmissions.

`HTTP/2` (RFC 7540), which was published in May 2015 as the successor of `HTTP/1.1`, further accelerates data transmission by compressing the header with the algorithm HPACK (RFC 7541). Also, `HTTP/2` allows requests to be bundled (*Multiplex*). Furthermore, a server can send data on its own (*Server Push*) of which it is aware that it is immediately needed by the browser. Examples of such data include CSS (Cascading Style Sheets) files, which specify the layout of web pages, or script files.

Since `HTTP/2` is not a text-based, but a binary protocol, it is not possible to communicate over it by using tools such as Telnet or `nc`, for example, to examine a server.

kann die Verbindung immer wieder verwendet werden. Für den Transfer eines HTML-Dokuments mit 10 Bildern ist somit nur eine einzige TCP-Verbindung nötig. Dadurch wird das Dokument schneller geladen. Zudem können abgebrochene Übertragungen bei `HTTP/1.1` fortgesetzt werden.

Das im Mai 2015 als Nachfolger von HTTP/1.1 verabschiedete `HTTP/2` (RFC 7540) beschleunigt abermals die Datenübertragung unter anderem durch eine Kompression des Headers mit dem Algorithmus HPACK (RFC 7541). Zudem ermöglicht `HTTP/2` das Zusammenfassen (*Multiplex*) von Anfragen und ein Server kann von sich aus Daten senden (*Server Push*), von denen er weiß, dass sie der Browser umgehend benötigen wird. Beispiele für solche Daten sind CSS-Dateien (Cascading Style Sheets), die die Darstellung der Webseiten definieren, oder Script-Dateien.

Da `HTTP/2` kein textbasiertes, sondern ein binäres Protokoll ist, kann nicht mit einfachen Werkzeugen wie Telnet oder `nc` darüber kommuniziert werden, um beispielsweise einen Server zu untersuchen.

9.5

Simple Mail Transfer Protocol (SMTP)

SMTP (RFC 5321) is a protocol for the exchange (delivery) of emails via TCP. By default, the protocol uses port 25. Emails are fetched using the protocols POP3 or IMAP. For sending emails, the user's mail program connects to a SMTP server, which forwards emails to the destination, via further SMTP servers if necessary. Because SMTP is a text-based protocol, it is possible to connect to a SMTP server via Telnet or `nc` and send emails *manually*.

The sender and destination addresses are freely selectable with SMTP. The addresses in the commands `MAIL FROM-` and `RCPT TO` may differ from the addresses in the fields `From` and `To` in the email header. Authentication is not

Simple Mail Transfer Protocol (SMTP)

SMTP (RFC 5321) ist ein Protokoll, das den Austausch (Versand) von Emails via TCP ermöglicht. Standardmäßig verwendet das Protokoll Port 25. Das Abholen von Emails erfolgt mit den Protokollen POP3 oder IMAP. Zum Versand von Emails verbindet sich das Mailprogramm des Benutzers mit einem SMTP-Server, der die Emails ggf. über weitere SMTP-Server zum Ziel weiterversendet. Da SMTP ein textbasiertes Protokoll ist, kann man sich auch via Telnet oder alternativ via `nc` mit einem SMTP-Server verbinden und so auch Emails *von Hand* versenden.

Die Absender- und Empfängeradresse sind bei SMTP frei wählbar. Die Adressen im `MAIL FROM-` und `RCPT TO`-Kommando können sich von den Adressen in den Feldern `From` und `To` im Header der Email unterscheiden. Eine Authentifizierung findet nicht zwingend statt.

mandatory. Therefore, the sender designation in SMTP cannot be trusted.

In SMTP gibt es also keine Verlässlichkeit der Absenderangabe in Emails.

Table 9.4: Status Codes (Reply Codes) of SMTP Servers

Code	Description
2xx	Command successfully executed
4xx	Temporary failure. Executing the command may be successful later
5xx	Permanent failure. The command cannot be executed

An SMTP server replies to requests with three-digit status codes (see Table 9.4) and short texts that may vary or be omitted. Table 9.5 contains some SMTP commands.

Due to the lack of security features, the operation of a SMTP server is not without security risks. However, additional software exists that extends the security features of SMTP servers. The implementation of access control for assuring authenticity and integrity, as well as confidentiality via encryption can be achieved with SMTP-Auth, SMTPS, PGP, S/MIME, and SSL/TLS.

Ein SMTP-Server antwortet auf Anfragen mit dreistelligen Statuscodes (siehe Tabelle 9.4) und kurzen Texten, die variieren oder entfallen können. Tabelle 9.5 enthält einige SMTP-Kommandos.

Wegen der fehlenden Sicherheitsmerkmale ist der Betrieb eines SMTP-Servers nicht ohne Sicherheitsrisiken. Es existiert aber Zusatzsoftware, die die Funktionalität von SMTP-Servern erweitert. Die Realisierung von Zugangskontrolle, Sicherung von Echtheit und Integrität sowie Vertraulichkeit via Verschlüsselung können unter anderem mit SMTP-Auth, SMTPS, PGP, S/MIME und SSL/TLS realisiert werden.

Table 9.5: SMTP Commands

Command	Function
`HELO`	Start SMTP session and identify client
`MAIL From:<...>`	Enter the email address of the sender
`RCPT To:<...>`	Enter the email address of the receiver
`DATA`	Enter the content of an email
`RSET`	Abort to enter an email
`NOOP`	No operation. Keeps the connection (avoids timeouts)
`QUIT`	Log out from the SMTP server

9.6

Post Office Protocol Version 3 (POP3)

The protocol POP (RFC 918) allows to list, fetch and delete emails from an email server. It uses TCP and port 110 by default. Version 3 (POP3) is the latest, it was first published in 1988 (RFC 1081). The most up-to-date specification is contained in RFC 1939 from 1996.

Post Office Protocol Version 3 (POP3)

Das Protokoll POP (RFC 918) ermöglicht das Auflisten, Abholen und Löschen von Emails von einem Email-Server und verwendet TCP und standardmäßig Port 110. Die aktuelle Version ist Version 3 (POP3), die erstmals 1988 (RFC 1081) beschrieben wurde. Die aktuellste Beschreibung

With POP3, the entire communication is transmitted in plain text. Because it is a text-based protocol, it is possible to *manually* list, fetch and delete emails via Telnet or `nc` at a POP3 server. Table 9.6 contains some POP3 commands.

befindet sich in RFC 1939 von 1996. Die vollständige Kommunikation wird bei POP3 im Klartext übertragen. Da es sich um ein textbasiertes Protokoll handelt, kann man via Telnet oder alternativ via `nc` Emails auch *von Hand* auflisten, abholen und löschen. Tabelle 9.6 enthält einige POP3-Kommandos.

Table 9.6: POP3 Commands

Command	Function
`USER xxx`	Enter username
`PASS xxx`	Enter password
`STAT`	Print the total number and total size (in bytes) of emails in the mailbox
`LIST (n)`	Print the message numbers and size of all emails or a specific email
`RETR n`	Print a specific email from server
`DELE n`	Erase a specific email from server
`RSET`	Reset all `DELE` commands
`NOOP`	No operation. Keeps the connection (avoids timeouts)
`QUIT`	Disconnect from server and execute the `DELE` commands

9.7

File Transfer Protocol (FTP)

FTP (RFC 959) is a protocol that allows files to be downloaded from and uploaded to FTP servers. The entire communication via FTP takes place in plain text. This also includes the transmission of passwords.

If the FTP server and the client are connected, a TCP connection (*control port*) exists between them. This connection is used to send the control commands to the server. For this purpose, FTP uses port 21 by default. The server responds with a status code and usually with explanatory text to each command.

A separate TCP connection (*data port*) is established for each process. These connections are used to send and receive files and for the transfer of directory lists. FTP uses port 20 by default for this purpose. FTP connections can be established in *active mode* and *passive mode*.

In active FTP (*active mode*), the client opens a port with a port number that is greater than

File Transfer Protocol (FTP)

FTP (RFC 959) ist ein Protokoll, das das Herunterladen von Dateien von FTP-Servern und Hochladen zu diesen ermöglicht. Der vollständige Datenaustausch via FTP erfolgt im Klartext. Das schließt auch das Senden von Passwörtern mit ein.

Sind FTP-Server und -Client miteinander verbunden, besteht zwischen beiden eine TCP-Verbindung (*Control Port*). Über diese Verbindung werden die Steuerkommandos zum Server gesendet. Dafür verwendet FTP standardmäßig Port 21. Auf jedes Kommando antwortet der Server mit einem Statuscode und meistens noch mit einem erklärenden Text.

Für jeden Vorgang wird jeweils eine separate TCP-Verbindung (*Data Port*) aufgebaut. Diese Verbindungen werden zum Senden und Empfangen von Dateien und zur Übertragung von Verzeichnislisten verwendet. Dafür verwendet FTP standardmäßig Port 20. FTP-Verbindungen können im *aktiven Modus* und im *passiven Modus* aufgebaut werden.

Bei aktivem FTP (*Active Mode*) öffnet der Client einen Port mit einer Portnummer größer

1023 and communicates this and its IP address to the server using the `PORT` command. The server then establishes the FTP connection between its port 20 and the client's port. The server, therefore, initiates connection establishment.

In passive FTP (*passive mode*), the client sends the `PASV` command to the server. The server opens a port and sends a message to the client, in which it transmits the port number. The client opens a port with a port number that is greater than 1023 and establishes the FTP connection from its port to the server port. The client, therefore, initiates connection establishment. Passive FTP is used, for example, when the client is behind a firewall.

1023 und teilt diese und die eigene IP dem Server via `PORT`-Kommando mit. Der Server baut anschließend die FTP-Verbindung zwischen seinem Port 20 und dem Client-Port auf. Der Verbindungsaufbau geht also vom Server aus.

Bei passivem FTP (*Passive Mode*) sendet der Client das `PASV`-Kommando an den Server. Dieser öffnet einen Port und sendet eine Nachricht an den Client, in der er diesem die Portnummer mitteilt. Der Client öffnet einen Port mit einer Portnummer größer 1023 und baut die FTP-Verbindung von seinem Port zum Server-Port auf. Der Verbindungsaufbau geht also vom Client aus. Passives FTP wird unter anderem dann verwendet, wenn der Client hinter einer Firewall liegt.

Table 9.7: FTP Commands

Command	Function
`ABOR`	Abort the active file transfer
`CWD`	Change Working Directory
`DELE`	Erase file
`LIST`	Print information about a file or a directory
`NOOP`	No operation. Keeps the connection (avoids timeouts)
`PASS`	Enter password
`PASV`	Switch to passive FTP mode
`PORT`	Enter the IP and the port number for active mode
`PWD`	Print Working Directory
`QUIT`	Disconnect from the server
`SIZE`	Print the size of a file
`STAT`	Print the connection status
`USER`	Enter username

Table 9.8: Status Codes (Reply Codes) of FTP Servers

Code	Description
1xx	Command accepted, but its execution is not yet finished
2xx	Command successfully executed
3xx	Further information is required from the client
4xx	Temporary failure. Executing the command may be successful later
5xx	Permanent failure. The command cannot be executed

FTP servers respond with a status code to each command that provides information about the status of the communication between client and FTP server (see Table 9.8). Table 9.7 contains some FTP commands.

FTP-Server beantworten jedes Kommando mit einem Statuscode, der über den Status der Kommunikation zwischen Client und FTP-Server Auskunft gibt (siehe Tabelle 9.8). Tabelle 9.7 enthält einige FTP-Kommandos.

Because FTP is a text-based protocol, it is possible to work with FTP servers via Telnet or `nc`.

An encrypted alternative to FTP is the SSH File Transfer Protocol, which is also called Secure File Transfer Protocol (SFTP) and usually provided in addition to SSH at port number 22.

Da FTP ein textbasiertes Protokoll ist, kann man auch via Telnet oder alternativ via `nc` mit FTP-Servern arbeiten.

Eine verschlüsselte Alternative zu FTP ist das SSH File Transfer Protocol, das auch Secure File Transfer Protocol (SFTP) genannt wird und üblicherweise zusätzlich zu SSH auf Port 22 angeboten.

10

Network Virtualization

Network virtualization is a buzzword for various concepts that enable to combine or separate network resources into logical units. The benefits of network virtualization are independence from physical environments and flexibility, as well as greater security against data theft and human error.

Network virtualization variants are Virtual Private Networks (VPN) and Virtual Local Area Networks (VLAN).

Netzwerkvirtualisierung

Netzwerkvirtualisierung ist ein Schlagwort für unterschiedliche Ansätze, um Netzwerkressourcen zu logischen Einheiten zusammenzufassen oder aufzuteilen. Vorteile der Netzwerkvirtualisierung sind Unabhängigkeit von den physischen Gegebenheiten und Flexibilität sowie eine höhere Sicherheit gegenüber Datendiebstahl und menschlichen Fehlern.

Varianten der Netzwerkvirtualisierung sind Virtual Private Networks (VPN) und Virtual Local Area Networks (VLAN).

10.1

Virtual Private Networks (VPN)

VPNs are virtual private networks (logical subnets) within public networks (e.g., the Internet). A node can be physically connected to a public network, but via VPN, it is connected to a private network. Typical application scenarios for VPNs are:

- *Site-to-Site VPN* connects two sites to a single network (see Figure 10.1). A useful scenario consists of the integration of remote company locations into a corporate network.
- *Remote Access VPN* or *End-to-Site VPN* integrates a computer into a remote network (see Figure 10.2). The VPN client establishes a connection to the remote VPN gateway. A popular deployment scenario consists of the integration of home office

Virtual Private Networks (VPN)

VPNs sind virtuelle private Netze (logische Teilnetze) innerhalb öffentlicher Netze (zum Beispiel das Internet). Ein Teilnehmer kann physisch an einem öffentlichen Netz angeschlossen sein, ist jedoch via VPN einem privaten Netz zugeordnet. Häufige Einsatzszenarien von VPNs sind:

- *Site-to-Site VPN* verbindet zwei Standorte zu einem einzigen Netzwerk (siehe Abbildung 10.1). Ein sinnvolles Szenario ist die Integration entfernter Unternehmensstandorte in ein Firmennetz.
- *Remote Access VPN* oder *End-to-Site VPN* integriert einen Rechner in ein entferntes Netzwerk (siehe Abbildung 10.2). Der VPN-Client baut dabei eine Verbindung zum entfernten VPN-Gateway auf. Ein häufiges Einsatzszenario ist die Inte-

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_10

workstations into the LAN of a company or an authority so that all employees can work with the same security standards from home via the company network.

gration von Home-Office-Arbeitsplätzen in das LAN eines Unternehmens oder einer Behörde, damit alle Mitarbeiter mit identisch hohen Sicherheitsstandards auch von zuhause über das Firmennetz arbeiten können.

Figure 10.1: Site-to-Site VPN

Figure 10.2: Remote Access VPN or End-to-Site VPN

One benefit of VPN is the option of using encryption, which improves security. Via VPN, it is also possible to access the Internet not via the local Internet connection, but instead via the network that is assigned via Remote Access VPN. Depending on the application purpose, this does not only improve security but also increases freedom because it allows working anonymously in geographical regions where censorship restrictions are in place. Campus networks using WLAN are another useful area of application for VPN.

Ein Vorteil von VPN ist die Möglichkeit der Verschlüsselung, was zu einer höheren Sicherheit führt. Es besteht auch via VPN die Möglichkeit, Zugriffe ins Internet nicht über den lokalen Internetanschluss durchzuführen, sondern über das via Remote Access VPN zugeordnete Netz. Das führt je nach Anwendungszweck nicht nur zu einer höheren Sicherheit, sondern auch zu freierem und eventuell anonymem Arbeiten in Regionen mit Zensurbeschränkungen. Ein weiteres sinnvolles Einsatzgebiet von VPNs sind Campusnetzwerke mit WLAN.

10.1.1

Technical types of VPNs

VPN can be implemented on the Data Link Layer, the Network Layer, and the Transport Layer.

A *Layer-2-VPN* can be designed as a Site-to-Site VPN or as a Remote Access VPN and, for example, be based on the Point-to-Point Tunneling Protocol (PPTP) or the Layer 2 Tunneling Protocol (L2TP). Here the VPN gateways involved and the VPN clients encapsulate the frames, for example, PPP frames from connections via Modem, ISDN or DSL, by additional frame headers.

Technische Arten von VPNs

VPN kann auf der Sicherungsschicht, Vermittlungsschicht und auf der Transportschicht realisiert werden.

Ein *Layer-2-VPN* kann als Site-to-Site VPN oder Remote Access VPN konzipiert sein und zum Beispiel auf dem Point-to-Point Tunneling Protocol (PPTP) oder dem Layer 2 Tunneling Protocol (L2TP) basieren. Hierbei kapseln die beteiligten VPN-Gateways und -Clients die Netzwerkrahmen, zum Beispiel PPP-Rahmen von Modem-, ISDN- oder DSL-Anschlüssen, durch zusätzliche Rahmen-Header.

Layer-3-VPN is typically Site-to-Site VPN. Each participating site requires a VPN client software or a hardware solution (VPN Gateway). IP packets are encapsulated with a VPN protocol at one tunnel end, transmitted to the other tunnel end and unpacked there. A VPN tunnel is thus a virtual connection between two ends (sites). A typical protocol for *Layer-3-VPN* is Internet Protocol Security (IPsec).

A *Layer-4-VPN* is typically a Remote Access VPN based on the Transport Layer Security (TLS) protocol. TLS is also known as the older Secure Sockets Layer (SSL). It allows secure communication via TLS/SSL header, but not tunneling like *Layer-3-VPN*. A web browser is sufficient as client software.

IPsec, and thus *Layer-3-VPN*, is usually the basis for Site-to-Site VPN because Remote Access VPN requires a VPN client. TLS/SSL and thus *Layer-4-VPN* is usually the basis for Remote Access VPN because a web browser is sufficient as a client and no further client software is required.

Layer-3-VPN ist meist Site-to-Site VPN. Jeder teilnehmende Standort benötigt eine VPN-Client-Software oder Hardwarelösung (VPN-Gateway). IP-Pakete werden an Tunnelenden mit einem VPN-Protokoll gekapselt, zum anderen Tunnelende übertragen und dort entpackt. Ein VPN-Tunnel ist somit eine virtuelle Verbindung zwischen zwei Enden (Standorten). Ein verbreitetes Protokoll für *Layer-3-VPN* ist das Internet Protocol Security (IPsec).

Bei einem *Layer-4-VPN* handelt es sich meist um ein Remote Access VPN, das auf dem Protokoll Transport Layer Security (TLS) basiert. TLS ist auch unter der älteren Bezeichnung Secure Sockets Layer (SSL) bekannt. Es ermöglicht eine sichere Kommunikation via TLS/SSL-Header, aber kein Tunneling wie *Layer-3-VPN*. Als Client-Software genügt ein Webbrowser.

IPsec und damit *Layer-3-VPN* ist meist die Basis für Site-to-Site VPN, da Remote Access VPN zwingend einen VPN-Client erfordert. TLS/SSL und damit *Layer-4-VPN* ist meist die Basis für Remote Access VPN, da hier als Client ein Webbrowser genügt und keine weitere Software auf der Seite der Clients nötig ist.

10.2

Virtual Local Area Networks (VLAN)

Distributed devices can be combined to a single virtual (logical) network via VLAN. VLANs separate physical networks into logical subnets (overlay networks) because VLAN-enabled Switches do not forward packets from one VLAN to another. A VLAN thus forms an externally isolated network over existing networks. This makes it possible to consolidate devices and services, which belong together, into separate VLANs and thus improve security. One benefit of VLAN is that it does not affect other networks.

The two existing VLAN types are the outdated *static VLAN* and the more modern *dynamic VLAN* concept.

With static VLAN, the ports of a Switch are partitioned into logical Switches. Each port is permanently assigned to a VLAN or connects different VLANs. One example of a Switch

Virtual Local Area Networks (VLAN)

Verteilt aufgestellte Geräte können via VLAN in einem einzigen virtuellen (logischen) Netzwerk zusammengefasst werden. VLANs trennen physische Netze in logische Teilnetze (Overlay-Netze), denn VLAN-fähige Switches leiten Datenpakete eines VLAN nicht in ein anderes VLAN weiter. Somit bildet ein VLAN ein nach außen isoliertes Netz über bestehende Netze. Das macht es möglich, zusammengehörende Geräte und Dienste in eigenen VLANs zu konsolidieren und somit die Sicherheit zu verbessern. Ein Vorteil von VLAN ist, dass es andere Netze nicht beeinflusst.

Die beiden existierenden Typen von VLAN sind das veraltete *statische VLAN* und das modernere Konzept des *dynamischen VLAN*.

Beim statischen VLAN werden die Anschlüsse eines Switches in logische Switches unterteilt. Jeder Anschluss ist fest einem VLAN zugeordnet oder verbindet unterschiedliche VLANs. Ein Bei-

whose ports are distributed among two VLANs is shown in Figure 10.3. The first four ports are assigned to a VLAN, and the remaining four ports are assigned to another VLAN. One drawback of this concept is that it is difficult to automate.

spiel für einen Switch, dessen Schlüsse auf zwei VLANs aufgeteilt sind, ist in Abbildung 10.3 zu sehen. Dort sind die ersten vier Anschlüsse einem VLAN zugeordnet und die übrigen vier Anschlüsse einem anderen VLAN. Ein Nachteil dieses Konzepts ist, dass es schlecht automatisierbar ist.

Figure 10.3: Static VLAN

For dynamic VLAN according to IEEE 802.1Q, the frames contain a particular VLAN tag. Before standardization by the IEEE in 1998, proprietary packet-based VLAN solutions existed, such as the Cisco Inter-Switch Link (ISL) and 3Com's Virtual LAN Trunk (VLT). These solutions were merged later into the IEEE 802.1Q specification. One benefit of dynamic VLAN is that it can be created, modified and removed purely by software, using scripts.

Figure 6.16 in Section 6.4 shows an Ethernet frame (IEEE 802.3) with VLAN tag (IEEE 802.1Q). The VLAN tag includes 32 bits and is subdivided into three fields (see Figure 10.4).

Bei dynamischem VLAN nach IEEE 802.1Q enthalten die Netzwerkrahmen eine spezielle VLAN-Markierung (*Tag*). Vor der Standardisierung 1998 durch das IEEE Konsortium existierten proprietäre paketbasierte VLAN-Lösungen wie der Cisco Inter-Switch Link (ISL) und Virtual LAN Trunk (VLT) von 3Com. Diese Lösungen wurden später im Standard IEEE 802.1Q zusammengefasst. Ein Vorteil von dynamischem VLAN ist, dass es mit Hilfe von Skripten rein softwaremäßig erzeugt, verändert und entfernt werden kann.

Abbildung 6.16 in Abschnitt 6.4 zeigt einen Ethernet-Rahmen (IEEE 802.3) mit VLAN-Tag (IEEE 802.1Q). Die VLAN-Markierung umfasst 32 Bits und ist in drei Datenfelder unterteilt (siehe Abbildung 10.4).

Figure 10.4: VLAN Tag inside an Ethernet Frame

The protocol ID (16 bits) always has the value `0x8100`. Three bits represent priority (QoS). The value `0` indicates the lowest and the value `7`

Die Protokoll-ID (16 Bits) hat immer den Wert `0x8100`. Drei Bits repräsentieren die Priorität (QoS). Der Wert `0` steht für die niedrigste

the highest priority. This allows to prioritize specific data (e.g., VoIP). The canonical format (one bit) specifies the most significant bit of the MAC addresses. This field has the value `0` for Ethernet and the value `1` for Token Ring. The remaining 12 bits contain the ID of the VLAN to which the packet inside the frame belongs to. With 12 bits, 4,096 different VLANs can be addressed.

Dynamic VLAN is often used in data centers and small office/home office scenarios. Separating the data traffic according to economic aspects is useful. Furthermore, VLAN protects against operational errors and defective software. For example, it is possible to set up a VLAN as "production network" with the important services and an additional VLAN for experiments, project work or computer games.

In cloud computing infrastructure services such as the free solution Eucalyptus [30], each virtual machine (instance) is assigned to a so-called security group. Each security group has its own firewall rules. An infrastructure service based on Eucalyptus can create a separate VLAN for each security group with the intention to isolate the data traffic of the instances according to the security groups.

Another field of application is IPTV. With Telekom Entertain, for example, customers get a fixed line DSL connection and IPTV. The system uses two VLANs to prioritize IPTV traffic. The "normal" Internet traffic is transmitted via PPPoE by using the VLAN with the ID 7, and the IPTV without dial-in uses the VLAN with the ID 8.

und der Wert `7` für die höchste Priorität. Damit können bestimmte Daten (z. B. VoIP) priorisiert werden. Das kanonisches Format (ein Bit) legt fest, welches das höchstwertiges Bit der MAC-Adressen ist. Bei Ethernet hat dieses Datenfeld den Wert `0` und bei Token Ring den Wert `1`. Die übrige 12 Bits enthalten die ID des VLAN, zu dem das Paket im Rahmen gehört. Mit 12 Bits können 4.096 unterschiedliche VLANs adressiert werden.

Häufige Einsatzgebiete von dynamischem VLAN sind Rechenzentren oder auch Büros zuhause. Dabei findet sinnvollerweise eine Trennung des Datenverkehrs nach ökonomischen Gesichtspunkten statt. Zudem ermöglicht VLAN eine Absicherung vor Bedienfehlern und fehlerhafter Software. Beispielsweise bietet sich die Einrichtung eines VLAN als „Produktionsnetz“ mit den wichtigen Diensten und ein zusätzliches VLAN für Experimente, Projektarbeiten oder Computerspiele an.

Bei Cloud-Infrastrukturdiensten wie zum Beispiel der freien Lösung Eucalyptus [30], ist jede virtuelle Maschine (Instanz) einer sogenannten Sicherheitsgruppe zugeordnet. Jede Sicherheitsgruppe hat eigene Firewall-Regeln. Ein Infrastrukturdienst auf Basis von Eucalyptus kann für jede Sicherheitsgruppe ein eigenes VLAN anlegen, mit dem Ziel der Isolation des Datenverkehrs der Instanzen anhand der Sicherheitsgruppen.

Ein weiteres Einsatzgebiet ist IPTV. Bei Telekom Entertain zum Beispiel erhalten die Kunden einen DSL-Anschluss mit Festnetzanschluss und IPTV. Das System verwendet zwei VLANs, um den IPTV-Datenverkehr zu bevorzugen. Das „normale“ Internet wird via PPPoE über das VLAN mit der ID 7 übertragen und das IPTV ohne Einwahl über das VLAN mit der ID 8.

11

Command Line Tools

Numerous tools exist for standard operating systems to configure computer networks and analyze network-related issues. This chapter provides an overview of essential command line tools for Linux/UNIX operating systems. These tools often are available under the same or similar names in Mac OS X or Microsoft Windows, or they can be installed with little effort. Table 11.1 allocates these tools to the layers of the hybrid reference model (see Figure 4.3).

All tools that are described in this chapter provide a large number of functions. Accordingly, their behavior can be influenced by numerous command line parameters. A complete listing of these functions and the corresponding command line parameters cannot be given in the context of this work. Only a brief selection of concrete application scenarios and the most important functions can be presented. For a detailed discussion of the existing command line parameters, it is essential to consult the help and documentation pages (*man pages*) of the command line tools.

With command line tools of the physical layer, such as `ethtool` and `mii-tool`, the status of the connection, as well as various connection parameters (including data rate and connection mode), are monitored and adjusted. Tools for WLAN networks such as `iwlist` and `iwconfig` also make it possible to identify wireless networks and to connect to these. For establishing a connection with a WLAN network that only uses the insecure encryption standard WEP (see

Kommandozeilenwerkzeuge

Zur Netzwerkkonfiguration und Analyse von Netzwerkproblemen existieren für die verbreiteten Betriebssysteme zahlreiche Werkzeuge. In diesem Kapitel finden Sie eine Übersicht über wichtige Kommandozeilenwerkzeuge für Linux/UNIX-Betriebssysteme. Diese sind häufig unter gleichem oder ähnlichem Namen auch unter Mac OS X oder Microsoft Windows verfügbar oder können mit geringem Aufwand nachinstalliert werden. Tabelle 11.1 enthält eine Zuordnung der Werkzeuge zu den Schichten des hybriden Referenzmodells (siehe Abbildung 4.3).

Alle in diesem Kapitel vorgestellten Werkzeuge bieten eine große Anzahl an Funktionen. Dementsprechend kann ihr Verhalten mit zahlreichen Kommandozeilenparametern beeinflusst werden. Eine vollständige Auflistung dieser Funktionen und der entsprechenden Kommandozeilenparameter ist im Rahmen dieses Werkes nicht möglich. Es können nur eine knappe Auswahl an konkreten Einsatzszenarien und die wichtigsten Funktionen vorgestellt werden. Für eine detaillierte Auseinandersetzung mit den verfügbaren Kommandozeilenparametern ist eine Recherche der Hilfe- und Dokumentationsseiten (*Man-Pages*) der Kommandozeilenwerkzeuge unerlässlich.

Mit Kommandozeilenwerkzeugen der Bitübertragungsschicht wie z. B. `ethtool` und `mii-tool` werden der Verbindungsstatus sowie diverse Verbindungsparameter (u. a. die Datenrate und der Verbindungsmodus) kontrolliert und angepasst. Werkzeuge für WLAN-Funknetze wie z. B. `iwlist` und `iwconfig` ermöglichen zudem das Auffinden gewünschter Funknetze und das Verbinden mit diesen. Für den Aufbau einer Verbindung mit einem WLAN-

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0_11

Table 11.1: Network Configuration and Analysis Tools for Linux/UNIX

Layers[a] 1	2	3	4	5	Tool	Purpose
X					`ethtool`	Analyze status and the configuration of network interfaces on layer 1
X					`mii-tool`	Identical application purpose as `ethtool`, but smaller range of functions
X	X				`iwconfig`	Analyze and modify the configuration of WLAN network interfaces
X	X				`iwlist`	List reachable WLAN networks and their connection parameters
	X	X			`arp`	Return and modify the entries in the local ARP cache
	X	X			`ifconfig`	Enable/disable network interfaces, check and modify addresses of the layers 2 and 3
	X	X			`ip`	Replaces the tools `arp`, `ifconfig`, `netstat` and `route`
	X	X	X	X	`iftop`	Monitor network traffic
	X	X	X	X	`tcpdump`	Monitor and analyze the network traffic
		X			`ping`	Check accessibility of a system using ICMP echo requests
		X			`route`	Return and modify the entries in the routing table
		X			`traceroute`	Identify Routers that deliver IP packets until they reach their destination
		X	X		`netstat`	Return statistical data, list open ports, active connections and the routing table
			X		`nmap`	Check port numbers for open network services and operating systems version
				X	`dhclient`	Request a network configuration from DHCP servers in reach
				X	`dig`	Perform a domain name resolution and return DNS server replies
				X	`ftp`	Download and upload files to and from FTP servers
				X	`netperf`	Measure data rate during a specified period of time
				X	`nc`	Analyze network services
				X	`nslookup`	Perform a domain name resolution and return DNS server replies
				X	`ssh`	Encrypted remote control of computers
				X	`telnet`	Unencrypted remote control of computers and analysis of network services

[a] Layers in the hybrid reference model.

Section 5.1.3), `iwconfig` is sufficient. If wireless networks that are protected via WPA or WPA2 shall be used, `wpa_supplicant` is also required.

Tools of the Data Link Layer allow to control the physical addresses (MAC addresses) and in some cases to modify them for a certain period of time. Also, some tools (e.g., `ifconfig` and `ip`) can enable and disable network interfaces at the Data Link Layer level.

With tools of the Network Layer (e.g., `ifconfig` and `ip`), the logical addresses (IP addresses) are checked and manually assigned, and the local routing table (e.g., with `route` or `ip`) can be adapted. Tools such as `arp` for checking and modifying entries in the local ARP cache do also belong to this layer.

Tools for the Transport Layer, such as `netstat` and `nmap` allow to control the currently active services of a specific server, and therefore the monitoring of reachable port numbers.

Tools for the Application Layer are often client applications that use Application Layer protocols to interact with network services. In this context, among others, name resolution (e.g., with `nslookup`), automatic assignment of IP addresses via DHCP (e.g., with `dhclient`), transfer of files (e.g., via `ftp`) and remote control of computers (e.g., via `telnet` or `ssh`) take place.

One reason why this chapter is limited to command line tools is that their names and the way they are used are mostly similar or even identical, no matter what Linux distribution, Microsoft Windows variant or Mac OS X version is used. Also, these tools rarely require the installation of additional software packages because they are included in the operating systems by default. Another benefit is that these command line tools do not require a graphical user interface. Therefore they can be applied easily via Secure Shell (SSH) or Telnet (see section 9.3) on remote computers for administration and troubleshooting.

Funknetz, das nur den unsicheren Sicherheitsstandard WEP (siehe Abschnitt 5.1.3) verwendet, genügt `iwconfig`. Sollen Funknetze verwendet werden, die via WPA oder WPA2 abgesichert sind, ist zusätzlich `wpa_supplicant` nötig.

Werkzeuge der Sicherungsschicht ermöglichen u. a. die Kontrolle der physischen Adressen (MAC-Adressen) und teilweise auch deren temporäre Änderung. Zudem können einige Werkzeuge (z. B. `ifconfig` und `ip`) Netzwerkschnittstellen auf der Ebene der Sicherungsschicht aktivieren und deaktivieren.

Mit Werkzeugen der Vermittlungsschicht (z. B. `ifconfig` und `ip`) werden die logischen Adressen (IP-Adressen) kontrolliert, manuell zugewiesen und die lokale Routing-Tabelle (z. B. mit `route` oder `ip`) angepasst. Dieser Schicht zugeordnet sind auch Werkzeuge wie `arp` zur Kontrolle und Modifikation der Einträge im lokalen ARP-Cache.

Werkzeuge für die Transportschicht wie z. B. `netstat` und `nmap` ermöglichen die Kontrolle der laufenden Serverdienste und damit die Beobachtung der erreichbaren Portnummern.

Die Werkzeuge der Anwendungsschicht sind häufig Client-Anwendungen, die Anwendungsprotokolle verwenden, um mit Serverdiensten zu interagieren. In diesem Kontext findet u. a. die Namensauflösung (z. B. mit `nslookup`), die automatische Zuweisung von IP-Adressen via DHCP (z. B. mit `dhclient`), der Transfer von Dateien (z. B. via `ftp`) und die Fernsteuerung von Rechnern (z. B. via `telnet` oder `ssh`) statt.

Ein Grund, warum sich dieses Kapitel auf die Vorstellung der Kommandozeilenwerkzeuge beschränkt, ist, dass deren Namen und die Art und Weise der Anwendung, unabhängig von konkreten Linux-Distributionen, Varianten von Microsoft Windows und Versionen von Mac OS X, meist ähnlich oder sogar gleich sind. Zudem erfordern diese Werkzeuge nur selten die Installation zusätzlicher Softwarepakete, weil sie standardmäßig in den Betriebssystemen enthalten sind. Ein weiterer Vorteil ist, dass die Kommandozeilenwerkzeuge keine grafische Oberfläche voraussetzen. Darum können sie ohne Probleme via Secure Shell (SSH) oder Telnet (siehe Abschnitt 9.3) auf entfernten Rechnern zur Administration und Problemanalyse verwendet werden.

This chapter continues with an explanatory section for each tool that is mentioned in Table 11.1. The ordering of the sections corresponds to the ordering of the entries in the table.

Im weiteren Verlauf enthält dieses Kapitel zu jedem in Tabelle 11.1 aufgeführten Werkzeug einen erklärenden Abschnitt. Die Reihenfolge der Abschnitte entspricht der Reihenfolge der Einträge in der Tabelle.

11.1

ethtool

The configuration parameters of local Ethernet interfaces can be examined and modified at the Physical Layer by using the command line tool `ethtool`. This allows to determine the connection status (connected or disconnected) and the supported data rates, as well as to specify a data rate. It can also inform whether a connection is in full duplex or half duplex mode (see section 3.4.3). This parameter can be set manually. `ethtool` can also print out information about the driver and the firmware, as well as statistical parameters on the data that has been sent and received via a specific Ethernet interface.

If it is difficult to allocate the matching interface of the operating system for a physical Ethernet interface, it is helpful to manually make the LED of the physical interface blink with `ethtool`.

With `ethtool`, it is also possible to query the MAC address of local Ethernet interfaces. With the tools `ifconfig` and `ip`, the temporary modification of the MAC address is possible.

Table 11.2 contains some examples for using `ethtool`. `eth0` serves as an example of an Ethernet interface.

ethtool

Die Einstellungen lokaler Ethernet-Schnittstellen können auf der Ebene der Bitübertragungsschicht mit dem Kommandozeilenwerkzeug `ethtool` untersucht und verändert werden. Damit ist es u. a. möglich, den Verbindungsstatus (verbunden oder nicht verbunden) und die unterstützten Datenraten zu kontrollieren sowie eine Datenrate festzulegen. Es kann auch informieren, ob eine Verbindung im Modus Vollduplex oder Halbduplex (siehe Abschnitt 3.4.3) ist. Diese Einstellung kann manuell festgelegt werden. Auch Informationen zum Treiber und zur Firmware sowie statistische Daten zu den über eine bestimmte Ethernet-Schnittstelle gesendeten und empfangenen Daten kann `ethtool` ausgeben.

Kann eine physische Ethernet-Schnittstelle nicht sicher einer Schnittstelle im Betriebssystem zuordnet werden, ist es hilfreich, die LED der physischen Schnittstelle mit `ethtool` manuell zum Blinken bringen.

Mit `ethtool` ist es auch möglich, die MAC-Adresse lokaler Ethernet-Schnittstellen abzufragen und mit den Werkzeugen `ifconfig` und `ip` ist die temporäre Veränderung der MAC-Adresse möglich.

Tabelle 11.2 enthält einige Aufrufbeispiele für `ethtool`. Dabei dient `eth0` als Beispiel für eine Ethernet-Schnittstelle.

Table 11.2: Examples for using `ethtool`

Command	Function
`ethtool eth0`	Query the status and settings on the Physical Layer
`ethtool -P eth0`	Query the MAC address
`ethtool -S eth0`	Query statistical parameters
`ethtool -i eth0`	Query information about the driver and firmware
`ethtool -p eth0`	Let the LED blink

11.2

mii-tool

With the command line tool `mii-tool`, it is also possible to check and customize the connection status (connected or disconnected), as well as the connection mode (full duplex or half duplex) and the data rates of network interfaces. The tool is considered obsolete and offers less functionality than `ethtool`.

mii-tool

Auch mit dem Kommandozeilenwerkzeug `mii-tool` ist es möglich, den Verbindungsstatus (verbunden oder nicht verbunden) sowie den Verbindungsmodus (Vollduplex oder Halbduplex) und die Datenraten von Netzwerkschnittstellen zu kontrollieren und anzupassen. Das Werkzeug gilt als veraltet und bietet einen geringeren Funktionsumfang als `ethtool`.

11.3

iwconfig

If the command `iwconfig` is executed without further command line parameters, WLAN-specific information on all detected network interfaces is returned. Among other things, its output informs about whether a network interface supports WLAN at all. If this is the case, the network name (SSID) and information about link quality and data rate are returned. Furthermore, `iwconfig` informs whether the interface is connected to an access point. If this is the case, the output also contains their MAC address.

With `iwconfig`, it is not only possible to query the mentioned settings of WLAN interfaces. They can also be modified by this tool. The following command assigns the SSID `wlan0` to the network interface `My Network` so that the network interface connects to the access point with the matching SSID.

iwconfig

Wird `iwconfig` ohne weitere Kommandozeilenparameter aufgerufen, gibt es WLAN-spezifische Informationen zu allen erkannten Netzwerkschnittstellen aus. Die Ausgabe informiert u. a. darüber, ob eine Netzwerkschnittstelle überhaupt WLAN unterstützt. Ist das der Fall, werden u. a. der Netzwerkname (SSID) und Informationen zur Verbindungsqualität und Datenrate ausgegeben. Zudem informiert `iwconfig` darüber, ob die Schnittstelle mit einer Basisstation (Access Point) verbunden ist. Wenn ja, enthält die Ausgabe auch deren MAC-Adresse.

Mit `iwconfig` ist es nicht nur möglich, die genannten Einstellungen der WLAN-Netzwerkschnittstellen zu kontrollieren. Diese lässt sich durch das Werkzeug auch anpassen. Das folgende Kommando weist der Netzwerkschnittstelle `wlan0` die SSID `Mein Netzwerk` zu, damit sich die Netzwerkschnittstelle an der Basisstation mit der entsprechenden SSID anmeldet.

```
iwconfig wlan0 essid "My Network"
```

To modify settings, it is necessary to execute `iwconfig` with system administrator (*root*) privileges.

Um Einstellungen zu verändern, ist es nötig, `iwconfig` mit den Rechten des Systemadministrators (*root*) aufzurufen.

11.4

iwlist

The command line tool `iwlist` returns a list of all WLAN networks in reach. For each access point, among others, the following information is given: MAC address of access point, channel used (the frequency), network name (SSID), signal strength, encryption algorithms offered and data rates supported.

Table 11.3 contains some examples of using `iwlist`. `wlan0` serves as an example of a WLAN interface. To obtain an up-to-date list of WLAN networks in reach, the tool must be executed with system administrator (*root*) privileges.

iwlist

Das Kommandozeilenwerkzeug `iwlist` gibt eine Liste aller erreichbaren WLAN-Funknetze aus. Für jede Basisstation gibt es u. a. folgende Informationen aus: MAC-Adresse der Basisstation, verwendeter Kanal (die Frequenz), Netzwerkname (SSID), Signalstärke, angebotene Verschlüsselungsalgorithmen und unterstützte Datenraten.

Tabelle 11.3 enthält einige Aufrufbeispiele für `iwlist`. Dabei dient `wlan0` als Beispiel für eine WLAN-Schnittstelle. Um eine aktuelle Liste aller erreichbaren WLAN-Funknetze auszugeben, muss das Werkzeug mit den Rechten des Systemadministrators (*root*) aufgerufen werden.

Table 11.3: Examples for using `iwlist`

Command	Function
`iwlist wlan0 scan`	Query the list of WLAN networks in range
`iwlist wlan0 freq`	Query the supported frequencies and the currently used frequency
`iwlist wlan0 rate`	Query the current data rate
`iwlist wlan0 txpower`	Query the current transmit power

11.5

arp

With `arp` it is possible to print out the content of the local ARP cache (see section 6.10) and to insert entries into the ARP cache. This command returns the local ARP cache without resolving the hostnames.

arp

Mit `arp` ist es möglich, den lokalen ARP-Cache (siehe Abschnitt 6.10) auszugeben sowie Eintragungen in den ARP-Cache vorzunehmen. Das folgende Kommando gibt den lokalen ARP-Cache aus, ohne die Hostnamen aufzulösen.

```
arp -n
```

In Linux, the command line tool `arp` is increasingly being replaced by `ip`.

Das Kommandozeilenwerkzeug `arp` wird unter Linux zunehmend durch `ip` ersetzt.

11.6

ifconfig

The command line tool `ifconfig` (*interface configurator*) allows to enable and disable network interfaces and specify the IP address, netmask, and broadcast address. Also, `ifconfig` can temporarily alter the MAC address. That is why this tool not only works on the Network Layer but also on the Data Link Layer.

The following command assigns the IP address 10.0.0.2 to the network interface `eth0` and specifies that the subnet used is an entire class C subnet (netmask = 255.255.255.0) with the broadcast address 10.0.0.255.

ifconfig

Das Kommandozeilenwerkzeug `ifconfig` (*interface configurator*) ermöglicht es, Netzwerkschnittstellen zu aktivieren und zu deaktivieren sowie die IP-Adresse und die Netzmaske und die Broadcast-Adresse festzulegen. Zudem kann `ifconfig` die MAC-Adresse temporär ändern. Darum arbeitet dieses Werkzeug nicht nur auf der Vermittlungsschicht, sondern auch auf der Sicherungsschicht.

Das folgende Kommando weist der Netzwerkschnittstelle `eth0` die IP-Adresse 10.0.0.2 zu und legt fest, dass es sich bei dem verwendeten Subnetz um ein vollständiges Klasse-C-Subnetz (Netzmaske = 255.255.255.0) mit der Broadcast-Adresse 10.0.0.255 handelt.

```
ifconfig eth0 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255
```

Table 11.4 contains some examples for `ifconfig`. `eth0` serves as an example for an Ethernet interface.

Tabelle 11.4 enthält einige Aufrufbeispiele für `ifconfig`. Dabei dient `eth0` als Beispiel für eine Ethernet-Schnittstelle.

Table 11.4: Examples for using `ifconfig`

Command	Function
`ifconfig`	Query status of all active local network interfaces
`ifconfig -a`	Query status of all local network interfaces
`ifconfig eth0`	Query status of a specific network interface
`ifconfig eth0 up`	Enable network interface
`ifconfig eth0 down`	Disable network interface
`ifconfig eth0 hw ether <MAC>`	Temporary modify the MAC address

In Linux, the command line tool `ifconfig` is increasingly being replaced by `ip`.

Microsoft Windows includes the command line tool `ipconfig`, which offers a similar range of functions as `ifconfig`.

Das Kommandozeilenwerkzeug `ifconfig` wird unter Linux zunehmend durch `ip` ersetzt.

Unter Microsoft Windows existiert mit `ipconfig` ein Kommandozeilenwerkzeug, das einen ähnlichen Funktionsumfang wie `ifconfig` bietet.

11.7

ip

The command line tool `ip` is intended to replace the established tools `arp`, `ifconfig`, `netstat`,

ip

Zukünftig soll das Kommandozeilenwerkzeug `ip` die etablierten Werkzeuge `arp`, `ifconfig`, `netstat` und `route` ersetzen. Es erweitert de-

and `route`. It extends their range of functions and combines them into a single command.

ren Funktionsumfang und fasst ihn in einem einzigen Kommando zusammen.

Table 11.5: Examples for using `ip`

Command	Function
`ip link show dev eth0`	Query status (e. g., Maximum Transmission Unit) of a specific network interface and its address of the Data Link Layer
`ip link set dev eth0 up`	Enable network interface
`ip link set dev eth0 down`	Disable network interface
`ip link set dev eth0 mtu <Bytes>`	Specify the Maximum Transmission Unit (MTU) in Bytes
`ip link set dev eth0 addr <MAC>`	Temporary modify the MAC address
`ip neigh`	Query local ARP cache
`ip neigh flush all`	Erase all records of local ARP cache
`ip addr`	Query status (e. g., MTU) of all local network interfaces and of the addresses the Data Link Layer and the Network Layer
`ip addr show dev eth0`	Query status (e. g., MTU) of a specific network interface and the addresses of the Data Link Layer and the Network Layer
`ip route`	Query routing table
`ip route add default via <IP>`	Set IP address of a Router as the default route

With `ip` it is possible to check and modify the addresses of the network interfaces on the Data Link Layer and the Network Layer. This tool can also print out the local ARP cache and routing table and modify their entries.

Mit `ip` ist es u. a. möglich, die Adressen der Netzwerkschnittstellen auf der Sicherungsschicht und auf der Vermittlungsschicht zu kontrollieren und zu verändern. Zudem kann das Werkzeug den lokalen ARP-Cache und die Routing-Tabelle ausgeben und deren Einträge verändern.

The following command assigns the IP address 10.0.0.2 to the network interface `eth0` and specifies that the subnet used as an entire class C subnet (netmask = 255.255.255.0) with the broadcast address 10.0.0.255. The broadcast address is the default address and does not need to be specified explicitly in such a case.

Das folgende Kommando weist der Netzwerkschnittstelle `eth0` die IP-Adresse 10.0.0.2 zu und legt fest, das es sich bei dem verwendeten Subnetz um ein vollständiges Klasse-C-Subnetz (Netzmaske = 255.255.255.0) mit der Broadcast-Adresse 10.0.0.255 handelt. Die Broadcast-Adresse entspricht dem Standard und muss in einem solchen Fall nicht explizit angegeben werden.

```
ip addr add 10.0.0.2/24 broadcast 10.0.0.255 dev eth0
```

Table 11.5 contains some examples for using `ip`. `eth0` serves as an example of a network interface.

The `ip` tool can also be used to set up and remove Virtual Local Area Networks (see section 10.2).

Tabelle 11.5 enthält einige Aufrufbeispiele für `ip`. Dabei dient `eth0` als Beispiel für eine Netzwerkschnittstelle.

Das Werkzeug `ip` kann auch dazu verwendet werden, Virtual Local Area Networks (siehe Abschnitt 10.2) einzurichten und zu entfernen.

11.8

iftop

The command line tool `iftop` allows the real-time monitoring of network traffic. The following command causes `iftop` to record all network connections of the network interface `eth0` in real time and to return detailed information about the traffic sent and received.

iftop

Das Kommandozeilenwerkzeug `iftop` ermöglicht die Überwachung des Datenverkehrs in Echtzeit. Das folgende Kommando führt dazu, dass `iftop` alle Netzwerkverbindungen der Netzwerkschnittstelle `eth0` in Echtzeit erfasst und detaillierte Informationen zum gesendeten und empfangenen Datenverkehr ausgibt.

```
iftop -i eth0
```

11.9

tcpdump

The command line tool `tcpdump` allows the monitoring of the traffic. It is a useful tool for analyzing network issues and for discovering unwanted connections. The following command causes `tcpdump` to return information about the entire traffic that is sent and received via the network interface `eth0`.

tcpdump

Das Kommandozeilenwerkzeug `tcpdump` ermöglicht die Überwachung des Datenverkehrs. Es ist ein hilfreiches Werkzeug zur Analyse von Netzwerkproblemen sowie zur Entdeckung unerwünschter Verbindungen. Das folgende Kommando führt dazu, dass `tcpdump` Informationen zum kompletten Datenverkehr ausgibt, der über die Netzwerkschnittstelle `eth0` gesendet und empfangen wird.

```
tcpdump -i eth0
```

The tool provides several command line parameters for filtering traffic. The following command returns information about the traffic that is sent to and received from a specified port number of a particular address.

Das Werkzeug stellt zahlreiche Kommandozeilenparameter zur Verfügung, um den Datenverkehr zu filtern. Das folgende Kommando gibt Informationen über den Datenverkehr aus, der zu einer angegeben Portnummer einer bestimmten Adresse geht und von dieser kommt.

```
tcpdump port <portnumber> and host <address> -i eth0 -enq
```

By default, the output of `tcpdump` contains information from the headers of the Network Layer packets and the Transport Layer segments. The command line parameter `-e` indicates that the tool also returns information from the headers of the Physical Layer frames.

An IP address or a hostname can be specified as the address for the destination to be inspected. The command line parameter `-n` indicates that hostnames are not resolved in

Standardmäßig enthält die Ausgabe von `tcpdump` Informationen aus den Headern der Pakete der Vermittlungsschicht und Segmente der Transportschicht. Der Kommandozeilenparameter `-e` weist an, dass das Werkzeug auch Informationen aus den Headern der Rahmen der Bitübertragungsschicht ausgibt.

Als Adresse für das zu überprüfende Ziel kann eine IP-Adresse oder ein Hostname angegeben werden. Der Kommandozeilenparameter `-n` weist an, dass Hostnamen in der Ausgabe nicht

the output. The command line parameter `-q` indicates that the tool returns less protocol information, which shortens the length of the lines in the output.

If the output is written into a file by using the parameter `-w <File>`, it can be opened later with graphical tools Wireshark, for example, to analyze network traffic.

aufgelöst werden. Der Kommandozeilenparameter `-q` weist an, dass weniger Protokollinformationen ausgeben werden, was die Zeilenlänge der Ausgabe verkürzt.

Wird mit dem Parameter `-w <Datei>` die Ausgabe in eine Datei geschrieben, kann diese später zum Beispiel mit dem grafischen Werkzeugen Wireshark geöffnet werden, um den Datenverkehr zu analysieren.

11.10

route

The `route` command returns the routing table. This tool can also be used to insert and remove entries from the routing table. The following command prints out the routing table without resolving the hostnames.

route

Das Kommando `route` kann die Routing-Tabelle ausgeben. Zudem ist es mit diesem Werkzeug möglich, Einträge in die Routing-Tabelle einzufügen und aus dieser zu entfernen. Das folgende Kommando gibt die Routing-Tabelle aus, ohne die Hostnamen aufzulösen.

```
route -n
```

The following command specifies a default route. In this example, the Router has the IP address 10.0.0.1.

Eine Standardroute legt das nachfolgende Kommando fest. In diesem Beispiel hat der Router die IP-Adresse 10.0.0.1.

```
route add default gw 10.0.0.1
```

In Linux, the command line tool `route` is increasingly being replaced by `ip`.

Das Kommandozeilenwerkzeug `route` wird unter Linux zunehmend durch `ip` ersetzt.

11.11

ping

The command line tool `ping` sends one or more ICMP echo requests. This way it is possible to find out whether a destination system can be reached. An IP address or a hostname can be specified as the address for the destination to be inspected.

By default, `ping` constantly sends one ICMP echo request per second to the specified destination address. The number of ICMP echo requests to be sent can be specified as well as the network interface to be used, and the interval, i.e., the pause between the ICMP echo requests.

ping

Das Kommandozeilenwerkzeug `ping` sendet eine oder mehrere ICMP Echo-Anfragen. Damit kann untersucht werden, ob ein Zielsystem erreichbar ist. Als Adresse für das zu überprüfende Ziel kann eine IP-Adresse oder ein Hostname angegeben werden.

Standardmäßig sendet `ping` kontinuierlich eine ICMP Echo-Anfrage pro Sekunde an die angegebene Zieladresse. Die Anzahl der zu sendenden ICMP Echo-Anfragen kann genauso wie die zu verwendende Netzwerkschnittstelle und das Intervall, also die Pause zwischen den ICMP Echo-Anfrage angegeben werden.

The following command sends an ICMP echo request every two seconds via the network interface `wlan0` to the web server of publisher Springer. The command sends a total of five ICMP echo requests.

Das folgende Kommando sendet alle zwei Sekunden eine ICMP Echo-Anfrage über die Netzwerkschnittstelle `wlan0` an den Webserver des Springer-Verlages. Insgesamt sendet das Kommando fünf ICMP Echo-Anfragen.

```
ping -c 5 -i 2 -I wlan0 www.springer.de
```

`ping6` is a version of `ping` that is compatible with the Network Layer protocol IPv6.

Mit `ping6` existiert eine zum Vermittlungsprotokoll IPv6 kompatible Version von `ping`.

11.12

traceroute

The `traceroute` tool determines via which Routers IP packets are forwarded to the destination address. An IP address or a hostname can be specified as the destination address. The following command identifies the route from the Router of the local network to the web server of publisher Springer.

traceroute

Das Werkzeug `traceroute` ermittelt, über welche Router IP-Pakete bis zur Zieladresse weitergeleitet werden. Als Adresse für das zu überprüfende Ziel kann eine IP-Adresse oder ein Hostname angegeben sein. Das folgende Kommando ermittelt den Weg vom Router des lokalen Netzes zum Webserver des Springer-Verlages.

```
traceroute www.springer.de
```

`traceroute6` is a version of `traceroute` that is compatible with the Network Layer protocol IPv6.

Mit `traceroute6` existiert eine zum Vermittlungsprotokoll IPv6 kompatible Version von `traceroute`.

11.13

netstat

The command line tool `netstat` (*network statistics*) can be used to check which ports are open on the local system and which network connections exist to remote computers.

The following command shows which network services on the local system can be reached at which port numbers via the Transport Layer protocols TCP and UDP. This command also prints out for each socket the numeric user IDs of its owner.

netstat

Das Kommandozeilenwerkzeug `netstat` (*network statistics*) ermöglicht es zu überprüfen, welche Ports auf dem lokalen System geöffnet sind und welche Netzwerkverbindungen zu entfernten Rechnern bestehen.

Das folgende Kommando zeigt an, welche Netzwerkdienste auf dem lokalen System unter welchen Portnummern über die Transportprotokolle TCP oder UDP erreichbar sind. Zudem gibt das Kommando für jeden Socket die numerischen Benutzer-IDs des Eigentümers aus.

```
netstat -tulpen | grep -v 127.0.0.1
```

The `netstat` tool can also return the routing tables of the local network devices and statistical data on the IP and ICMP packets that are sent and received as well as for the connections with

Das Werkzeug `netstat` kann auch Routing-Tabellen der lokalen Netzwerkgeräte und statistische Daten zu den gesendeten und empfangenen IP- und ICMP-Paketen sowie zu den Ver-

local TCP and UTP sockets. Table 11.6 contains some examples for using `netstat`.

bindungen mit lokalen TCP- und UDP-Sockets ausgeben. Tabelle 11.6 enthält einige Aufrufbeispiele für `netstat`.

Table 11.6: Examples for using `netstat`

Command	Function
`netstat -rn`	Print out routing tables
`netstat -i`	Query statistical data for all network devices
`netstat -s`	Query statistical data for the protocols IP, ICMP, TCP and UDP
`netstat -ant`	Return states (see Figure 8.5) of TCP connections
`netstat -tlnp`	List all open (listening) ports

`netstat` only returns extensive information if the tool is executed with system administrator (*root*) privileges. Since `netstat` can also be used to query statistical data for the IP and ICMP protocols, this tool not only operates on the Transport Layer but also on the Network Layer.

Umfangreiche Informationen gibt `netstat` nur dann aus, wenn das Werkzeug mit den Rechten des Systemadministrators (*root*) aufgerufen wird. Da es mit `netstat` auch möglich ist, statistische Daten zu den Protokollen IP und ICMP abzufragen, arbeitet dieses Werkzeug nicht nur auf der Transportschicht, sondern auch auf der Vermittlungsschicht.

11.14

nmap

The command line tool `nmap` (*network mapper*) is a port scanner. It allows to inspect the port numbers on the local system or remote systems. The tool tries to find out which network services are accessible at which port numbers. This makes it possible to detect open ports and potentially unwanted network services that may be running in the background. This tool can also be used to find out the version numbers of network services and information about the operating system.

The following command tests all addresses in the class C subnet 10.0.0.0 for their accessibility via `ping` and thus quickly checks which systems are accessible in the local subnet.

nmap

Das Kommandozeilenwerkzeug `nmap` (*network mapper*) ist ein Portscanner. Mit diesem ist es möglich, die Portnummern auf dem lokalen System oder auf entfernten Systemen zu untersuchen. Das Werkzeug versucht herausfinden, welche Netzwerkdienste unter welchen Portnummern erreichbar sind. Damit ist es unter anderem möglich, offene Ports und eventuell unerwünschte, im Hintergrund laufende Netzwerkdienste zu entdecken. Zudem kann das Werkzeug die Versionsnummern der Netzwerkdienste und Informationen über das Betriebssystem herausfinden.

Das folgende Kommando testet alle Adressen im Klasse-C-Subnetz 10.0.0.0 auf Erreichbarkeit über `ping` und überprüft so in kurzer Zeit, welche Systeme im lokalen Subnetz erreichbar sind.

```
nmap -sP 10.0.0.0/24
```

Detailed information about the available network services, their version numbers, and port numbers, as well as information about the run-

Detaillierte Informationen zu den erreichbaren Netzwerkdiensten, deren Versionsnummern und Portnummern sowie Informationen zum

ning operating system can be fetched with this command:

laufenden Betriebssystem liefert das folgende Kommando:

```
nmap -A <address>
```

An IP address or a hostname can be specified as the address for the destination to be inspected.

Als Adresse für das zu überprüfende Ziel kann eine IP-Adresse oder ein Hostname angegeben sein.

11.15

dhclient

The DHCP client `dhclient` allows to assign logical addresses of a DHCP server on the Network Layer by using the Application Layer protocol DHCP (see section 9.2).

The following command tries to fetch a network configuration for the network interface `eth0` from a DHCP server.

dhclient

Der DHCP-Client `dhclient` ermöglicht die automatisierte Zuweisung logischer Adressen auf der Vermittlungsschicht von einem DHCP-Server mit Hilfe des Anwendungsprotokolls DHCP (siehe Abschnitt 9.2).

Das folgende Kommando versucht, für die Netzwerkschnittstelle `eth0` von einem DHCP-Server eine Netzwerkkonfiguration zu erhalten.

```
dhclient eth0
```

The command line parameter `-4` can be used to request IPv4 addresses explicitly. As an alternative, by using the command line parameter `-6`, `dhclient` requests IPv6 addresses.

Mit dem Kommandozeilenparameter `-4` können explizit IPv4-Adressen angefordert werden. Alternativ fordert `dhclient` mit dem Kommandozeilenparameter `-6` IPv6-Adressen an.

11.16

dig

With the tool `dig`, it is possible to query the servers of the Domain Name System (DNS). DNS (see section 9.1) is a protocol for resolving domain names. The tool returns replies from DNS servers that are generated by them during a domain name resolution. In this way, `dig` assists in the troubleshooting of the DNS system.

The following command resolves the domain name `www.springer.de` and returns the response of the DNS server used.

dig

Mit dem Werkzeug `dig` ist es möglich, die Server des Domain Name System (DNS) abzufragen. Das DNS (siehe Abschnitt 9.1) ist ein Protokoll zur Namensauflösung. Das Werkzeug gibt die Antworten von DNS-Servern aus, die diese bei einer Namensauflösung zurücksenden. Auf diese Weise erleichtert `dig` die Fehlersuche im DNS-System.

Das folgende Kommando führt eine Namensauflösung für den Hostnamen `www.springer.de` durch und gibt die Antwort des verwendeten DNS-Servers aus.

```
dig www.springer.de
```

11.17

ftp

An FTP client for the command line interface is `ftp`. It allows to download files from FTP servers and to upload files to them by using the FTP Application Layer protocol (see section 9.7).

With the following command, the FTP client tries to establish a connection to an FTP server via port number 21 on the system with the hostname `ftp.kernel.org`.

ftp

Ein FTP-Client für die Kommandozeile ist `ftp`. Damit ist es möglich, mit Hilfe des Anwendungsprotokolls FTP (siehe Abschnitt 9.7) von FTP-Servern Dateien herunterzuladen und zu diesen hochzuladen.

Beim folgenden Kommando versucht der FTP-Client, eine Verbindung mit einem FTP-Server über Portnummer 21 am System mit dem Hostnamen `ftp.kernel.org` aufzubauen.

```
ftp ftp.kernel.org 21
```

11.18

nc

The command `nc`, which is an alias to the command `netcat`, allows for interaction with different network services and is considered to be the Swiss army knife for network administrators. This tool can be used to analyze network services and to directly transmit data between systems. The capabilities of this tool are so sophisticated that only a selection of examples is given here.

The following command requests a web page from the web server of publisher Springer and returns the server's reply in the command line.

nc

Das Kommando `nc` bzw. `netcat` ermöglicht die Interaktion mit verschiedenen Netzwerkdiensten und gilt als eine Art Schweizer Taschenmesser für Netzwerkadministratoren. Mit diesem Werkzeug lassen sich Netzwerkdienste analysieren und Daten direkt zwischen Systemen versenden. Die Möglichkeiten sind so vielfältig, dass an dieser Stelle nur eine kurze Auswahl an Beispielen erfolgt.

Das folgende Kommando fordert die Webseite auf dem Webserver des Springer-Verlages an und gibt die Antwort des Servers auf der Kommandozeile aus.

```
echo -e "GET / HTTP/1.1\nHost: springer.de\n" | nc springer.de 80
```

The following command checks the accessibility of some port numbers on the local system.

Das folgende Kommando überprüft die Erreichbarkeit einiger Portnummern auf dem lokalen System.

```
nc -zv localhost 21 22 23 25 53 80 110 443
```

In order to send a file (unencrypted) with `nc` via a network, it is sufficient to start a server with the command line parameter `-l` on the receiving system. It is also important to specify a free port number (in this example `8888`) and an output file.

Um eine beliebige Datei (unverschlüsselt) mit `nc` über ein Netzwerk zu versenden, genügt es auf dem empfangenden System mit dem Kommandozeilenparameter `-l` einen Server zu starten. Wichtig sind noch die Angabe einer freien Portnummer (in diesem Beispiel `8888`) und einer Zieldatei.

```
nc -l 8888 > /tmp/inputfile
```

The following command acts as a client and sends an input file to the specified system and port number.

Das folgende Kommando überträgt als Client eine Quelldatei zum angegebenen System und der angegebenen Portnummer.

```
nc localhost 8888 < /tmp/outputfile
```

11.19

netperf

One way to measure the data rate between two computers within a specified time is to use `netperf`. This command line tool requires the server `netserver` to measure the data rate. This is usually deployed and executed automatically when `netperf` is deployed. The following command checks whether `netserver` is currently running.

netperf

Eine Möglichkeit, um die Übertragungsrate zwischen zwei Computern innerhalb einer festgelegten Zeit zu messen, bietet `netperf`. Das Kommandozeilenwerkzeug benötigt zur Messung der Übertragungsrate den Server `netserver`. Dieser wird bei der Installation von `netperf` üblicherweise automatisch installiert und gestartet. Das folgende Kommando prüft, ob `netserver` aktuell läuft.

```
netstat -pan | grep netserver
```

The following command starts `netserver` manually and specifies the port number at which the server is accessible.

Das folgende Kommando startet `netserver` manuell und legt die Portnummer fest, unter der der Server erreichbar sein sein wird.

```
netserver -p [<portnumber>]
```

The data rate is measured by launching the client `netperf`. The syntax of `netperf` is:

Die Messung der Übertragungsrate geschieht durch einen Aufruf des Clients `netperf`. Die Syntax von `netperf` ist:

```
netperf -H <addresses> -p [<portnumber>] -l <seconds> -f M
```

The address can be an IP address or a hostname. The command line parameter `-l <seconds>` allows specifying the duration of the test in seconds. The command line parameter `-f M` improves the readability of the output. It forces the data rate to be displayed in MB/s.

Die Adresse kann eine IP-Adresse oder ein Hostname sein. Der Kommandozeilenparameter `-l <Sekunden>` ermöglicht die Angabe der Test-Länge in Sekunden. Der Kommandozeilenparameter `-f M` verbessert die Lesbarkeit des Ergebnisses, denn er legt fest, dass Übertragungsrate in MB/s ausgeben wird.

11.20

nslookup

Similar to `dig`, the command line tool `nslookup` allows to query servers of the Domain Name System (DNS) for a domain name and to obtain the associated IP address.

nslookup

Wie `dig` ermöglicht es das Kommandozeilenwerkzeug `nslookup`, bei Servern des Domain Name System (DNS) nach einem Domainna-

The following command resolves the domain name `www.springer.de` and returns the reply of the DNS server used.

men zu fragen und so die zugehörige IP-Adresse zu erhalten.

Das folgende Kommando führt eine Namensauflösung für den Hostnamen `www.springer.de` durch und gibt die Antwort des verwendeten DNS-Servers aus.

```
nslookup www.springer.de
```

11.21

ssh

The Application Layer Protocol SSH allows the remote control of computers and is a secure alternative to Telnet. It enables an encrypted and thus secure connection between different computers via an insecure network. The syntax of the SSH client `ssh` is:

ssh

Das Anwendungsprotokoll SSH ermöglicht die Fernsteuerung von Rechnern und ist eine sichere Alternative zu Telnet. Es ermöglicht eine verschlüsselte und damit sichere Verbindung zwischen unterschiedlichen Rechnern über ein unsicheres Netzwerk. Die Syntax des SSH-Clients `ssh` ist:

```
ssh [-l <username>] [-p <portnumber>] <addresses>
```

The address can be an IP address or a hostname. If no port number is specified, the client attempts to connect to a network service on port 22. If no user name is specified when `ssh` is executed, the client attempts to login to the SSH server with the current user name.

Die Adresse kann eine IP-Adresse oder ein Hostname sein. Ist keine Portnummer angegeben, versucht der Client, eine Verbindung zu einem Netzwerkdienst an Port 22 herzustellen. Ist beim Aufruf von `ssh` kein Benutzername festgelegt, versucht der Client, sich mit dem aktuellen Benutzernamen am SSH-Server anzumelden.

11.22

telnet

With `telnet`, it is possible to remote control computers via the Application Layer protocol Telnet (see section 9.3). The syntax of `telnet` is:

telnet

Mit `telnet` ist es möglich, über das Anwendungsprotokoll Telnet (siehe Abschnitt 9.3) Rechner fernzusteuern. Die Syntax von `telnet` ist:

```
telnet <address> [<portnumber>]
```

The address can be an IP address or a hostname. If no port number is specified, the client attempts to connect to a network service on port 23.

This tool is useful for troubleshooting local or remote network services. Examples include web servers, FTP servers or SMTP servers. An-

Die Adresse kann eine IP-Adresse oder ein Hostname sein. Ist keine Portnummer angegeben, versucht der Client, eine Verbindung zu einem Netzwerkdienst an Port 23 herzustellen.

Zur Fehlersuche bei lokalen oder entfernten Netzwerkdiensten ist es ein hilfreiches Werkzeug. Beispiele sind Web-Server, FTP-Server

other field of application is the administration of databases.

Communication via Telnet is done in plain text. This means that even passwords are transmitted unencrypted between client and server. Telnet is therefore only appropriate for the remote control of computers or analysis of network services in secure environments. A secure alternative for the remote control of computers is `ssh`. An alternative solution for troubleshooting local or remote network services is the `nc` tool.

oder SMTP-Server. Ein weiteres Einsatzgebiet ist die Administration von Datenbanken.

Die Kommunikation via Telnet erfolgt im Klartext. Das bedeutet, dass auch Passwörter unverschlüsselt zwischen Client und Server übertragen werden. Telnet eignet sich darum nur in sicheren Umgebungen zur Fernsteuerung von Rechnern oder Analyse von Netzwerkdiensten. Eine sichere Alternative zur Fernsteuerung von Rechnern ist `ssh`. Eine alternative Lösung zur Fehlersuche bei lokalen oder entfernten Netzwerkdiensten ist das Werkzeug `nc`.

Technical Terms Fachbegriffe

Access control	Zugangskontrolle
Access Point	Basisstation
Acknowledgment	Bestätigung
Address class	Netzklasse
Address space	Adressraum, Adressbereich
Address range	Adressbereich
Address resolution	Adressauflösung
Agglomeration area	Ballungsgebiet
Alternating current voltage	Wechselspannung
Alternating magnetic fields	Magnetische Wechselfelder
Amplitude	Schwingungsweite
Advertised receive window	Empfangsfenster
Anisotropy	Richtungsabhängigkeit
Application layer	Anwendungsschicht
Attenuation	Dämpfung
Authenticity	Authentizität, Echtheit
Bandwidth	Bandbreite
Baseband	Basisband
Baseline Wander	Verschiebung des Durchschnitts
Beacon packets	Leuchtfeuer-Datenpakete
Bit sequence	Bitfolge
Bit stream	Datenstrom, Bitstrom
Braiding	Drahtgeflecht
Broadband	Breitband
Cable shield	Gesamtschirmung (eines Kabels)

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0

Cabling effort	Verkabelungsaufwand
Carrier frequency	Trägerfrequenz
Carrier sense	Trägerprüfung
Channel width	Kanalbreite
Checksum	Prüfsumme
Clock	Takt
Clock cycle	Taktzyklus
Clock pulse	Takt
Clock source	Taktquelle
Clock recovery	Taktrückgewinnung
Collision detection	Kollisionserkennung
Collision domain	Kollisionsdomäne
Common key	gemeinsamer Schlüssel
Common transmission medium	gemeinsames Übertragungsmedium
Compensating current	Ausgleichsstrom
Complementary signal	Komplementärsignal
Confidentiality	Vertraulichkeit
Congestion control	Überlastkontrolle
Connection type	Anschlussart
Congestion window	Überlastungsfenster
Connectionless	verbindungslos
Connection establishment	Verbindungsaufbau
Connection-oriented	verbindungsorientiert
Connection termination	Verbindungsabbau
Control line	Steuerleitung
Convergence time	Konvergenzzeit
Crosstalk	Nebensprechen, Übersprechen
Data encapsulation	Datenkapselung
Data exchange	Datenaustausch
Data line	Datenleitung
Data Link Layer	Sicherungsschicht
Data rate	(Daten-)Übertragungsrate
Data transmission rate	(Daten-)Übertragungsrate

Direct current voltage	Gleichspannung
Direct sequence spread spectrum	Frequenzspreizverfahren
directed	gerichtet
Directional dependence	Richtungsabhängigkeit
Dynamic frequency selection	Dynamisches Frequenzwahlverfahren
Edge	Kante (in einem Graphen)
Edge weight	Kantengewicht
Electromagnetic interferences	elektromagnetische Störungen
Electronic signature	Elektronische Unterschrift
Encapsulation	(Daten-)Kapselung
Enclosed rooms	geschlossener Raum
Even parity	gerade Parität
Fiber-optic cables	Lichtwellenleiter
Flow control	Flusskontrolle
Forwarding	Weitervermittlung, Weiterleitung
Frequency hopping method	Frequenzsprungverfahren
Frequency hopping spread spectrum	Frequenzspreizverfahren
Frequency range	Frequenzbereich
Frequency stages	Frequenzstufen
Functioning	Arbeitsweise
Fundamental frequency	Grundfrequenz
Gross transmission rate	Bruttoübertragungsrate
Ground potential	Masse, Erdpotential
Harmonic	Oberwelle
Host identifier	Hostadresse
Idle	Leerlauf, unbenutzt
Impedance	Wechselstromwiderstand
Interfering signal	Störung
Integer multiple	ganzzahliges Vielfaches
Jack	Buchse
Jack panel	Klinkenfeld
Jam signal	Störsignal
Key distribution	Schlüsselverteilung

Latency	Latenz
Leaf node	Blattknoten
Light impulse	Lichtimpuls
Line capacitance	Leitungskapazität
Line code	Leitungscode
Line encoding scheme	Leitungscode
Link state	Verbindungsstatus
Link state information	Verbindungs(status)information
Machine time	Rechenzeit
Mainframe	Großrechner
Moisture	Feuchtigkeit
Media access	Medienzugriff
Media access control method	Medienzugriffsverfahren
Media access unit	Ringleitungsverteiler
Microwave range	Mikrowellenbereich
Minimum spacing	Mindestabstand
Modulation method	Modulationsverfahren
Morse characters	Morsezeichen
Multi-carrier method	Mehrträgerverfahren
Multipath propagation	Mehrwegeausbreitung
Multiple access	Mehrfachzugang, Mehrfachzugriff
Narrowband	Schmalband
Net transmission rate	Nettoübertragungsrate
Netmask	Netzmaske, Subnetzmaske
Network allocation vector	Netzbelegungsvektor
Network capacity	Netzkapazität
Network failure	Netzwerkausfall
Network fault	Netzstörung
Network identifier	Netzadresse
Network layer	Vermittlungsschicht
Network outage	Netzwerkausfall
Network segments	Netzabschnitt
Network size	Netzwerkausdehnung

Noise	Rauschen
Odd parity	ungerade Parität
Offset	Abweichung
Oscillations	Schwingungen
Package	Paket (auf der Vermittlungsschicht)
Pair shielding	Paarschirm
Patch cable	Patch-Kabel, 1:1-Kabel
Payload (data)	Nutzdaten
Physical Layer	Bitübertragungsschicht
Plug	Stecker
Power of two	Zweierpotenz
Propagation delay	Ausbreitungsverzögerung
Propagation path	Ausbreitungsweg
Propagation speed	Ausbreitungsgeschwindigkeit
Protocol converter	Protokollumsetzer
Pseudo random	pseudozufällig
Quartz glass	Quarzglas
Queue	Warteschlange
Random	Zufall
Radio technology	Funktechnik
Receive buffer	Empfangspuffer
Receiver capacity	Empfangskapazität
Receive window	Empfangsfenster
Receiver	Empfänger
Refraction index	Brechungsindex
Reliability	Ausfallsicherheit
Remainder	Rest (einer Division)
Resolution	Auflösung
Root	Wurzel
Round-trip time	Umlaufzeit
Round-trip delay time	Umlaufzeitverzögerung
Routing table	Weiterleitungstabelle
Segment	Segment (auf der Transportschicht)

Session key	Sitzungsschlüssel
Session layer	Sitzungsschicht
Signal deformations	Signalverformung
Shared key	gemeinsamer Schlüssel
Shared transmission medium	gemeinsames Übertragungsmedium
Short-time disturbances	Kurzzeitstörungen
Sliding-window	Schiebefenster
Square-wave signal	Rechtecksignal
Starting node	Startknoten
Sub-carrier	Zwischenträger
Subnet mask	Netzmaske, Subnetzmaske
Success message	Erfolgsmeldung
Symbol rate	Symbolrate
Telegraph	Telegraf
Terminal device	Endgerät
Terrestrial	terrestrisch
Time division multiple access	Zeitmultiplexverfahren
Time scale	Zeitskala
Time sharing	Dialogbetrieb
Time slice	Zeitscheibe
Timeout	Zeitschranke
Throughput	(Daten-)Durchsatz
Transfer time	Übertragungsdauer
Transmission delay	Übertragungsverzögerung
Transmission duration	Übertragungsdauer, Sendedauer
Transmission-line capacitance	Leitungskapazität
Transmission media	Übertragungsmedien
Transmission medium	Übertragungsmedium
Transmission path	Übertragungsstrecke
Transmission power	Übertragungsleistung, Sendeleistung
Transmission power control	Sendeleistungsregelung
Transmission rate	Übertragungsrate
Transmission retry	Übertragungswiederholung

Transmission speed	Übertragungsgeschwindigkeit
Transmission time	Übertragungszeit, Übertragungsdauer
Transmission timing	Übertragungstakt
Transmit buffer	Sendepuffer
Transmit window	Sendefenster
Transport layer	Transportschicht
Threshold value	Schwellwert
Undirected	ungerichtet
Vampire tap	Vampirklemme
Vendor	Hersteller
Voltage curve	Spannungsverlauf
Velocity factor	Ausbreitungsfaktor, Verkürzungsfaktor
Weighted graph	(kanten)gewichteter Graph
Wired	kabelgebunden
Wireless	drahtlos
Wiring	Pinbelegung

Glossary

Anycast An Anycast message is sent to a single node from a group of nodes in a network

ARP Address Resolution Protocol. Protocol for address resolution. Translates IP addresses to MAC addresses

Arpanet Predecessor of the Internet with decentralized structure and packet switching

ASCII American Standard Code for Information Interchange. 7-bit character encoding

ATM Asynchronous Transfer Mode. Obsolete networking technology for WANs

Baud Speed at which one symbol is transmitted per second

Binary system Place value system that uses the number 2 as base

Bit Smallest possible unit of information. Two possible states

Bluetooth Wireless communication system for data transmission over short distances

BPDU Bridge Protocol Data Unit. Message from the STP

Glossar

Anycast Eine Anycast-Nachricht wird an einen einzelnen Teilnehmer aus einer Gruppe von Teilnehmern in einem Netzwerk gesendet

ARP Address Resolution Protocol. Protokoll zur Adressauflösung. Übersetzt IP-Adressen in MAC-Adressen

Arpanet Vorläufer des heutigen Internets mit dezentraler Struktur und Paketvermittlung

ASCII American Standard Code for Information Interchange. 7-Bit-Zeichenkodierung

ATM Asynchronous Transfer Mode. Veraltete Vernetzungstechnologie für WANs

Baud Geschwindigkeit, mit der ein Symbol pro Sekunde übertragen wird

siehe Dezimalsystem.

Bit Kleinstmögliche Informationseinheit. Zwei mögliche Zustände

Bluetooth Funksystem zur Datenübertragung auf kurzen Distanzen

BPDU Bridge Protocol Data Unit. Nachricht des STP

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0

Bridge Data Link Layer device with two ports for connecting physical networks

Broadcast A Broadcast message is sent to all nodes in a network

Byte Group of 8 bits

Checksum Instrument to ensure data integrity

Decimal system Place value system that uses the number 10 as base

DNS Domain Name System. Protocol for domain name resolution

see Binary system

Ethernet Cable-based networking technology for LANs

CSMA/CA Media Access Control Method of WLAN

CSMA/CD Media Access Control Method of Ethernet

FDDI Fiber Distributed Data Interface. Cable-based networking technology which has been replaced mostly by ATM and Ethernet. It uses optical fibers as the transmission medium and thus offers a maximum length of 100 km. The media access control method uses a token frame similar to Token Ring

Forwarding Forward of IP packets by a Router according to the information in its local forwarding table

Bridge Netzwerkgerät der Sicherungsschicht mit zwei Schnittstellen, das physische Netze verbindet

Broadcast Eine Broadcast-Nachricht wird an alle Teilnehmer in einem Netzwerk gesendet

Byte Gruppe von 8 Bits

CRC Cyclic Redundancy Check. Verfahren zur Bestimmung einer Prüfsumme

Dezimalsystem Stellenwertsystem mit der Basis 10

DNS Domain Name System. Protokoll zur Namensauflösung

Dualsystem Stellenwertsystem mit der Basis 2

Ethernet Kabelgebundene Vernetzungstechnologie für LANs

CSMA/CA Medienzugriffsverfahren von WLAN

CSMA/CD Medienzugriffsverfahren von Ethernet

FDDI Fiber Distributed Data Interface. Von ATM und Ethernet weitgehend verdrängte kabelgebundene Vernetzungstechnologie, die Lichtwellenleiter als Übertragungsmedium verwendet und dadurch eine maximale Ausdehnung von 100 km ermöglicht. Das Medienzugriffsverfahren basiert ähnlich wie bei Token Ring auf einem Token-Rahmen

Forwarding Weiterleiten von IP-Paketen durch einen Router anhand der Informationen in dessen lokaler Weiterleitungstabelle

Frame Message on the Data Link Layer

siehe Rahmen

FTP File Transfer Protocol. Application Layer protocol for exchanging files

FTP File Transfer Protocol. Anwendungsprotokoll zum Austausch von Dateien

GAN Global Area Network. Connects WANs

GAN Global Area Network. Verbindet WANs

Gateway Protocol translator. Enables communication between networks based on different protocols. Examples are VPN Gateways

Gateway Protokollumsetzer. Ermöglicht Kommunikation zwischen Netzen, die auf unterschiedlichen Protokollen basieren. Beispiele sind VPN-Gateways

see Optical Fiber

Glasfaser siehe Lichtwellenleiter

Header Supplementary data before the payload data

Header Zusatzinformationen vor den Nutzdaten

Hexadecimal system Place value system that uses the number 16 as base

Hexadezimalsystem Stellenwertsystem mit der Basis 16

HTTP Hypertext Transfer Protocol. Application Layer protocol for data transmission

HTTP Hypertext Transfer Protocol. Anwendungsprotokoll zur Datenübertragung

Hub Repeater with more than two ports

Hub Repeater mit mehr als zwei Schnittstellen

ICMP Internet Control Message Protocol. Protocol for the exchange of diagnostic and error messages via the IP

ICMP Internet Control Message Protocol. Protokoll zum Austausch von Informations- und Fehlermeldungen über das IP

IEEE Professional association of electrical and computer engineers that defines standards, among others, for computer networks

IEEE Institute of Electrical and Electronics Engineers. Berufsverband von Ingenieuren der Elektrotechnik und Informatik, der Standards u. a. für Computernetze definiert

IP Internet Protocol. Protocol of the Network Layer

IP Internet Protocol. Protokoll der Vermittlungsschicht

IP address Logical address on the Network Layer

IP-Adresse Logische Adresse auf der Vermittlungsschicht

ISO International Organization for Standardization

ISO Internationale Organisation für Normung

ISP Internet Service Provider. Provides services for accessing the Internet

ISP Internet Service Provider. Anbieter bzw. Dienstleister, der eine Anbindung zum Internet bereitstellt

Coaxial cable Bipolar cable with a concentric (coaxial) structure for the transmission of electrical signals

Koaxialkabel Zweipolige Kabel mit konzentrischem (koaxialem) Aufbau zur Übertragung elektrischer Impulse

LAN Local Area Network. Local computer network

LAN Local Area Network. Lokales Computernetz

Line code Specifies how signals are transmitted on the transmission medium

Leitungscode Definiert die Art und Weise, wie Signale auf dem Übertragungsmedium übertragen werden

see Optical fiber

Lichtwellenleiter Kabel mit Quarzglas- oder Kunststofffasern zur Übertragung von Licht

MAC address Physical address on the Data Link Layer

MAC-Adresse Physische Adresse auf der Sicherungsschicht

MAN Metropolitan Area Network. Connects LANs

MAN Metropolitan Area Network. Verbindet LANs

Modem Network device of the Physical Layer and Data Link Layer that modulates digital signals onto a carrier frequency in the high-frequency range and demodulates them

Modem Netzwerkgerät der Bitübertragungsschicht und Sicherungsschicht, das digitale Signale auf eine Trägerfrequenz im Hochfrequenzbereich aufmoduliert und demoduliert

MSS Maximum Segment Size. The maximum amount of payload in bytes in a TCP segment

MSS Maximum Segment Size. Maximale Anzahl Nutzdaten in Bytes in einem TCP-Segment

MTU Maximum Transmission Unit. Maximum packet size of the Network Layer that can be transmitted without fragmentation in a frame of the Data Link Layer

MTU Maximum Transmission Unit. Maximale Paketgröße der Vermittlungsschicht, die ohne Fragmentierung in einem Rahmen der Sicherungsschicht übertragen werden kann

Multicast A Multicast message is sent to a group of nodes in a network

Multicast Eine Multicast-Nachricht wird an eine Gruppe von Teilnehmern in einem Netzwerk gesendet

NAT Network Address Translation. A method for automatically modifying the addresses of the network layer and the transport layer by a Router or a Gateway to connect a (private) network with the Internet through a public address

NAT Network Address Translation. Verfahren zur automatisierten Anpassung der Adressen von Vermittlungsschicht und Transportschicht durch einen Router oder Gateway, um ein (privates) Netz über eine öffentliche Adresse mit dem öffentlichen Netz zu verbinden

Nibble Group of 4 bits, also called half-byte

Nibble Gruppe von 4 Bits bzw. ein Halbbyte

NTP Network Time Protocol. Application Layer protocol for synchronizing clocks between computer systems

NTP Network Time Protocol. Anwendungsprotokoll zur Synchronisierung von Uhren zwischen Computersystemen

Octal system Place value system that uses the number 8 as base

Oktalsystem Stellenwertsystem mit der Basis 8

Octet see byte

Oktett siehe Byte

Optical fiber Cable with fused silica or plastic optical fibers for the transmission of light

siehe Lichtwellenleiter

OSI Open Systems Interconnection. Model of protocol layers of the ISO

OSI Open Systems Interconnection. Schichtenmodell der ISO

OSPF Open Shortest Path First. Link-State Routing Protocol based on the Dijkstra algorithm

OSPF Open Shortest Path First. Link-State-Routing-Protokoll auf Basis des Dijkstra-Algorithmus

Packet Message on the Network Layer

Paket Nachrichteneinheit auf der Vermittlungsschicht

PAN Personal Area Network. Private network with a short range

PAN Personal Area Network. Privates Netz mit geringer Reichweite

Piconet Network of Bluetooth devices. A master controls the access to the transmission media

Piconetz Netz aus Bluetooth-Geräten. Ein Master regelt den Medienzugriff

PoE Power over Ethernet. Method for powering network devices over a twisted pair cable

PoE Power over Ethernet. Verfahren zur Stromversorgung von Netzwerkgeräten über das Twisted-Pair-Kabel

POP3 Post Office Protocol. Application Layer protocol for listing, fetching, and deleting of emails

POP3 Post Office Protocol. Anwendungsprotokoll zum Auflisten, Abholen und Löschen von Emails

PowerLAN Network technology that uses the power grid of a building as a shared transmission medium

PowerLAN Netzwerktechnologie, die das Stromnetz eines Gebäudes als gemeinsames Übertragungsmedium verwendet

Powerline see PowerLAN

Powerline siehe PowerLAN

PPP Point-to-Point Protocol. Protocol of the data link layer to establish, maintain and terminate point-to-point connections. Typically used in circuit-switching networks such as telephone modems and ISDN devices

PPP Point-to-Point Protocol. Protokoll der Sicherungsschicht um Punkt-zu-Punkt-Verbindungen aufzubauen, aufrecht zu erhalten und zu beenden. Wird typischerweise in leitungsvermittelnden Netzen u.a. von Telefonmodems und ISDN-Anschlussgeräten verwendet

Protocol Agreement of communication rules

Protokoll Vereinbarung von Kommunikationsregeln

see Checksum

Prüfsumme Mittel zur Gewährleistung von Datenintegrität

see Frame

Rahmen Nachrichteneinheit auf der Sicherungsschicht

Repeater Network device of the Physical Layer with two ports that extends the range of LANs by forwarding the signals without analyzing them

Repeater Netzwerkgerät der Bitübertragungsschicht mit zwei Schnittstellen, das LANs vergrößert, indem es die Signale weiterleitet, ohne sie zu analysieren

RIP Routing Information Protocol. Distance vector routing protocol based on the Bellman-Ford algorithm

RIP Routing Information Protocol. Distanzvektor-Routing-Protokoll auf Basis des Bellman-Ford-Algorithmus

RJ45 Physical connection standard for twisted pair cables

RJ45 Anschlusssystem für Twisted-Pair-Kabel

Router Network device of the Network Layer that forwards packets between logical networks. Also, a Router allows to connect a LAN with a WAN

Router Netzwerkgerät der Vermittlungsschicht, das Pakete zwischen logischen Netzen vermittelt. Zudem ermöglicht ein Router die Verbindung eines LAN mit einem WAN

Routing Process of creating the forwarding table in a Router using a protocol like RIP or OSPF

RTT Round Trip Time. The time it takes for a data transmission to be transmitted from the source to the destination and back

Scatternet Network of multiple piconets

Segment Message on the Transport Layer

SMTP Simple Mail Transfer Protocol. Application Layer protocol for sending emails

Socket Platform-independent, standardized interface between the implementation of the network protocols in the operating system and the applications. A socket consists of a port number and an IP address

Spanning tree Cycle-free graph

SSH Secure Shell. Application Layer protocol for the remote control of computers

STP Spanning Tree Protocol. Protocol of the data link layer with which Bridges agree on a spanning tree

(Layer-2-)Switch Bridge with more than two ports

(Layer-3-)Switch Router for use only inside a LAN. Has no port to connect to a WAN

Telnet Application Layer protocol for the remote control of computers

Routing Erstellung der Weiterleitungstabelle in einem Router mit Hilfe eines Protokolls wie z. B. RIP oder OSPF

RTT Round Trip Time (Rundlaufzeit). Die Zeit, die eine Datenübertragung benötigt, um von der Quelle zum Ziel und zurück übertragen zu werden

Scatternetz Zusammenschluss mehrere Piconetze

Segment Nachrichteneinheit auf der Transportschicht

SMTP Simple Mail Transfer Protocol. Anwendungsprotokoll zum Versand von Emails

Socket Plattformunabhängige, standardisierte Schnittstelle zwischen der Implementierung der Netzwerkprotokolle im Betriebssystem und den Anwendungen. Ein Socket besteht aus einer Portnummer und einer IP-Adresse

Spannbaum Zyklenfreier Graph

SSH Secure Shell. Anwendungsprotokoll zur Fernsteuerung von Rechnern

STP Spanning Tree Protocol. Protokoll der Sicherungsschicht, mit dem Bridges sich auf einen auf einen Spannbaum einigen

(Layer-2-)Switch Bridge mit mehr als zwei Schnittstellen

(Layer-3-)Switch Router für den ausschließlichen Betrieb innerhalb eines LAN. Bietet keine Schnittstelle zum Anschluss an ein WAN

Telnet Anwendungsprotokoll zur Fernsteuerung von Rechnern

TCP Transmission Control Protocol. Connection-oriented Transport Layer Protocol

TCP Transmission Control Protocol. Verbindungsorientiertes Transportprotokoll

Token Ring Obsolete cable-based networking technology for LANs

Token Ring Veraltete kabelgebundene Vernetzungstechnologie für LANs

Topology Logical or physical connection scheme of a computer network

Topologie Logische oder physische Verbindungsstruktur eines Computernetzes

Trailer Additional data following the payload data

Trailer Zusatzinformationen nach den Nutzdaten

Twisted pair Cable for the transmission of electrical impulses, the cores of which are twisted together in pairs

Twisted-Pair Kabel zur Übertragung elektrischer Impulse, deren Adern paarweise miteinander verdrillt sind

UDP User Datagram Protocol. Connectionless Transport Layer protocol

UDP User Datagram Protocol. Verbindungsloses Transportprotokoll

Unicast A Unicast message is sent to a single node in a network

Unicast Eine Unicast-Nachricht wird an einen einzelnen Teilnehmer in einem Netzwerk gesendet

Unicode Multi-byte character encoding

Unicode Mehrbyte-Zeichenkodierung

VLAN Virtual Local Area Network. Logical subnet on the Network Layer

VLAN Virtual Local Area Network. Logisches Teilnetz auf der Vermittlungsschicht

VPN Virtual Private Network. Logical subnet within public networks. Can be implemented on the Data Link Layer, on the Network Layer, and the Transport Layer

VPN Virtual Private Network. Logische Teilnetz innerhalb öffentlicher Netze. Anwendungsmöglichkeit auf der Sicherungsschicht, Vermittlungsschicht und Transportschicht

WAN Wide Area Network. Connects MANs

WAN Wide Area Network. Verbindet MANs

WEP Wired Equivalent Privacy. Obsolete security standard for WLAN

WEP Wired Equivalent Privacy. Veralteter Sicherheitsstandard für WLAN

WiMAX Wireless Metropolitan Area Networks. Wireless network standard for MANs

WiMAX Wireless Metropolitan Area Networks. Funknetzstandard für MANs

WLAN Wireless LAN. Wireless network standard for LANs. In this book means WLAN always wireless networks that meet the IEEE 802.11 (WiFi) standards

WPA(2) Wi-Fi Protected Access. Security standard for WLAN

WLAN Wireless LAN. Funknetzstandard für LANs. In diesem Buch steht WLAN immer für drahtlose Netzwerke, die den IEEE 802.11 (WiFi) Standards entsprechen

WPA(2) Wi-Fi Protected Access. Sicherheitsstandard für WLAN

References

[1] Kappes M (2013) Netzwerk- und Datensicherheit. 2. Auflage, Springer Vieweg, Wiesbaden

[2] Ballmann B (2012) Network Hacks. Springer Vieweg, Heidelberg

[3] Beijnum, van I. (2007) IPv4 Address Space. The Internet Protocol Journal. Volume 10. Issue 3

[4] DEC (1984) DDCMP. Functional Specification, Phase IV, Version 4.1

[5] Grumm H, Sommer M (2011) Einführung in die Informatik. Oldenburg, München

[6] Hompel M, Büchter H, Franzke U (2008) Identifikationssysteme und Automatisierung. Springer, Berlin Heidelberg

[7] Chen K, Xue Y, Samarth S, Nahrstedt K (2003) Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks. Elsevier. `http://cairo.cs.uiuc.edu/publications/papers/elsevier2004-bdp.pdf`

[8] IEEE – Registration Authority OUI Public Listing. `http://standards.ieee.org/develop/regauth/oui/public.html`

[9] Peterson L, Davie B (2008) Computernetze. dpunkt, Heidelberg

[10] Rahnema M (2008) UMTS Network Planning, Optimization, and Inter-Operation with GSM. Wiley

[11] Rech J (2008) Ethernet. Heise, Hannover

[12] Rech J (2012) Wireless LANs. Heise, Hannover

[13] Roth J (2010) Prüfungstrainer Rechnernetze. Vieweg, Wiesbaden

[14] Sauter M (2018) Grundkurs Mobile Kommunikationssysteme. 7. Auflage, Springer Vieweg, Wiesbaden

[15] Schreiner R (2009) Computernetzwerke. Hanser, München

[16] Tanenbaum A S, Wetherall D J (2012) Computernetzwerke. 5. Auflage, Pearson, München

[17] Tanenbaum A S (2003) Computernetzwerke. 4. Auflage, Pearson, München

[18] Trick U, Weber F B (2015) SIP und Telekommunikationsnetze. 5. Auflage, De Gruyter Oldenbourg, Berlin

[19] Meinel C, Sack H (2009) Digitale Kommunikation: Vernetzen, Multimedia, Sicherheit. Springer, Berlin Heidelberg

[20] Lienemann G, Larisch D (2011) TCP/IP Grundlagen und Praxis. Heise, Hannover

[21] Kerner H (1992) Rechnernetze nach OSI. Addison-Wesley, Bonn

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0

[22] Koutny M, Mlynek P, Misurec J (2013) Analysis of CSMA/CA used in Power Line Communication. IEEE

[23] Kurose J, Ross K (2008) Computernetzwerke. Vieweg+Teubner, Wiesbaden

[24] Medhi D (2007) Network Routing: Algorithms, Protocols, and Architectures. Morgan Kaufmann

[25] Root Server Technical Operations `http://www.root-servers.org`

[26] Scherff J (2010) Grundkurs Computernetzwerke. 2. Auflage, Pearson, München

[27] Wikipedia. `http://de.wikipedia.org/w/index.php?title=Fehlerkorrekturverfahren`

[28] Wikipedia. `http://de.wikipedia.org/wiki/CSMA/CD`

[29] Velocity factor. `https://en.wikipedia.org/wiki/Velocity_factor`

[30] Nurmi D, Wolski R, Grzegorczyk C, Obertelli G, Soman S, Youseff L, Zagorodnov D (2009) The Eucalyptus Open-source Cloud-computing System. IEEE

[31] Riggert W (2014) Rechnernetze. 5. Auflage, Hanser, München

[32] Zisler H (2016) Computer-Netzwerke. 4. Auflage, Rheinwerk Verlag, Bonn

[33] Mandl P, Weiß J, Bakomenko A (2010) Grundkurs Datenkommunikation. 2. Auflage, Vieweg+Teubner, Wiesbaden

Index

1000BASE-T, 66–68
100BASE-FX, 86
100BASE-T4, 66, 89
100BASE-TX, 66, 67, 71, 86
100Base-VG, 87
10BASE-T, 66, 67, 71, 83
10BASE2, 24, 32, 33, 63, 64, 75, 83
10BASE5, 24, 46, 63, 75, 121, 122
10BROAD36, 47
10GBASE-T, 67
1:1 Verdrahtung, 67
1:1 Wiring, 67
2.5GBASE-T, 67
40GBASE-T, 67
4B5B, 86
5-4-3 Repeater Rule, 75
5-4-3-Repeater-Regel, 75
5B6B, 86
5GBASE-T, 67
8B10B, 88
8B6T, 89
8P8C, 65, 66

Abschirmung, 68
Access Control, 33
Access Point, 22, 49, 92
Active FTP, 209
Ad-hoc Mode, 49, 111, 112
Ad-hoc-Modus, 49, 111, 112
Address
    logical, 39, 133
    physical, 38, 101
Address Resolution, 129, 163
Addressing, 101, 133, 172
Adressauflösung, 129, 163
Adresse
    logische, 39, 133
    physisch, 38, 101
Adressierung, 101, 133, 172
Advertised Receive Window, 185
Advertisement, 150
Aktives FTP, 209
Alternate Mark Inversion, 82
Alternating Current, 28
Anwendungsschicht, 36, 40, 195
Anycast, 134
Application Layer, 36, 40, 195
Area, 157
ARP, 129, 163
arp (Command), 224
arp (Kommando), 224
ARP Cache, 129, 224
Arpanet, 15, 36
ASCII, 9, 43, 105, 114, 115, 204
Asynchrone Übertragung, 20
Asynchronous Transmission, 20
ATM, 18
Attachment Unit Interface, 63
Attenuation, 28, 73
AUI, 63
Ausbreitungsfaktor, 30
Ausdehnung, 18
Auto-MDIX, 68
Autonomes System, 149
Autonomous System, 149

B8ZS, 82
Backoff Time, 119, 125
Backoffzeit, 119, 125
Bandbreite, 29
Bandbreite-Verzögerung-Produkt, 30
Bandwidth, 29
Bandwidth-Delay Product, 30
Baseband, 47
Baseline Wander, 77
Basisband, 47
Basisstation, 22, 49, 92
Baud Rate, 29
Baudrate, 29
Baum-Topologie, 27
Beacon Frame, 50, 109
Beacon-Rahmen, 50, 109
Bellman-Ford Algorithm, 149
Bellman-Ford-Algorithmus, 149
BGP, 149
Binary System, 5
Biphase-L, 84
Bipolar Encoding, 82

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019
C. Baun, *Computer Networks / Computernetze*, https://doi.org/10.1007/978-3-658-26356-0

BISYNC, 105, 115, 116
Bit, 3
Bit Rate, 29
Bit Sequence, 4
Bit Stuffing, 106
Bitübertragungss., 37, 45, 106
Bitfolge, 4
Bitrate, 29
Bitstopfen, 106
Bluetooth, 18, 27, 59, 118
    Scatternet, 60
Border Gateway Protocol, 149
BPDU, 98
Breitband, 47
Bridge, 22, 38, 39, 91, 93
    Designated, 98
    Designierte, 98
Bridge Protocol Data Unit, 98
Bridge-ID, 97
Broadband, 47
Broadcast, 134
Broadcast Domain, 132
Broadcast-Domäne, 132
Browser, 40
BSSID, 112
Bus Network, 24
Bus-Topologie, 24
Byte, 7
Byte Stuffing, 105

Cable
    Crossover, 67
    Optical Fiber, 71
    Patch, 67
    Twisted Pair, 64
CAN Bus, 77
CAN-Bus, 77
Cellular Network, 27
Character Encoding, 9
Character Stuffing, 105
Cheapernet, 63, 64
CIDR, 138
Clock Recovery, 79
Clock Source, 20
Coaxial Cable, 17, 62
Collision, 119
Collision Domain, 33, 75, 101, 132
Communication, 41
    connection-oriented, 40
    connectionless, 40
Complementary Signal, 64
Computer Networking, 15
Computervernetzung, 15
Conditional DePhase Encoding, 84
Congestion Avoidance, 190
Congestion Control, 187
Congestion Window, 188
Connection-oriented, 40
Connectionless, 40
Contention Window, 119, 125
Count-to-Infinity, 151
CRC, 115
Crossover, 67
CSMA/CA, 33, 38, 91, 122–124
    PCF, 128
    RTS/CTS, 125
CSMA/CD, 24, 33, 38, 75, 91, 108, 118
Cyclic Redundancy Check, 115

Dämpfung, 28, 73
Darstellungsschicht, 43
Data Link Layer, 38, 91
Data Signal, 28
Data Transmission, 19
    Asynchronous, 20
    Directional Dependence, 21
    Duplex, 21
    Half-Duplex, 21
    Parallel, 19
    Serial, 19
    Simplex, 21
    Synchronous, 20
Datagram Socket, 173
Dateigröße, 7
Datenübertragung, 19
    Asynchron, 20
    Duplex, 21
    Halbduplex, 21
    Parallel, 19
    Richtungsabhängigkeit, 21
    Seriell, 19
    Simplex, 21
    Synchron, 20
Datenkapselung, 35, 41
Datensignal, 28
DDCMP, 104
De-encapsulation, 41
Decimal System, 5
DECnet, 104
Default Gateway, 132, 163
Delay, 29
Desktop Firewall, 23
Dezimalsystem, 5
dhclient (Command), 231

dhclient (Kommando), 231
DHCP, 199, 231
    Ack, 200
    Client, 199
    Port Numbers, 199
    Portnummern, 199
    Relay, 199, 202
    Server, 199
Dialogstation, 16
Dienst, 35
Diff. Manchester Encoding, 84
Diff. Manchesterkodierung, 84
dig (Command), 231
dig (Kommando), 231
Dijkstra Algorithm, 149, 155
Dijkstra-Algorithmus, 149, 155
Dimension, 18
Direct Current, 28
Distance Vector Routing Protocols, 149
Distanzvektorprotokoll, 149
DNS, 41, 195, 231, 233
Drahtlose Übertragung, 62
Dreiwege-Handshake, 179, 180
DSSS, 52
Dualsystem, 5
Duplex, 21
Durchsatz, 29

Einweglatenz, 31
Electric Conductors, 62
Elektrische Leiter, 62
Empfangsfenster, 185
Empfangspuffer, 184
Encapsulation, 41
Encoding, 76
End-to-Site VPN, 213
Endgerät, 23
Endspan, 71
Entkapselung, 41
Error-correction, 117
Error-detection, 114
Ethernet, 18, 45, 107, 113, 116, 122, 142, 176
    Ausdehnung, 121
    DIX, 104
    Frame Length, 121
    Pinout, 66
    Rahmenlänge, 121
    Size, 121
    Steckerbelegung, 66
ethtool (Command), 222
ethtool (Kommando), 222
EUI, 144
Extended System ID, 98
Extended Unique Identifier, 144
External Firewall, 23
Externe Firewall, 23

Fading, 49, 123
Fast Ethernet, 26, 86, 87, 89
Fast Recovery, 191
Fast Retransmit, 190
FDDI, 26, 32, 86, 114
Fehlererkennung, 114
Fehlerkorrektur, 117
Fensterskalierung, 193
Fiber Distributed Data Interface, 25
Fiber-optic Cable, 17, 18, 62
Fibre Channel, 26, 88
File Size, 7
File Transfer Protocol, 43
Firewall, 22
FireWire, 18, 88
Flow Control, 183
Flusskontrolle, 183
Forwarding, 131, 148
Forwarding Table, 93
Fourier Series, 28
Fourierreihe, 28
FQDN, 196
Fragmentation, 142
Fragmentieren, 142
Frame, 38, 91, 104
Frequency, 28
Frequenz, 28
FTP, 16, 41, 43, 209, 232
ftp (Command), 232
ftp (Kommando), 232
Full-Duplex, 21
Fundamental Frequency, 28

G.hn, 128
GAN, 18
Gateway, 22, 92, 132
Gebäudeverkabelung, 72
Gemeinsames Übertragungsmedium, 32
Generator Polynomial, 115
Generatorpolynom, 115
GG45, 70
Gigabit Ethernet, 75, 88
Gleichspannung, 28
Global Area Network, 18
Grundfrequenz, 28

Halbbyte, 6

Halbduplex, 21
Half-byte, 6
Half-Duplex, 21
Harmonic, 28
Harmonische, 28
HDLC, 106, 128
Hexadecimal System, 6
Hexadezimalsystem, 6
Hidden Terminal, 49, 123, 127
HomeGrid, 128
Hops, 150
Horizontal Communication, 41
Horizontale Kommunikation, 41
Host Identifier, 135
Hostadresse, 135
HTTP, 41, 204
Hub, 21, 27, 73
Hybrid Reference Model, 37, 44
Hybrides Referenzmodell, 37, 44

ICMP, 167, 228
ifconfig (Command), 225
ifconfig (Kommando), 225
iftop (Command), 227
iftop (Kommando), 227
InfiniBand, 26, 88
Information Representation, 9
Informationsdarstellung, 9
Infrastructure Mode, 49, 111, 112
Infrastruktur-Modus, 49, 111, 112
Inter-AS-Routing, 150
Inter-Switch Link, 216
Interface Identifier, 144
Interference, 48
Interferenz, 48
Interframe
    Gap, 108
    Spacing, 108
Internet, 149
Internet Layer, 36, 37, 39, 131
Internet Protocol, 39, 131
Internetschicht, 36, 37, 39, 131
Internetworking, 39, 131, 162
Intra-AS-Routing, 149
IP, 39, 131
ip (Command), 225
ip (Kommando), 225
IP Address, 39, 133, 135
IP-Adresse, 39, 133, 135
ipconfig (Command), 225
ipconfig (Kommando), 225
IPsec, 215
IPv4, 135
IPv6, 143
IPX, 39
IrDA, 18, 62
ISDN, 69, 82, 92, 116
iwconfig (Command), 223
iwconfig (Kommando), 223
iwlist (Command), 224
iwlist (Kommando), 224

Jitter, 73

Kabel
    Crossover, 67
    Lichtwellenleiter, 71
    Patch, 67
    Twisted-Pair, 64
Knoten, 23
Koaxialkabel, 17, 62
Kodierung, 76
Kollision, 119
Kollisionsdomäne, 33, 75, 101, 132
Kommunikation, 40, 41
    verbindungslos, 40
    verbindungsorientiert, 40
Komplementärsignal, 64
Kreise, 94

L2TP, 214
LAN, 18
Laser Bridge, 62, 92
Latency, 29
Latenz, 29
Layer-2-Switch, 22, 92
Layer-2-VPN, 214
Layer-3-Switch, 22, 132
Layer-3-VPN, 215
Layer-4-VPN, 215
Lease Time, 200
Leitungscode, 76, 106
Leuchtfeuer-Rahmen, 50
Lichtgeschwindigkeit, 30
Lichtwellenleiter, 17, 18, 62, 71
Line Code, 76, 106
Link Layer, 36
Link-State Routing Protocol, 149, 155
Link-State-Routing-Protokoll, 149, 155
Local Area Network, 18
Logical Topology, 24
Logische Topologie, 24
Long Preamble Format, 109
Loop, 94

LSB, 5, 108
LTE, 17

MAC Address, 38, 91, 101
MAC-Adresse, 38, 91, 101
MAN, 18
Manchester, 83, 106
Manchester II, 84
Maschen-Topologie, 26
Maximum Segment Size, 176
Maximum Transm. Unit, 113, 142, 164, 176
Media Access Control Method, 32, 33, 117
Media Access Unit, 26
Medienzugriffsverfahren, 33, 117
Mehrprotokoll-Router, 132
Mehrwegeausbreitung, 48
Mesh Network, 26
Metric, 150
Metrik, 150
Metropolitan Area Network, 18
Midspan, 71
mii-tool (Command), 223
mii-tool (Kommando), 223
MIMO, 53
MLT-3, 80
Modem, 22, 75, 92
Modulation Method, 52
Modulationsverfahren, 52
Mono-Mode Fiber, 72
Monomodefaser, 72
Morse Alphabet, 76
Morse Code, 76
Morsealphabet, 76
Morsekode, 76
MSB, 5
MSS, 176
MTU, 113, 142, 164, 176
Multi-Mode Fiber, 72
Multicast, 134
Multimodefaser, 72
Multipath Propagation, 48
Multiport Bridge, 38, 39, 92
Multiport Repeater, 21, 73
Multiprotocol Router, 132
Multiprotokoll-Router, 132

Name Server, 195
Nameserver, 195
NAT, 164
NAT Translation Table, 165
NAT-Übersetzungstabelle, 165
NAT66, 167
nc (Command), 232
nc (Kommando), 232
netperf (Command), 233
netperf (Kommando), 233
netstat (Command), 229
netstat (Kommando), 229
Network Access Layer, 36
Network Address Translation, 164
Network Analysis, 219
Network Configuration, 219
Network Device, 21
Network Identifier, 135
Network Layer, 39, 131
Network Prefix, 144
Network Protocol, 17
Network Service, 17
Network Time Protocol, 135
Network Topology, 24
Network Virtualization, 213
Networking Technologies, 45
Netzadresse, 135
Netzmaske, 137
Netzwerkanalyse, 219
Netzwerkdienst, 17
Netzwerkgerät, 21
Netzwerkkonfiguration, 219
Netzwerkprotokoll, 17
Netzwerktopologie, 24
Netzwerkvirtualisierung, 213
Netzzugangsschicht, 36
Nibble, 6
nmap (Command), 230
nmap (Kommando), 230
Node, 23
NRZ, 77
NRZI, 79
nslookup (Command), 233
nslookup (Kommando), 233
NTP, 135
Numbers, 4
NVT, 203

Octal System, 6
OFDM, 53
Oktalsystem, 6
One-way Latency, 31
Operation, 35
Optical Fiber, 71
OSI Reference Model, 42, 43
OSI-Referenzmodell, 42, 43
OSPF, 149, 155

Packet, 39, 131
Pager, 21
Pairing, 61
Paket, 39, 131
PAN, 18
Parallel Transmission, 19
Parallele Übertragung, 19
Parität, 114
Parity, 114
Passive FTP, 209
Passives FTP, 209
PATA, 19
Patch, 67
PD, 71
Period, 28
Periodendauer, 28
Personal Area Network, 18
Personal Firewall, 23
Physical Layer, 37, 106
Physical Topology, 24
Physiscal Layer, 45
Physische Topologie, 24
Piconet, 60, 118
ping (Command), 228
ping (Kommando), 228
PoE, 71
    Endspan, 71
    Injector, 71
    Midspan, 71
    PD, 71
    PSE, 71
Point-to-Point Protocol, 92
Polling, 128
POP3, 41, 208
Port Number, 39, 172
Port Scanner, 230
Portnummer, 39, 172
Power over Ethernet, 71
PowerLAN, 24, 47, 125
Powerline Communication, 24, 47, 125
PPP, 92, 106, 128
PPP over Ethernet, 92, 114, 142
PPPoE, 92, 114, 142
PPTP, 214
Präambel, 83, 107
Präfix, 144
Preamble, 83, 107
Prefix, 144
Presentation Layer, 43
Primärbereich, 72
Primary Area, 72
Private Addresses, 139
Private Adressen, 139
Private Port, 173
Protocol, 35
Protocol Translator, 22
Protokoll, 35
Protokollumsetzer, 22
PSE, 71
Pseudo-Header, 174, 175

Rahmen, 38, 91, 104
Rauschen, 73
Receive Buffer, 184
Receive Window, 185
Reference Models, 35
Registered Port, 173
Remote Access VPN, 213
Repeater, 21, 73
Resource Record, 196
Return-to-Zero, 80
Ring Network, 25
Ring-Topologie, 25
Ringleitungsverteiler, 26
RIP, 149, 150
RIPng, 154
RJ45, 65–67
Root Name Server, 197
Root-Nameserver, 197
Round Trip Time, 31, 74, 119, 192
route (Command), 228
route (Kommando), 228
Router, 22, 39, 131
Routing, 39, 131, 149
RTT, 31, 74, 119, 192
Rundlaufzeit, 74, 119, 192

SATA, 19
Satellit, 62
Satellite, 62
Scatternet, 60
Schichtenmodell, 35
Schiebefenster, 184
Schwellenwert, 189
Scrambler, 82
Secondary Area, 73
Segment, 39, 40, 171, 174, 176
Sekundärbereich, 73
Semantic, 35
Semantik, 35
Sendefenster, 185
Sendepuffer, 185
Sentinel Character, 105

Sentinel-Zeichen, 105
Serial Transmission, 19
Serielle Übertragung, 19
Service, 35
Session Layer, 42
SFTP, 211
Shared Media, 32
Shielding, 68
Short Preamble Format, 109
Shortest Path First, 155
Sicherungsschicht, 38, 91
Signal Weakening, 28, 73
Signalabschwächung, 28, 73
Signalisierungsgeschwindigkeit, 29
Silly Window Syndrom, 187
Silly Window Syndrom Avoid., 187
Simplex, 21
Single-Mode Fiber, 72
Site-to-Site VPN, 213
Sitzungsschicht, 42
Sliding-Window, 184
Slot Time, 119, 125
Slow Start, 189
SMTP, 41, 207
Socket, 173
Spannbaum, 96
Spanning Tree, 96
Spanning Tree Protocol, 97
Speed of Light, 30
Speichergröße, 7
Split Horizon, 153
SSH, 41, 203, 234
ssh (Command), 234
ssh (Kommando), 234
SSID, 50
SSL, 215
Star Network, 26
Start Frame Delimiter, 107, 109
Station, 23
Stern-Topologie, 26
Storage Size, 7
STP, 97
Stream Socket, 173
String, 12
Structured Cabling, 72
Strukturierte Verkabelung, 72
Subnet, 137, 138
Subnet Address, 138
Subnet Mask, 137
Subnet Präfix, 146
Subnet Prefix, 146
Subnetz, 137, 138
Subnetzadresse, 138
Switch, 22, 27, 38, 39, 132
Symbol Rate, 29
Symbolrate, 29
Synchrone Übertragung, 20
Synchronisierung, 79
Synchronization, 79
Synchronous Transmission, 20
Syntax, 35

Taktquelle, 20
Taktrückgewinnung, 79
TCP, 40, 175
    AIMD, 193
    Congestion Avoidance, 190
    Congestion Control, 187
    Congestion Window, 188
    Connection Termination, 182
    Dreiwege-Handshake, 179
    Empfangsfenster, 189
    Fast Recovery, 191
    Fast Retransmit, 190
    Fenstergröße, 189
    Fensterskalierung, 193
    Flow Control, 183
    Flusskontrolle, 183
    Receive Window, 189
    Reno, 191
    Schiebefenster, 184
    Schwellenwert, 189
    Segment, 176
    Silly Window Syndrom, 187
    Silly Window Syndrom Avoid., 187
    Sliding-Window, 184
    Slow Start, 189
    Tahoe, 190
    Three-Way Handshake, 179
    Threshold, 189
    Überlastkontrolle, 187
    Überlastungsfenster, 188
    Vegas, 192
    Verbindungsabbau, 182
    Window Scale, 193
    Window Scale Option, 193
    Window Size, 189
    WSopt, 193
TCP/IP Reference Model, 36, 43
TCP/IP-Referenzmodell, 36, 43
tcpdump (Command), 227
tcpdump (Kommando), 227
Telefonmodem, 92

Telephone Modem, 92
Telnet, 16, 41, 43, 203, 234
telnet (Command), 234
telnet (Kommando), 234
Teminal, 16
TERA, 70
Terminal Device, 23
Tertiärbereich, 73
Tertiary Area, 73
Tetrade, 6
TFTP, 202
Thick Ethernet, 24, 46, 63, 75, 121, 122
Thin Ethernet, 24, 32, 33, 63, 64, 75
ThinWire, 64
Three-Way Handshake, 179, 180
Threshold, 189
Time Division Multiplexing, 60
Time Window, 20
TLD, 196
TLS, 215
Token Passing Scheme, 32
Token Ring, 18, 25, 26, 32, 47, 84, 106, 107, 114, 118, 176
Token-Passing-Verfahren, 32
Top Level Domain, 196
Topologie, 24
Topology, 24
Traceroute, 167
traceroute (Command), 229
traceroute (Kommando), 229
Transceiver, 63
Transmission Control Protocol, 175
Transmission Medium, 17
Transmit Buffer, 185
Transmit Window, 185
Transport Layer, 36, 39, 171
Transport Layer Protocol, 171
Transportprotokoll, 171
Transportschicht, 36, 39, 171
Tree Network, 27
Triggered Updates, 150
Twisted Pair, 17, 64

Überlastkontrolle, 187
Überlastungsfenster, 188
UDP, 40, 174
Unicast, 134
Unicode, 12
Unipolar RZ, 81
Unipolares RZ, 81
Uplink, 27
Uplink Port, 67
US-ASCII, 9, 114, 115, 204
USB, 18, 88, 116
User Datagram Protocol, 174
UTF-8, 12

Vampire Tap, 63
Vampirklemme, 63
Velocity Factor, 30
Verbindungslos, 40
Verbindungsorientiert, 40
Verfallsdatum, 200
Verkürzungsfaktor, 30
Vermittlungsschicht, 39, 131
Vernetzungstechnologien, 45
Vertical Communication, 41
Vertikale Kommunikation, 41
Verwürfler, 82
Verzögerung, 29
Virtual LAN Trunk, 216
VLAN, 97, 98, 107, 108, 215
Vollduplex, 21
VPN, 92, 132, 213

WAN, 18
Wechselbetrieb, 21
Wechselspannung, 28
Wegbestimmung, 149
Weiterleitung, 148
Weiterleitungstabelle, 93
Well-known Port, 173
WEP, 57
Wide Area Network, 18
WiFi, 17, 48, 253
WiMAX, 18
Window Scale, 193
Wireless Transmission, 62
WLAN, 17, 18, 27, 48, 62, 109, 142
WPA, 57
WPA2, 59

Zahlen, 4
Zeichenkette, 12
Zeichenkodierung, 9
Zeichenstopfen, 105
Zeitfenster, 20
Zeitmultiplexverfahren, 60
Zellen-Topologie, 27
Zone, 196
Zugriffsverfahren, 32, 33
Zyklische Redundanzprüfung, 115

Springer Vieweg

springer-vieweg.de

Jetzt im Springer-Shop bestellen:

springer.com/978-3-662-53142-6

Springer Vieweg

springer-vieweg.de

Jetzt im Springer-Shop bestellen:

springer.com/978-3-662-57468-3