Patrick Michl

Netzwerktechnik

Grundlagen und Einführung

1. Auflage 2011

INHALT

NETZWERKTECHNIK

Grundlagen und Einführung

Seite

ÜBERSICHT

Die Komplexität der Vorgänge die heute in einem Computernetzwerk, insbesondere im Internet auftritt kann nur schwer überschätzt werden. Der Vergleich mit beispielsweise Suaheli zeigt aber mindestens zwei faktische Unterschiede: Erstens können weltweit zwar in etwa 80 Millionen Menschen Suaheli sprechen, hingegen kommunizieren jetzt in diesem Augenblick fast eine Milliarde Endgeräte (PCs, Notebooks, Smartphones, etc.) über das Internet miteinander. Zweitens kennt die „Sprache des Internet“ keine lokalen Dialekte, sondern beruht auf Standards, die weltweit gleich sind, da jedes Endgerät mit jedem anderen in der Lage sein muss Informationen auszutauschen.

Die Standardisierung in einem so enormen Umfang erforderte einen langatmigen Prozess, der verschiedenste bestehende und auch zukünftige Technologien berücksichtigen musste. Prof. Andrew S. Tanenbaum formulierte diesen Zusammenhang optimistisch: „*The nice thing about standards is that you have so many to choose from*.“ (Computer Networks, 2nd ed., p. 254). Zielführend für eine globale Standardisierung war die Aufteilung der technischen „Kommunikationsaufgaben“ bzw. technischen Anforderungen in eine Hierarchie, die auf verschiedenen Schichten basiert. Die niedrigen Schichten übernehmen „niedrigere“, also weniger komplexe Arbeiten, höhere Schichten entsprechend komplexere. Die „Sprache“ die innerhalb einer Schicht gesprochen wird orientiert sich selbstverständlich an deren Aufgaben bzw. dem Arbeitsumfeld. So spielen bei niedrigeren Schichten eher die Begriffe „Signal“, „Bit“ oder „Byte“ eine Rolle - komplexere „Daten“ sind ihnen sogar gänzlich fremd - andererseits haben höhere Schichten die an „Dateien“ oder „Streams“ gewöhnt sind keinerlei Bezug zu „Signalen“ etc.

Um nun einen Überblick über die Netzwerktechnik zu erhalten wollen wir uns schrittweise annähern. Dabei wird zugunsten eines schnellen Einstieges in den ersten beiden Kapiteln auf Vorüberlegungen bzgl. der Schichten verzichtet und die Netzwerktechnik stattdessen problemorientiert entwickelt. Beginnend bei der niedrigsten Schicht, der physikalischen Signalübertragung werden aufbauend durch Berücksichtigung zusätzlicher Anforderungen immer leistungsfähigere Kommunikationsstrukturen entworfen. Das erklärte Ziel ist es hierbei die Kommunikation zwischen vielen Teilnehmern durch eine geeignete Infrastruktur zu ermöglichen und dabei verschiedene Aspekte der Kommunikation zu berücksichtigen. Diese Vorgehensweise ermöglicht es die Vokabeln der einzelnen Schichten einzuführen und zu diskutieren. Im dritten Kapitel werden wir uns anschließend mit den Schichten mittels sogenannter Schichtenmodelle im theoretischen Sinne sowie der Umsetzung mittels Protokolle beschäftigen. Auf diesen aufbauend werden wir und schließlich im vierten Kapitel der konkreten Umsetzung der Kommunikation im Internet mittels der „Internetprotokollfamilie“ bzw. „TCP/IP Protokollfamilie“ widmen, wobei wir uns auf die wichtigsten Protokolle beschränken werden.

1 DATENÜBERTRAGUNG

Ausgangspunkt für die Kommunikation in einem Netzwerk ist die Informationsübertragung von einem Teilnehmer zu einem anderen. In Abschnitt 1.1 wollen wir uns daher zunächst der technischen Umsetzung einer „einfachen Datenübertragung“ widmen, wobei „einfach“ in dem Sinne zu verstehen ist, dass Daten ausschließlich von einem bestimmten Teilnehmer zu einem anderen übertragen werden sollen, ohne die Rückrichtung oder sogar weitere Teilnehmer zu berücksichtigen. Die hierfür erforderliche Theorie bewegt sich zwischen der physikalischen Ebene und der Informationsebene und bedient sich einer Sprache basierend auf „Signalen“, „Daten“ und „Codierungen“.

Diese Begrifflichkeiten werden daher in den Unterabschnitten 1.1.1 und 1.1.2 entwickelt, um sie anschließend in 1.1.3 und 1.1.4 in ein Datenübertragungssystem umzusetzen. Als nützliches Werkzeug erhalten wir dabei eine kompakte Beschreibung einfacher Datenübertragungen mittels sogenannter Übertragungskanäle. Hierdurch ist es uns möglich bei weiteren Betrachtungen stillschweigend auf physikalische Betrachtungen zu verzichten, um uns abstrakteren Anforderungen. In den folgenden Unterabschnitten 1.1.5 bis 1.1.7 wird die Funktionalität des Datenübertragungssystems um Fehlerkorrektur, Datenkompression und Verschlüsselung erweitert. Abschließend folgt in Unterabschnitt 1.1.8 eine tabellarische Gegenübergestellung der verschiedenen Codierungen, die bei einem Datenübertragungssystem zum Einsatz kommen.

Mit dem Ziel einer Kommunikation zwischen mehreren Teilnehmern wollen wir uns in Abschnitt 1.2 insbesondere zwei Problemen stellen: Mit den bisherigen Möglichkeiten würde für jede Datenübertragung von einem Teilnehmer zu einem anderen stets ein exklusiver Kanal vorausgesetzt. Wenn man nun die Anzahl der Teilnehmer immer weiter erhöht, wird die Anzahl der benötigten Kanäle unverhältnismäßig hoch: Bei hundert Teilnehmern bräuchte man immerhin knapp zehntausend Kanäle um alle miteinander zu verbinden! Diese ernüchternde Zahl lässt keine andere Option, als die gemeinsame Nutzung von Kanälen.

Dadurch ergibt sich unser erstes Problem: Wie können Kanäle gemeinsam genutzt werden? Die Lösung liefern sogenannte „Kanalteilungsverfahren“, welche wir in den Unterabschnitten 1.2.1 und 1.2.2 behandeln werden. Zunächst werden wir uns mit dem Spezialfall von zwei Teilnehmern beschäftigen, wie er beispielsweise bei einem Telefonat auftritt. Dadurch rückt eine weitere Kanaleigenschaft in den Vordergrund: Die Richtung der Datenübertragung. Kann nur in eine Richtung übertragen werden, oder abwechselnd oder sogar in beide Richtungen gleichzeitig? Anschließend wollen wir den Fall auf eine beliebige Anzahl von Teilnehmern verallgemeinern.

Angenommen ein Übertragungskanal wird nun von mehreren Teilnehmern gemeinsam genutzt. Während der Kommunikation kommt zu diesen ein neuer Teilnehmer hinzu und möchte ebenfalls auf den Kanal zugreifen. Dadurch ergibt sich das zweite Problem: Wie können mehrere Teilnehmer ohne vorherige Absprache über einen gemeinsamen Übertragungskanal kommunizieren, bzw. wie kann sich die Anzahl der Teilnehmer flexibel ändern? In Unterabschnitt 1.2.3 werden wir diesem Problem mit sogenannten „Zugriffsverfahren“ begegnen. Diese definieren eine grundsätzliche Regelung des Kommunikationsablaufes, so dass einzelne Teilnehmer keine Information über ihre Umgebung (im Sinne anderer Teilnehmer) besitzen müssen um auf einen Kanal zuzugreifen.

1.1 Einfache Datenübertragung

1.1.1 Signale

Def. 1.1 (Signale) *Ein Signal bezeichnet die physikalische Repräsentation von Information. Die Eigenschaften eines Signals die zur Rekonstruktion der Information notwendig sind werden als Signalkomponenten bezeichnet.*

Folgende Signalkomponenten werden unterschieden:

- *Ortskomponenten* dienen der Identifikation von räumlich getrennten Übertragungen, wie z.B. die Koordinaten auf einer Bildschirm-Fläche, oder eine Identifikationsnummer für einen elektrischen Leiter.

- Die *Zeitkomponente* dient der Identifikation des Zeitpunktes bei der Übertragung. Bei der Übertragung von Information mittels eines Signals entspricht die Zeitkomponente in vielen Fällen der Reihenfolge der übertragenen Messwerte der Wertkomponenten.

- Aus den *Wertkomponenten* werden die Informationen, die zur Rekonstruktion der zu übertragenden Information benötigt werden ermittelt. Die Messwerte der Wertkomponenten werden als *Signalwert* bezeichnet und mit Hilfe vektorwertiger Funktionen über die Zeit und den Ort dargestellt. Ist der Signalwert ausschließlich von der Zeit (dem Ort) abhängig, so spricht man auch von *Zeitsignalen (Ortssignalen).*

Die Signalkomponenten dienen der Klassifikation von Signalen, indem zwischen abgestuften und stufenlosen Veränderungen ihrer Werte unterschieden wird. Dementsprechend wird ein Signal als *diskret* oder als *kontinuierlich* bezüglich einer bestimmten Signalkomponente bezeichnet. Zeitsignale werden wie folgt klassifiziert:

- *Analoge Signale* sind sowohl kontinuierlich bezüglich der Zeitkomponente, als auch der Wertkomponenten. Da in der Natur kein „Sprünge“ auftreten, entspricht diese der natürlichen Gestalt von Signalen. Analoge Signale werden durch zeitliche Funktionen mit kontinuierlichen Signalwerten dargestellt (Bsp. 1.1)

- *Zeitdiskrete Signale* sind kontinuierlich bezüglich der Wertkomponenten und diskret bezüglich der Zeitkomponente. Sie entstehen aus analogen Signalen durch zeitliche *Quantisierung* und werden zur Reduktion der Informationsmenge verwendet. Zeitdiskrete Signale werden als Folgen kontinuierlicher Signalwerte dargestellt. (Bsp. 1.2)

- *Wertdiskrete Signale* sind das Gegenstück zu zeitdiskreten Signalen und entstehen aus analogen Signalen durch Quantisierung der Wertkomponenten. Zeitdiskrete Signale werden durch zeitliche Funktion dargestellt, wobei die Signalwerte den Zeichen aus einem endlichen Zeichenvorrat entsprechen.

- *Digitale Signale* sind sowohl diskret bezüglich der Zeitkomponente, als auch der Wertkomponenten und können somit als Folge von Zeichen (aus einem endlichen Zeichenvorrat) dargestellt werden.

Bsp. 1.1 (Elektrischer Schwingkreis) *In einem elektrischen Schwingkreis wird an zwei Punkten die Spannung U(t) gemessen. Die Raumkomponente identifiziert dabei die Messung (1, 2), die Zeitkomponente den Zeitpunkt t der Messung und die Wertkomponenten $U_1(t)$ und $U_2(t)$ die jeweilige Spannung. Da sowohl t, als auch $U_1(t)$ und $U_2(t)$ kontinuierliche Werte annehmen können, handelt es sich um ein analoges Signal.*

Bsp. 1.2 (Filme) *Ein Beispiel für zeitdiskrete Signale liefert die Filmtechnik. Mittels der sog. „Nachbildwirkung" und des „Stroboskopeffekts" wird die Illusion einer fließenden Bewegung erzeugt, obwohl nur einzelne Bilder in kurzen Abständen projiziert werden.*

Signalübertragungssysteme

Die physikalische Übertragung eines Signals erfolgt mittels eines (*Signal-*)*Leiters*, also eines geeigneten Übertragungsmediums. Um das Signal nun an die physikalischen Eigenschaften des Übertragungsmediums anzupassen ist es häufig notwendig dieses mittels eines sog. „Trägers" aufzubereiten. Unter einem „Träger" versteht man ein analoges Zeitsignal mit Amplitude, Phase und Frequenz.

Man unterscheidet bei der Übertragung mittels eines Trägers zwischen dem *Nutzsignal* und dem *Trägersignal*. Sollen nun Daten mittels eines Trägers übertragen werden, so wird die Amplitude, die Phase oder die Frequenz des Trägersignals durch das Nutzsignal angesteuert (bzw. verändert). Dieses Verfahren wird als *Modulation* (v. lat.: modulatio, „Takt") bezeichnet - Entsprechend spricht man je nachdem welcher Parameter verändert wird von *Amplituden-*, *Phasen-* oder *Frequenzmodulation*. Das Trägersignal wiederum dient der Ansteuerung eines *Senders*, der z.B. aus einem Oszillator, einem Verstärker und einer Sendeantenne besteht. Mittels des Senders wird das Trägersignal in den (Signal-)Leiter (das Übertragungsmedium) z.B. in Form von elektromagnetischen Wellen induziert.

Auf der Gegenseite wird das Trägersignal durch einen *Empfänger* erfasst, der z.B. aus einer Empfangsantenne und einem Verstärker besteht. Anschließend folgt die Umsetzung der Parameter des Trägersignals in das Nutzsignal. Dieses Verfahren wird als *Demodulation* bezeichnet.

Def. 1.2 (Signalübertragungssystem) *Ein System zur räumlichen Übertragung von Signalen heißt Signalübertragungssystem. Ein Signalübertragungssystem besteht aus Sender, (Signal-)Leiter und Empfänger, sowie einem Modulator und ein einem Demodulator, falls ein Trägersignal verwendet wird. Je nach dem Signaltyp der übertragenen Nutzsignale spricht man von analogen, zeitdiskreten, wertdiskreten oder digitalen Signalübertragungssystemen.*

Störsignale

Während der Übertragung von Signalen können Störbeeinflussungen in Form von *Störsignalen* auftreten. Handelt es sich dabei um Störungen mit breitem unspezifischem Frequenzspektrum so spricht man auch von *Rauschsignalen*. Diese sind besonders bei Funkübertragungen wie WLAN von Bedeutung und können bei der Übertragung mittels abgeschirmter Kabel nahezu vernachlässigt werden. Manchmal werden solche Störsignale beabsichtigt durch einen Störsender erzeugt um eine Signalübertragung zu verhindern, oder einen Fehler anzuzeigen (Abschnitt 1.2.3). In diesem Fall spricht man von einem *Jammingsignal* (v. engl.: jamming, „stören").

1.1.2 Daten und Codierung

Im Hinblick auf die Netzwerktechnik werden im Folgenden ausschließlich digitale Signalübertragungssysteme behandelt. Hierbei kann das zu übertragende Signal stets als eine Folge von Zeichen dargestellt werden. Diese Zeichenfolgen werden als Daten bezeichnet.

Def. 1.3 (Daten) *Daten sind beliebige Folgen von Zeichen. Um eine inhaltliche Bedeutung der Daten zu berücksichtigen werden diese als Nachrichten bezeichnet.*

Die Darstellung von Nachrichten mittels Zeichenfolgen erfolgt mittels eines Codes.

Def. 1.4 (Code) *Ein Code bezeichnet ein System zur Darstellung von Nachrichten mittels Zeichenfolgen. Dabei werden die einzelnen Abschnitte der Folge zu atomaren Informationsbausteinen, den Codewörtern zusammengefasst.*

Die *Codewortlänge* bezeichnet die Anzahl der Zeichen, die ein Codewort umfasst. Zu gegebenen Daten bezeichnet die *Datenmenge* die Summe der Längen aller enthaltenen Codewörter. Im einfachsten Fall besitzen alle Codewörter die gleiche Länge (Bsp. 1.3, Bsp. 1.4). Bei einigen Codes werden jedoch explizit variable Codewortlängen verwendet (Bsp. 1.12). In diesem Fall kann zwar keine feste, jedoch zu gegebenen Daten eine *mittlere Codewortlänge* angegeben werden. Diese errechnet sich aus dem Verhältnis der Datenmenge zur Anzahl der Codewörter.

Def. 1.5 (Codierung) *Die eindeutige Zuordnung der Codewörter eines Codes zu den Codewörtern eines anderen Codes heißt Codierung.*

Man spricht von einer *verlustfreien* bzw. *entzifferbaren Codierung*, wenn die codierten Nachrichten durch eine weitere Codierung ohne Verlust in ihre ursprüngliche Darstellung gebracht werden können. Ist es nach der Codierung nicht möglich die ursprüngliche Nachricht wiederherzustellen so spricht man von einer *verlustbehafteten Codierung*.

Bsp. 1.3 (BCD-Code) *Der BCD-Code (engl. „Binary coded Decimal“) beschreibt die Ziffernweise Darstellung von Dezimalzahlen durch 4-Bit Codewörter. Dabei erfolgt die Zuordnung durch Binärdarstellung:*
0 → `0000`, 1 → `0001`, 2 → `0010`, 3 → `0011`, *usw.*

Bsp. 1.4 (ASCII-Code) *Beim ASCII-Code (American Standard Code for Information Interchange) werden Ziffern, Buchstaben und häufig auftretende Sonderzeichen durch 7-Bit Codewörter dargestellt: z.B.* A → `1000001`, 1 → `0110001`, ! → `0100001`

Datenübertragungssysteme

Zur Übertragung von Nachrichten ist die Unterscheidung zwischen *Nachrichtenzeichen* und *Signalzeichen*, bzw. zwischen *Nachrichtencode* und *Signalcode* sinnvoll und häufig erforderlich (Bsp. 1.5, Bsp. 1.6). Zur Umwandlung zwischen Nachrichtencode und Signalcode muss ein System zur Datenübertragung daher entsprechende Komponenten zur Codierung bereitstellen. Diese werden als *Encoder* (v. engl. „to encode“, verschlüsseln) bzw. als *Decoder* (v. engl. „to decode“, entschlüsseln) bezeichnet.

Def. 1.6 (Datenübertragungssystem) *Ein System zur räumlichen Übertragung von Daten heißt Datenübertragungssystem. Ein Datenübertragungssystem umfasst ein digitales Signalübertragungssystem, sowie Encoder und Decoder zur Umwandlung zwischen Signalcode und Nachrichtencode.*

Die Kombination aus einem Encoder und dem zugehörigen Decoder ändert die Darstellung einer Nachricht nicht, wenn die zugrunde liegende Codierung verlustfrei ist und die Übertragung mittels des Signalübertragungssystems verlustfrei erfolgt.

Digitale Signale werden üblicherweise durch Ziffernfolgen im Binärsystem dargestellt. Dies lässt sich darauf zurückführen, dass bei einer kleineren Anzahl von Signalwerten die Unterscheidbarkeit verbessert wird. Dabei wird das Signalzeichen 1 als *mark* (v. engl. „mark“, Abdruck) und das Signalzeichen 0 als *space* (v. engl. „space“,. Aussparung) bezeichnet. Datenmengen werden in der Einheit *Bit* (v. engl. „binary digit“, Binärzeichen) angegeben.

Bsp. 1.5 (Feuerzeichentelegrafie) *Entlang der historischen Grenzen des Römischen Imperiums befanden sich Linien aus Wachtürmen, sog. „Specula“ (v. lat. „Hoffnungsschimmer“) zur Übertragung von Nachrichten. Ein mögliches Verfahren hierfür beschreibt der griechische Geschichtsschreiber Polybios (ca. 200 - 120 v. Chr.): In zwei unterscheidbaren Personengruppen werden jeweils eine bestimmte Anzahl von Fackeln entzündet. Diese identifizieren die Zeichen einer Tabelle mittels der Zeile und der Spalte. z.B. das Signalzeichen„2 Fackeln, 3 Fackeln“ entspricht dem Nachrichtenzeichen in der 2. Zeile, 3. Spalte*

Bsp. 1.6 (Morse-Telegrafie) *1837 entwickelte der Erfinder Samuel Morse ein Verfahren zur elektrischen Telegrafie, bei dem Nachrichten mittels kurzer und langer Spannungsimpulse (Punkte ◦ und Striche –) über einen elektrischen Leiter übertragen werden. z.B. die Signalzeichen ◦◦◦−−−◦◦◦ entsprechen den Nachrichtenzeichen SOS*

Im Weiteren wollen wir uns der Realisierung eines einfachen Datenübertragungssystems widmen und dieses dann schrittweise um verschiedene Funktionen ergänzen.

1.1.3 Leitungscodierung

Eine grundlegende Anforderung an ein Datenübertragungssystem ist die Gewährleistung, dass Daten die vom Sender über die (Signal-)Leitung geschickt wurden vom Empfänger richtig interpretiert werden können. Zudem soll dabei der Datenverlust möglichst gering zu halten sein. Hierfür erfolgt eine Codierung in den *Leitungscode*, der sich an den technischen Bedürfnissen des Signalübertragungssystems orientiert. Leitungscodes werden Bitweise erzeugt, weshalb man hier auch von einer *bitorientierten Codierung* spricht. Die zu codierenden Daten werden dabei als *Bitstrom* aufgefasst. Typische Anforderungen an den Leitungscode umfassen:

- *Taktrückgewinnung*: Soll mehrmals in Folge das gleiche Signalzeichen übertragen werden, so sind die einzelnen Zeichen nur mehr durch die Zeitkomponente voneinander zu unterscheiden, da der Signalwert konstant bleibt. Um nun zu gewährleisten, dass vom Empfänger die richtige Anzahl von Signalzeichen erkannt wird, müssen Sender und Empfänger bezüglich eines bestimmten Zeittakts synchronisiert sein. Dies kann Empfängerseitig mit Hilfe der sog. Taktrückgewinnung realisiert

werden (Bsp. 1.7). Dabei wird der Zeittakt aufgrund bestimmter wiederkehrender Muster im Signal synchronisiert.

- *Gleichspannungsfreiheit*: Wird das Signal über einen elektrischen Leiter übertragen, so sollte das Signal keine Gleichspannungsanteile besitzen. (Bsp. 1.8)

- *Fehlererkennung*: Um Fehler bei der Übertragung zu erkennen werden verschiedene Ansätze verfolgt. Eine Möglichkeit hierfür ist die Ersetzung einzelner Bits durch komplexere Codewörter (Bsp. 1.7), wodurch ungültige Codewörter als Fehler identifiziert werden können. Eine weitere Möglichkeit ist die Übertragung zusätzlicher Information über die einzelnen Zeichen mittels einer zusätzlichen Wertkomponente (Bsp. 1.8).

Bsp. 1.7 (Manchester-Code) *Im Manchester-Code werden lange Übertragungsphasen mit dem gleichen Signalwert verhindert, indem eine 1 durch 10 und eine 0 durch 01 übertragen wird. Dadurch kann der Takt aus dem regelmäßigen Wechsel des Signalwertes zurückgewonnen werden. Fehler werden erkannt, falls dreimal in Folge eine 0 oder eine 1 gemessen wird.*

Bsp. 1.8 (AMI-Code) *Im AMI-Code (v. engl. Alternate mark Inversion) wird die 1 mit abwechselnder Polarität übertragen (+1, -1). Bei elektrischen Leiten führt dies zu Gleichspannungsfreiheit. Fehler werden erkannt, wenn zweimal in Folge eine 1 mit gleicher Polarität gemessen wird.*

1.1.4 Übertragungskanäle

Def. 1.7 (Übertragungskanal) *Übertragungskanäle bezeichnen eine informationstheoretische Beschreibung von Datenübertragungssystemen. Hierfür werden die physikalischen Eigenschaften durch äußere Beobachtungsgrößen abstrahiert.*

Übertragungskanäle besitzen folgende veränderliche (von Störsignalen abhängige) Beobachtungsgrößen:

- Die *Fehlerrate* bezeichnet den mittleren relativen Anteil der Datenmenge, die auf einem Übertragungskanal fehlerhaft übertragen wird. Ein Kanal wird als *deterministisch* bezeichnet, wenn die Fehlerrate 0 beträgt.

- Die *Datenrate* bzw. *Datenübertragungsrate* bezeichnet die mittlere Datenmenge die auf einem Übertragungskanal pro Zeiteinheit übertragen wird.

- Die *Latenzzeit* bezeichnet die mittlere Zeitdauer zur Übertragung eines Signalzeichens.

Des Weiteren besitzen Übertragungskanäle unveränderliche (von Störsignalen unabhängige) Beobachtungsgrößen:

- Die *Kanalkapazität* beschreibt die maximale Datenrate. Dabei muss jedoch zwischen einem Wert unter optimalen Bedingungen und einem Wert unter realen Bedingungen unterschieden werden: Die *theoretische Kanalkapazität* beschreibt den optimalen Wert unter der Annahme, dass keine Störsignale auftreten und somit kei-

ne Fehlerkorrektur (1.1.5) verwendet werden muss. Da in der Realität jedoch Störbeeinflussungen auftreten, wird in der *realen Kanalkapazität* eine Fehlerkorrektur berücksichtigt die eine angestrebte Fehlerrate ermöglicht.

- Auch bei der Latenzzeit können theoretische Werte ermittelt werden. Dabei ist wie bei der Kanalkapazität die Frage ob die Werte unter Berücksichtigung einer Fehlerkorrektur oder ohne bestimmt werden. Daher wird auch hier zwischen der *realen Latenzzeit* und der *theoretischen Latenzzeit* unterschieden.

1.1.5 Fehlerkorrektur

Eine wünschenswerte Funktion eines Datenübertragungssystems ist die Gewährleistung, Übertragungsbedingte Fehler in den Daten möglichst gering zu halten. Innerhalb eines Übertragungskanals können diese durch den Leitungscode zwar erkannt, jedoch nicht korrigiert werden. Daher sind hierfür zusätzliche Maßnahmen zur *Fehlerkorrektur* erforderlich. Dabei werden zwei Vorgehensweisen unterschieden:

- Bei der *Vorwärtsfehlerkorrektur* wird eine zusätzliche Codierung mit der Fähigkeit zur Fehlererkennung und Fehlerkorrektur eingesetzt. Die Fehlerkorrektur bewegt sich dabei im Rahmen einer gewissen Korrekturfähigkeit des Codes und kann daher keine absolute Fehlerfreiheit gewährleisten. Dafür ermöglicht dieser Ansatz die Einhaltung fester Zeitschranken bei der Übertragung.

- Bei der *Rückwärtsfehlerkorrektur* erfolgt vom Empfänger automatisch eine Anfrage zur Neusendung einer Nachricht, falls Fehler erkannt wurden. Dieser Ansatz benötigt keine weitere Codierung und ermöglicht dadurch die zusätzliche Informationslast bei stabilen Übertragungen sehr gering zu halten. Jedoch kann die Einhaltung fester Zeitschranken nicht gewährleistet werden.

Kanalcodierung

Die Maßnahmen zur Fehlerkorrektur erfolgen außerhalb des Übertragungskanales. Daher bezeichnet man die Codierung, die bei der Vorwärtsfehlerkorrektur eingesetzt wird als *Kanalcodierung* (Abb. 1.1).

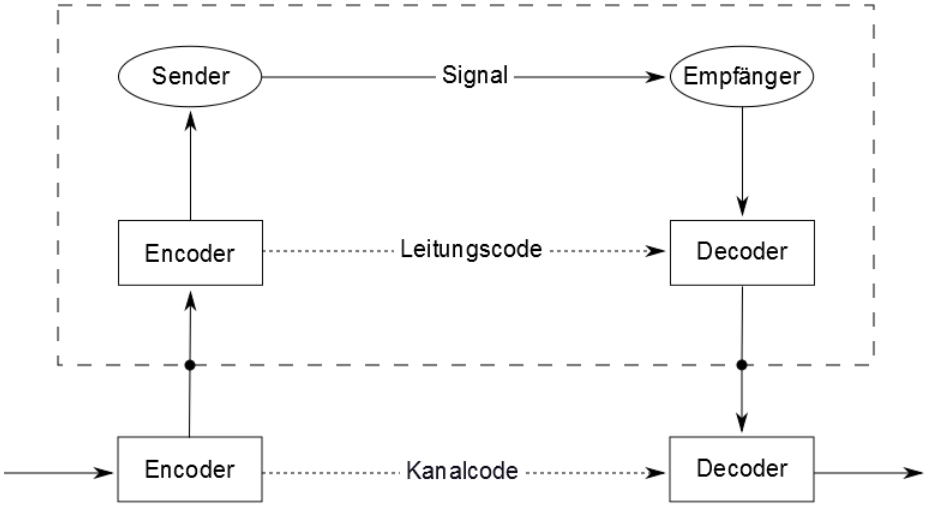

Abb. 1.1 *Kanalcodierung auf einem Übertragungskanal*

Man unterscheidet im Wesentlichen zwischen zwei Codeklassen zur Kanalcodierung:

- *Blockcodes* verwenden sog. *Paritätsbits*, um zusätzliche Informationen über jeweils gleich große Codeblöcke zu übertragen. Ein Paritätsbit gibt an, ob in einem gegebenen Codeblock eine gerade oder eine ungerade Anzahl des Signalzeichens 1 enthalten ist. Da zur Erzeugung der Blockcodes die Blöcke bei der Berechnung der Paritätsbits unabhängig voneinander codiert werden spricht man hier von *blockorientierter* Codierung (Bsp. 1.9)

- *Faltungscodes* bilden die zweite Klasse von Codes, die zur Vorwärtsfehlerkorrektur eingesetzt werden. Im Gegensatz zu Blockcodes werden Faltungscodes stromorientiert codiert. Das heißt, dass bei der Codierung von Zeichen Informationen der vorhergehend codierten Zeichen verwendet werden.

Bsp. 1.9 (VRC/LRC) *Beim vertical / longitudinal redundany check wird ein Bitstrom in jeweils j Codeblöcke der Länge i aufgeteilt. Zu diesen werden Zeilen- und Spaltenweise die Paritätsbits berechnet und dem Bitstrom beigefügt. Wenn nun ein Signalzeichen fehlerhaft übertragen wird, kann auf der Empfängerseite über die „falschen“ Paritätsbits sowohl die Zeile, als auch die Spalte des Fehlers und damit das fehlerhafte Bit identifiziert werden. Die Fehlerkorrektur erfolgt dann einfach durch Inversion des Bits. Da dies nur möglich ist, wenn in der gleichen Zeile und Spalte keine weiteren Bitfehler auftreten, kann mittels der Blockgröße mit den Codierungsparametern i und j Einfluss auf die Korrekturfähigkeit des Codes genommen werden.*

Interleaving

Def. 1.8 (Bitfehler) *Ein Bitfehler bezeichnet ein fehlerhaft übertragenes Signalzeichen, tritt ein Bitfehler mehrmals in Folge auf, so spricht man von einem Burstfehler. Die Beschreibung der Häufigkeit von Bitfehlern erfolgt zeitlich durch die sog. Bitfehlerrate in Bitfehler pro Sekunde, und anteilig durch das sog. Bitfehlerverhältnis in Bitfehler pro Signalzeichen.*

Sowohl Blockcodes als auch Faltungscodes ermöglichen zwar die Korrektur einzelner Bitfehler, jedoch nicht von Burstfehlern. Diese besitzen allerdings die Eigenschaft, dass sie relativ selten auftreten. Beim sog. *Interleaving* (v. engl.: interleave, „verschachteln“) wird sich diese Eigenschaft zunutze gemacht, indem der Bitstrom vor der Codierung in den Leitungscode nach einem bestimmten Muster umsortiert und nach der Rückcodierung wieder entmischt wird. Tritt nun bei der Übertragung ein Burstfehler auf, so wird dieser bei der Entmischung in einzelne Bitfehler zerteilt, welche mittels der Kanalcodierung korrigiert werden können.

Vergleich der Vorgehensweisen zur Fehlerkorrektur

Vorwärtsfehlerkorrektur ermöglicht es mittels verschiedener Codierungsparameter (Bsp. 1.9) Einfluss auf die Fehlerrate zu nehmen. Dabei führt eine bessere Fehlerrate automatisch zu einer geringeren realen Kapazität. Auch die reale Latenzzeit wird hierbei durch Interleaving erhöht. Dafür können sowohl die reale Kapazität, als auch die reale Latenzzeit innerhalb einer bestimmten Fehler-Toleranz als konstant angesehen werden. Dies ermöglicht es auf dem Übertragungskanal trotz Störbeeinflussung einen konstanten Datenstrom zu realisieren, wobei die maximale Fehlerrate vorgegeben wird. Dabei wird die reale Kanalkapazität mittels der Codierungsparameter optimiert. Die Vorwärtsfehlerkorrektur wird bei zeitkritischen Übertragungen eingesetzt. z.B. Übertragung von Videostreams

Durch Rückwärtsfehlerkorrektur sind die reale Kapazität und die realen Latenzzeit von der Bitfehlerrate abhängig. Bei Störbeeinflussungen sinkt daher die Datenrate ab und es kann kein konstanter Bitstrom realisiert werden. Treten jedoch keine Störbeeinflussungen auf, so entsprechen die reale Kapazität und die reale Latenzzeit den optimalen theoretischen Werten. Das ausschlaggebende Argument für den Einsatz von Rückwärtsfehlerkorrektur ist jedoch die Gewährleistung der Fehlerfreiheit, weshalb diese bei Inhaltskritischen Übertragungen eingesetzt wird. z.B. Übertragung von Dateien

1.1.6 Datenkompression

Ein bisher außer weitgehend Acht gelassener Aspekt ist die „Effizienz“ des Datenübertragungssystems. Dabei ist eine höhere Effizienz mit einer höheren *Datendurchsatzrate* gleichzusetzen. Die Datendurchsatzrate bezeichnet die mittlere (Daten-)Menge an Nutzdaten die auf einem Übertragungskanal pro Zeiteinheit übertragen wird. Im Gegensatz zur Datenübertragungsrate, welche rein von äußeren (physikalischen) Bedingungen bestimmt und mittels der realen Kanalkapazität beschränkt wird, kann die Datendurchsatzrate mittels einer geeigneten Codierung erhöht werden. Diese Codierung wird als „Kompression“ bezeichnet.

Def. 1.9 (Kompression) *Eine Codierung die zu einer Verringerung der Datenmenge führt heißt Kompression und ihre Umkehrung Dekompression.*

Das Datenübertragungssystem soll nun mit einer geeigneten Kompression ausgestattet werden. Um den Datendurchsatz zu erhöhen, gibt es zwei grundverschiedene Strategien bei der Kompression:

- **Erhöhung der Informationsdichte**

  Diese Vorgehensweise bedient sich der *verlustfreien Kompression*. Die Erhöhung der Informationsdichte wird hierbei erhöht, indem die mittlere Codewortlänge verringert wird. Werden beispielsweise bestimmte Codewörter nicht benötigt, so können diese aus dem Code entfernt und anschließend die restlichen Codewörter mittels kürzerer Codewörter codiert werden. (Bsp. 1.10, Bsp. 1.11, Bsp. 1.12)

  Bei der verlustfreien Kompression wird ein Code angestrebt, der eine möglichst kleine mittlere Codewortlänge aufweist. Die theoretisch minimale mittlere Codewortlänge die bei gegebenen Daten erreichbar ist wird als *Entropie* bezeichnet. Die Differenz zwischen der mittleren Codewortlänge und der Entropie heißt *Redundanz*. Daher heißt die entsprechende Verfahrensweise auch *Redundanzreduktion*.

- **Verringerung der Informationsmenge**

  Bei einer Verringerung der Informationsmenge spricht in naheliegender Weise von einer *verlustbehafteten Kompression*. Hierfür werden mittels entsprechender Modelle inhaltliche Überprüfungen der Daten vorgenommen die es erlauben zwischen „relevanten“ und „irrelevanten“ Daten zu unterscheiden. Häufig beruhen diese auf psychophysikalischen Aspekten der menschlichen Wahrnehmung, wie z.B. die Nichthörbarkeit von Frequenzen außerhalb 15Hz – 20kHz etc. (Bsp. 1.2).

  Das Ziel bei der verlustbehaftetet Kompression ist die Anteile irrelevanter Daten, der sog. *Irrelevanz* zu minimieren. Diese Verfahrensweise wird daher als *Irrelevanzreduktion* bezeichnet.

Bsp. 1.10 (Lauflängenkompression) *Bei der Laufläng̈enkompression werden in Folge auftretende Nachrichtenzeichen durch einen einzigen Repräsentanten sowie der Anzahl der darauf folgenden Wiederholungen des Zeichens codiert.*

Bsp. 1.11 (Wörterbuchkompression) *Bei der Wörterbuchkompression werden Daten auf wiederkehrende Blöcke untersucht. Diese können beispielsweise einzelne Wörter, Sätze, Bilder etc. darstellen. Anschließend wird jeweils ein Repräsentant dieser Blöcke separat in ein sog. Wörterbuch aufgenommen und alle Vorkommen durch eindeutige Symbole ersetzt. Diese Kompression wird daher auch als Substitutionskompression bezeichnet.*

Bsp. 1.12 (Huffman-Code) *Der Huffman-Code beschreibt eine binäre Darstellung von beliebigen Zeichenfolgen, wobei die Häufigkeit des Auftretens eines Zeichens berücksichtigt wird. Dabei werden seltener auftretende Zeichen durch längere Codeworte (im Binärsystem) dargestellt und häufiger auftretende Zeichen durch kürzere. Eine bemerkenswerte Eigenschaft des Huffman-Codes ist es stets eine minimale mittlere Codewortlänge zu erzeugen.*

1.1.7 Verschlüsselung

Ein weiterer wichtiger Aspekt der Datenübertragung ist die Gewährleistung der *Vertraulichkeit* von Informationen. Gemeint ist damit die Sicherheit gegenüber ungewollter Informationsweitergabe an Dritte. Insbesondere in Netzwerken können Übertragungskanäle als physikalische „Blackboxes“ auftreten, welche weder der Kontrolle des Senders, noch der

des Empfängers unterliegen. In so einem Falle spricht man von einem *unsicheren Übertragungskanal*. Damit ein Datenübertragungssystem die Vertraulichkeit von Daten über einen unsicheren Kanal gewährleisten kann wird hierfür eine weitere Codierung eingesetzt (Bsp. 1.13).

Def. 1.10 (Verschlüsselung) *Als Verschlüsselung wird eine verlustfreie Codierung bezeichnet, bei der für die Umkehrung die Kenntnis eines sog. Schlüssels erforderlich ist. Ein Schlüssel kann dabei beliebige Codierungsparameter umfassen. Die Umkehrung einer Verschlüsselung mit Hilfe eines Schlüssels wird als Entschlüsselung bezeichnet.*

Je nachdem ob bei der Verschlüsselung und Entschlüsselung gleiche oder unterschiedliche Schlüssel verwendet werden spricht man von *symmetrischer Verschlüsselung* oder von *asymmetrischer Verschlüsselung* bzw. von einem symmetrischen oder asymmetrischen Schlüssel. Grundlegend ist für eine symmetrische Verschlüsselung ein vorausgehender Austausch des Schlüssels und somit zumindest einmalig eine vertrauliche Datenübertragung erforderlich. Für die asymmetrische Verschlüsselung wird auf der Empfängerseite ein Schlüsselpaar erzeugt. Dieses besteht aus einem *öffentlichen Schlüssel*, der allgemein bekannt gemacht wird und einem *privaten Schlüssel*, der vom Besitzer geheim gehalten wird. Dabei stehen diese beiden Schlüssel in einem bestimmten mathematischen Verhältnis, welches es erlaubt Daten, die mit dem öffentlichen Schlüssel verschlüsselt wurden mit dem privaten Schlüssel zu entschlüsseln. Der Sender kann nun mit dem öffentlichen Schlüssel Daten vertraulich an den Empfänger schicken. Die hier zugrunde liegende Verfahrensweise wird auch als *Public-Key-Verfahren* bezeichnet.

Unabhängig von der Ermittlung und der Art des Schlüssels unterscheidet man bei der Codierung zwischen „Blockverschlüsselung“ und „Stromverschlüsselung“:

- Die *Blockverschlüsselung* bezeichnet eine blockorientierte Codierung, bei der Daten in gleichgroße Blöcke aufgeteilt und unabhängig voneinander verschlüsselt werden. Entspricht die Blockgröße der Codewortlänge eines Zeichens und haben alle Zeichen die gleiche Codewortlänge, so spricht man auch von einer *Chiffrierung* (Bsp. 1.13).

- Unter *Stromverschlüsselung* wird eine bitorientierte (oder zeichenorientierte) Codierung verstanden, bei der aus einem Schlüssel ein sog. Schlüsselstrom generiert wird. Der Schlüsselstrom wird dann zur bitweisen (oder zeichenweisen) Verschlüsselung des Datenstroms verwendet.

Bsp. 1.13 (Caesar-Verschlüsselung) *Nach der Überlieferung des römischen Schriftstellers Suetonius Tranquillus verwendete der römische Kaiser Gaius Julius Caesar für seine militärische Korrespondenz eine Codierung, die auf der Verschiebung der Buchstaben im Alphabet beruhte. Eine besondere Variante stellt die sog. ROT13 Verschlüsselung dar, bei der die zweimalige Anwendung der Verschiebung um 13 Stellen die ursprüngliche Nachricht wiederherstellt.*

1.1.8 Codierungen bei der Datenübertragung

Codierung	Leitungsebene	Kanalebene	Datenebene
bitorientiert	*Leitungscodierung (z.B. Manchester-Code, AMI-Code)*	*Faltungscodierung*	*Stromverschlüsselung*
zeichenorientiert	-		*Rendundanzreduktion (z.B. Huffman-Codierung), Chiffrierung, Irrelevanzreduktion*
blockorientiert	-	*Blockcodierung (z.B. VRC/LRC)*	*Rendundanzreduktion (z.B. Läuflängencodierung), Blockverschlüsselung*

Tbl. 1.1 *Codierungen bei der Datenübertragung*

1.2 Datenübertragung bei mehreren Teilnehmern

1.2.1 Richtungsabhängigkeit

Einzelne Übertragungskanäle werden hinsichtlich der Kommunikation danach klassifiziert, in welche Richtung eine Übertragung stattfinden kann. Man unterscheidet die Richtungsabhängigkeiten von Übertragungen durch folgende Betriebsarten:

- *Simplex* bezeichnet eine Betriebsart, bei dem der Informationsfluss über einen Kanal nur in eine Richtung erfolgen kann. Beispiel: „Pager“
- *Halbduplex* bezeichnet eine Betriebsart, die einen wechselseitigen Informationsfluss in beide Richtungen ermöglicht. Beispiel: „Walkie-Talkie“
- *Vollduplex* bezeichnet schließlich eine Betriebsart, die den gleichzeitigen Informationsfluss in beide Richtungen ermöglicht. Beispiel: „Telefon“

Vollduplex-Betrieb kann nun durch verschiedene Verfahren erreicht werden. Stehen mindestens zwei unabhängige Kanäle zur Verfügung, die jeweils in eine Richtung im Simplex-Modus betrieben werden können, so können diese zu einem Vollduplex Kanal zusammengeschlossen werden. In diesem Fall steht dem Informationsfluss in eine bestimmte Richtung die entsprechende Kanalkapazität des jeweiligen Übertragungskanales fest zur Verfügung. Kann jedoch nur ein einziger Kanal verwendet werden, so sind für den Duplexbetrieb bestimmte Verfahren erforderlich. Die wichtigsten sind:

- *Frequenzduplex* ist ein Verfahren bei dem zwei Amplitudenmodulierte Trägersignale mit verschiedenen Frequenzen überlagert werden. Da bei der Überlagerung keine gegenseitige Beeinflussung auftritt, können die jeweiligen Frequenzbänder als unabhängige Kanäle betrachtet werden. Daher sind auch die realen Kanalkapazitäten der beiden Richtungen unabhängig.

- *Zeitduplex*: Hierbei wechselt die Richtung des Informationsflusses in kurzen zeitlichen Abständen. Dadurch werden die jeweiligen Bitströme auf der Senderseite immer wieder kurz unterbrochen und nur in den dafür vorgesehenen Zeitfenstern übertragen. Dabei teilt sich die Kanalkapazität zwischen den beiden Richtungen auf. Ein Vorteil dieser Methode zum Duplexbetrieb ist, dass bei einem „sehr einseitigen" Informationsfluss die Zeitfenster entsprechend angepasst werden können, so dass in eine Richtung eine größere reale Kanalkapazität zur Verfügung steht, als in die andere Richtung.

1.2.2 Multiplexverfahren

Da Übertragungskanäle häufig von mehreren Teilnehmern genutzt werden ist es notwendig Kanalteilungsverfahren, sog. *Multiplexing* (v. lat.: multiplex, „vielfach") einzusetzen. Bei diesen Verfahren wird jeweils ein Kanal in mehrere *Subkanäle* geteilt, welche für sich wieder als eigenständige Kanäle mit den entsprechenden Beobachtungsgrößen (Datenrate, Fehlerrate, Kanalkapazität, Latenzzeit) aufgefasst werden können. Umgekehrt werden bei diesen Verfahren mehrere einzelne Bitströme zur Übertragung zu einem einzigen gebündelt. Um die Bitströme nach der Übertragung auf der Empfängerseite wieder störungsfrei in einzelne Bitströme zerlegen zu können, müssen diese unterscheidbar sein. Um diese Unterscheidbarkeit zu gewährleisten, werden „orthogonale Signale" und „orthogonale Codes" verwendet.

Orthogonale Signale

Signale heißen *orthogonal*, wenn eine Überlagerung nicht die Unterscheidbarkeit der Einzelsignale beeinträchtigt. Man unterscheidet folgende Kanalteilungsverfahren, die auf orthogonalen Signalen basieren:

- *Raummultiplex*: Dieses stellt das einfachste Verfahren für das Multiplexing dar. Hierbei werden vor der Kommunikation die freien Kanäle unter den Teilnehmern so aufgeteilt, dass jedem Informationsfluss ein exklusiver Übertragungskanal zur Verfügung steht. Die Orthogonalität der Signale beruht daher auf einer räumlichen Trennung.

- *Frequenzmultiplex* bezeichnet eine Verallgemeinerung des Frequenzduplex auf eine beliebige Anzahl von Teilnehmern. Dabei wird jedem Informationsfluss ein exklusives Frequenzband zugewiesen, welches eine Trennung der Signale im Frequenzraum bewirkt. Dieses ist sowohl bei der Übertragung per Funk als auch bei der Übertragung per Kabel möglich.

- *Zeitmultiplex* bezeichnet eine Verallgemeinerung des Zeitduplex auf eine beliebige Anzahl von Teilnehmern. Man unterscheidet zwischen synchronem und asynchronem Zeitmultiplex.

Beim *synchronen Zeitmultiplex* teilen sich die Teilnehmer einen Kanal, wobei die Zeitfenster periodisch durchlaufen werden. Dies hat den Nachteil, dass auch bei Nichtverwendung durch den entsprechenden Teilnehmer ein Teil der gesamten Kanalkapazität für diesen aufgewendet werden muss.

Beim a*synchronen Zeitmultiplex* wird das Problem der Nichtausnutzung der Kapazität dadurch umgangen, indem ungenutzte Zeitfenster von allen Teilnehmern verwendet werden können. Damit eine anschließende Zuordnung der Zeitfenster zu den jeweiligen Informationsflüssen möglich ist, wird jedes durch einen Identifikator in Form eines vorausgehenden ID-Codes gekennzeichnet.

Orthogonale Codes

Ein Code heißt *orthogonal*, wenn alle Codewörter die gleiche Länge besitzen und für je zwei Codewörter c_i und c_j gilt: $c_i \bullet c_j = 1$, falls $i = j$ und $c_i \bullet c_j = 0$, falls $i \neq j$. Dabei ist „•“ definiert durch: $c_i \bullet c_j = (c_i\,(1) \cdot c_j(1) + c_i(2) \cdot c_j(2) + \ldots + c_i(l) \cdot c_j(l)) / l$. Die Orthogonalität eines Codes ist damit gleichbedeutend, dass eine Addition von Codewörtern nicht die Unterscheidbarkeit beeinträchtigt. Dadurch können mehrere Codewörter gleichzeitig übertragen werden. Diese Vorgehensweise wird beim sog. „Codemultiplexing“ verfolgt. (Bsp. 1.14)

Codemultiplex bezeichnet Kanalteilungsverfahren, die auf orthogonalen Codes basieren. Dabei wird jedem Sender ein Codewort zur Codierung einzelner Bits zugewiesen, wodurch die Anzahl der möglichen Sender durch die Anzahl der vorhandenen Codewörter begrenzt wird. Um die Bitrate nicht zu verändern, haben die codierten Bits eine wesentlich feinere zeitliche Auflösung (kürzeren Zeittakt), wodurch zur Übertragung eine sog. „Spreizung“ des Frequenzbandes nötig wird (Bsp. 1.14).

Bsp. 1.14 (Orthogonale Codes): Codewörter: $c_1 = (1,1)$, $c_2 = (1,-1)$.

Prüfung der Orthogonalität:
$c_1 \bullet c_1 = (1 \cdot 1 + 1 \cdot 1) / 2 = 1$, $c_1 \bullet c_2 = (1 \cdot 1 + 1 \cdot -1) / 2 = 0$, $c_2 \bullet c_2 = (1 \cdot 1 + -1 \cdot -1) / 2 = 1$
$\Rightarrow$ Der Code ist orth. und kann für Codemultiplexing mit zwei Sendern verwendet werden

Überlagerung:
Die Bitfolgen s_1 und s_2 werden nun mittels c_1 und c_2 codiert und überlagert.
Für das n-te Bit der neu entstandenen Bitfolge s gilt dann: $s(n) = s_1(n) \cdot c_1 + s_2(n) \cdot c_2$

Rückgewinnung von s_1 (bzw. s_2) aus s durch Multiplikation mit c_1 (bzw. c_2):
$s(n) \cdot c_1 = (s_1(n) \cdot c_1 + s_2(n) \cdot c_2) \cdot c_1 = s_1(n) \cdot c_1 \bullet c_1 + s_2(n) \cdot c_2 \bullet c_1 = s_1(n)$
$s(n) \cdot c_2 = (s_1(n) \cdot c_1 + s_2(n) \cdot c_2) \cdot c_2 = s_1(n) \cdot c_1 \bullet c_2 + s_2(n) \cdot c_2 \bullet c_2 = s_2(n)$

1.2.3 Zugriffsverfahren

Unter der Voraussetzung, dass mehrere Teilnehmer über einen Kommunikationskanal miteinander kommunizieren wollen ergibt sich bei fehlender Absprache ein Problem. Durch Multiplexing wird hierfür zwar die Anzahl der Kanäle erhöht, jedoch weiß ein einzelner Teilnehmer nicht wann und über welchen Kanal er senden darf. Die Regelung hierfür wird mittels sog. *Zugriffsverfahren* übernommen. Bei diesen wird der Datenstrom vom Sender in sog. „Datenframes“ (Def. 3.5) aufgeteilt. Wurde vom Empfänger ein kompletter Datenblock richtig erhalten, so schickt dieser eine „Quittung“.

ALOHA

1971 wurde an der Universität von Honolulu, Hawaii ein Verfahren mit der Bezeichnung ALOHA (v. haw. aloha, „Hallo“) entwickelt. Dieses beruht auf den folgenden Schritten:

1. Ein Sender der Datenframes senden möchte beginnt ohne Verzögerung. Mittels Kanalcodierung werden dabei Fehler erkannt.

2. Kollisionen zwischen mehreren Teilnehmern werden durch eine zentrale Station oder durch Empfang der eigenen Daten auf dem Übertragungskanal erkannt.

3. Tritt eine Kollision auf, oder wurde nach der Übertragung in einem bestimmten Zeitfenster keine Quittung erhalten, so wartet der Sender eine zufällige Zeitdauer und beginnt dann wieder von vorne mit der Übertragung. Diese zufällige Wartezeit ist wichtig um eine immer wiederkehrende Kollisionssituation zu vermeiden.

Eine offensichtliche Schwäche dieses Verfahrens ist die nicht genutzte Zeit in der gerade alle Teilnehmer warten und niemand sendet. Eine Möglichkeit dies zu umgehen bietet das sog. *slotted ALOHA*. Dieses benutzt die gleichbleibende Kanalkapazität (durch die Kanalcodierung) und die feste Datenblockgröße, um die Übertragungsdauer der Datenblöcke zu ermitteln. Dadurch entsteht ein Zeitraster (Slots), welches den Teilnehmern zugrunde gelegt wird. Tritt nun eine Kollision auf, so weiß ein Sender exakt wie lange er warten muss bis er es nochmal versuchen kann. Diese Verbesserung wird durch eine aufwändigere Umsetzung bezahlt, da eine zentrale Zeit-Synchronisation notwendig wird.

CSMA

Neben slotted ALOHA lässt sich das ALOHA Verfahren auch durch eine vorausgehende Prüfung des Kanals verbessern. Dieses Verfahren wird als *CSMA* (v. engl. Carrier Sense Multiple Access, „Mehrfachzugriff mit Trägerprüfung“) bezeichnet und ist auf ein Trägersignal angewiesen. Ein Sender der ein Datenframe senden möchte prüft den Kanal vorher auf die Existenz eines Trägersignals und damit auf eine bereits laufende Übertragung. Wird diese Übertragung erkannt, so wird eine der folgenden Strategien verfolgt:

- *Persistent*: Es wird solange geprüft, bis das der Kanal frei ist und dann gesendet.

- *Non-Persistent*: Es wird eine zufällige Zeit gewartet und dann erneut geprüft. Ist der Kanal frei, so wird gesendet.

Nun könnte man zunächst annehmen, dass durch diese Verfahrensweisen keine Kollisionen auftreten, allerdings wird dabei die Latenzzeit der Übertragung nicht berücksichtigt. Auch wenn ein Sender einen freien Kanal erkennt, so kann es sein, dass bereits ein anderer Sender eine Nachricht auf diesem Kanal überträgt, diese Übertragung wegen der hohen Latenzzeit auf dem Übertragungsmedium jedoch noch nicht vom potenziellen Sender erkennbar war. Wenn die Latenzzeit viel größer als die Übertragungszeit eines Datenframes ist, so entspricht das CSMA Verfahren dem ALOHA Verfahren und eine Non-Persistente Trägerprüfung ist erforderlich. Wenn die Latenzzeit jedoch sehr viel kleiner als die Übertragungszeit eines Datenframes ist, so treten keine Kollisionen auf und eine Persistente Trägerprüfung kann verwendet werden. In den anderen Fällen lässt sich CSMA durch zusätzliche Erweiterungen der Strategie verbessern:

- *CSMA/CD* (v. engl. Collision Detection, „Kollisionserkennung“) Durch eine zentrale Station werden Kollisionen erkannt und den einzelnen Teilnehmern durch ein Jammingsignal bekannt gegeben. Anschließend wird ein zufälliger Teilnehmer ausgewählt, der mit der Übertragung beginnen darf. Dieses Verfahren wird insbesondere verwendet, wenn Kollisionen eher selten auftreten, wie es beispielsweise bei der Kabelgebundenen Übertragung aufgrund der geringen Latenzzeit der Fall ist.

- *CSMA/CA* (v. engl. Collision Avoidance, „Kollisionsvermeidung“) Es wird versucht eine Kollision „bestmöglich“ zu vermeiden. Um dies zu bewerkstelligen sendet ein potenzieller Sender sobald er einen Kanal als frei erkannt hat nach einer zufälligen Wartezeit eine Anfrage (sehr kurz codiert, mit ID des Senders). Wenn die anderen Sender diese Anfrage erhalten und damit einverstanden sind, so schicken sie eine Quittung. Damit erhält der potenzielle Sender die Erlaubnis zum Senden.

1.3 Aufgaben zu Kapitel 1

Aufgabe 1.1

Erklären und unterscheiden Sie die Begriffe „Bitrate“, „Übertragungsrate“ und „Durchsatzrate“.

Aufgabe 1.2

Welche Fehlerkorrektur, Kompression und Verschlüsselung eignen sich zur Übertragung von Dateien, welche zur Übertragung von Sprachsignalen?

Aufgabe 1.3

Tauschen Sie in dem Wort „IT“ jeden Buchstaben durch die entsprechende Nummer im Alphabet. Bei einstelligen Nummern ergänzen Sie eine führende Null. Führen Sie anschließend eine Codierung in den BCD-Code durch. Schreiben Sie die Bitfolge dabei in vier Blöcken untereinander und ermitteln Sie die Paritätsbits für die Zeilen und die Spalten.

Aufgabe 1.4

Erklären Sie unter Verwendung der Tabelle in Aufgabe 1.3 warum der Empfänger bei Empfang des Wortes „ET“ mit Hilfe der zuvor ermittelten Paritätsbits den Fehler zwar erkennt, aber nicht automatisch korrigieren kann. Um was für eine Art von Übertragungsfehler handelt es sich? Durch welches zusätzliche Verfahren hätte der Fehler korrigiert werden können?

Aufgabe 1.5

Welche Zusammenhänge bestehen zwischen der realen Kanalkapazität und der Bitfehlerrate?

Aufgabe 1.6

Erklären Sie die Begriffe „Öffentlicher Schlüssel“ und „Privater Schlüssel“.

Aufgabe 1.7

Eine Code besitzt die beiden Codewörter: c_1 = (1,1,-1,-1), c_2 = (1,-1,-1,1). Eignet sich dieser Code für Codemultiplexing? Wie viele Sender sind dabei möglich?

Aufgabe 1.8

Ein Auto fährt auf eine Straßenkreuzung zu, erkennt dass bereits andere Autos auf der Straße sind und hält an. Sobald die Straße frei ist beschleunigt der Fahrer den Wagen. Welchem Zugriffsverfahren entspricht diese Vorgehensweise? Beschreiben sie ein weiteres Zugriffsverfahren in diesem Kontext.

2 NETZWERKE

In Kapitel 1 haben wir uns mit der einfachen Datenübertragung sowie der Datenübertragung zwischen mehreren Teilnehmern auseinandergesetzt. Dabei wurde aber stets vorausgesetzt, dass eine direkte Verbindung mittels eines Übertragungskanales (muss nicht exklusiv sein) besteht. In Netzwerken muss dies jedoch nicht gegeben sein, so dass Daten nicht direkt, sondern über Zwischenstationen zu ihrem Ziel gelangen. Ein Beispiel hierfür haben wir bereits mit der „Feuerzeichentelegrafie“ (Bsp. 1.5) kennengelernt, und das Ganze als „Datenübertragungssystem“ bezeichnet. An dieser Stelle sollte daher darauf hinwiesen werden, dass Überschneidungen zwischen den verschiedenen Begrifflichkeiten insbesondere bei solchen auf verschiedenen Betrachtungsebenen auftreten können. Im folgenden Kapitel wollen wir an die vorherige Theorie anknüpfen und die Verfahren der Datenübertragung auf komplexeren Verbindungsstrukturen untersuchen, bei denen die Teilnehmer nicht direkt über einen Übertragungskanal verbunden sein müssen. Diese komplexere Verbindungsstruktur bezeichnen wir als Netzwerk.

Mit zunehmender Anzahl von Teilnehmern steigen die Größe und die Komplexität eines Netzwerkes an. Dabei ist zu beobachten, dass die Wahl der Verbindungen unmittelbaren Einfluss auf die Kosten des Netzwerkes, die Ausfallssicherheit oder auch die Geschwindigkeit bei der Datenübertragung ausüben. Um diese Zusammenhänge zu verstehen werden wir uns in Abschnitt 2.1 mit den strukturellen Eigenschaften, der sogenannten „Topologie“ des Netzwerkes befassen. Zunächst soll in Unterabschnitt 2.1.1 die für diese Theorie grundlegende Sprache der „Graphen“ eingeführt werden. Anschließend wird diese Sprache in Unterabschnitt 2.1.2 aufgegriffen um Bewertungsgrößen für Netzwerke zu definieren. Diese Bewertungsgrößen werden in Unterabschnitt 2.1.4 verwendet um sogenannte „elementare Topologien“ zu studieren. Die Analyse wird auf komplexeren Netzwerken bis hin zum „Internet“ in 2.1.7 fortgesetzt.

Aufgrund der komplexen Verbindungsstruktur eines Netzwerkes kann es nun vorkommen, dass für eine Datenübertragung zwischen zwei Teilnehmern dem Sender nicht nur einer, sondern viele Übertragungskanäle zur Verfügung stehen. Der Problemcharakter wird augenscheinlich, sobald Sender und Empfänger nicht direkt über einen Übertragungskanal verbunden sind. Dann stellt sich für den Sender die Frage wohin er die Daten senden soll, damit diese näher an den Empfänger gelangen. In Abschnitt 2.2 werden wir uns daher mit der Wegsteuerung, dem sog. „Routing“ auseinandersetzen. Hierfür werden in Unterabschnitt 2.2.1 zunächst einige graphentheoretische Begriffe eingeführt, die es ermöglichen Verbindungen allgemein zu bewerten. In Unterabschnitt 2.2.2 werden wir dann konkrete Algorithmen zur Auffindung „kürzester Pfade“ bei der Datenübertragung betrachten. Würde nun jeder Teilnehmer die Daten auf dem jeweilig kürzesten Pfad übertragen, wäre das Problem theoretisch gelöst. Allerdings ist dies nicht die einzig mögliche und unter einigen Aspekten nicht die beste Strategie. In Unterabschnitt 2.2.3 wird die Verfahrensweise der Zwischenstationen, welche als „Vermittlung“ bezeichnet wird zunächst grob unterteilt. Anschließend werden in 2.2.4 Netzwerkweite Routing-Strategien diskutiert. Schließlich werden wir uns in Abschnitt 2.2.5 wieder dem Internet und den hierbei verwendeten Routing-Strategien zuwenden.

Durch die Vorarbeit in den Abschnitten 2.1 und 2.2 können wir uns in Abschnitt 2.3 noch einmal kurz mit einigen Aspekten der Kommunikation befassen. Diese werden in Ab-

schnitt 2.3.1 die Verbindungsformen und schließlich in Abschnitt 2.3.2 die Aufgabenverteilung einschließen.

2.1 Topologie

2.1.1 Graphen

Um Vorgänge in Netzwerken zu verstehen, zu analysieren oder zu planen ist es häufig erforderlich die Verbindungsstruktur einzubeziehen. Diese Struktur wird als Topologie des Netzwerkes bezeichnet und durch sog. Graphen (Def. 2.1) beschrieben. Dabei werden die enthaltenen Informationen auf ein Minimum beschränkt. Diese Informationen umfassen die Kommunikationsteilnehmer und die Übertragungskanäle (bzw. beliebige Verbindungen der Teilnehmer).

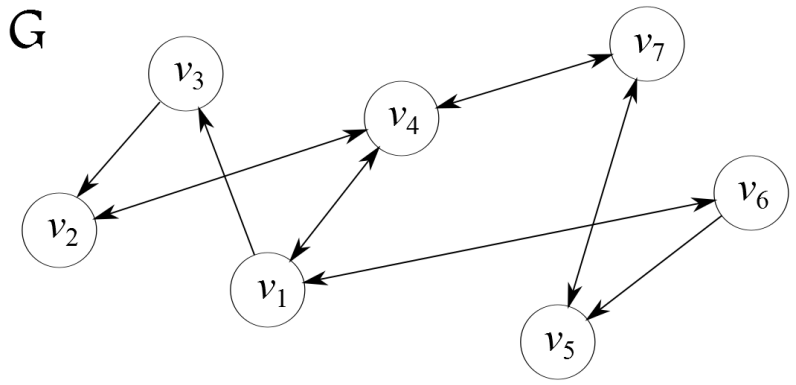

Abb. 2.1 *Modellierung eines Netzwerkes durch einen Graphen*

Def. 2.1 (Graphen) *Ein Graph G bezeichnet eine Struktur, die aus einer Menge von Knoten* $V = \{v_1, v_2, ...\}$ *und einer Menge von Kanten* $E = \{e_1, e_2, ...\}$ *besteht. Die Knoten bezeichnen dabei beliebige Objekte und die Kanten richtungsabhängige Verbindungen zwischen diesen.*

Kanten und Knoten

Eine Kante, die nur in eine Richtung zeigt, heißt *gerichtete Kante*, eine Kante die in beide Richtungen zeigt entsprechend *ungerichtete Kante*. Graphen, die ausschließlich ungerichtete Kanten enthalten, werden in dieser Konsequenz als *gerichtete Graphen*, ansonsten als *ungerichtete Graphen* bezeichnet.

Die Knoten eines Graphen werden aufgrund der Anzahl von Kanten mittels derer sie mit anderen Knoten verbunden sind unterschieden. Man bezeichnet einen Knoten als *Isolierten Knoten* bei keiner Kante, *Endknoten* bei einer einzelnen Kante und *Zwischenknoten* bei mehreren Kanten. Umgekehrt unterscheidet man Kanten nach der Anzahl der Knoten, die sie verbinden. Wenn eine Kante eine beliebige Anzahl von Knoten miteinander verbinden, so spricht man von *Point-to-Multipoint Verbindungen.* Wenn ein Kante genau zwei Knoten miteinander verbindet, so spricht man von einer *Point-to-Point Verbindung* .

Pfade

Sind zwei Knoten über eine Kante miteinander verbunden, so bezeichnet man diese als *Nachbarn* bzw. *benachbart*. Können sie mittels mehrerer aufeinander folgender Kanten miteinander verbunden werden, so heißen die Knoten *verbunden* und die Folge der Kanten ein *Pfad* zwischen diesen Knoten. Benachbarte Knoten sind also stets auch verbunden, da die Kante zwischen ihnen auch einen Pfad zwischen ihnen bildet.

Die *Länge eines Pfades* zwischen zwei Knoten bezeichnet die Anzahl der Kanten in der Kantenfolge des Pfades. Ein *kürzester Pfad* zwischen zwei Knoten bezeichnet somit einen Pfad mit einer minimalen Anzahl von Kanten. Die *Entfernung* zweier verbundener Knoten bezeichnet die Länge des (bzw. eines) kürzesten Pfades zwischen diesen. Man bezeichnet verschiedene Pfade als *unabhängig*, wenn sie keine gemeinsamen Kanten besitzen.

Häufig beschränkt man sich bei der Untersuchung von Graphen auf sog. *zusammenhängende Graphen* und meint damit, dass alle Knoten miteinander verbunden sind. Dieser Begriff kann noch präzisiert werden: Ist *k* die minimale Anzahl der unabhängigen Pfade zwischen zwei beliebigen Knoten, so heißt der Graph *k-fach zusammenhängend* und *k* heißt *Zusammenhangszahl* des Graphen. In einem 2-fach zusammenhängenden Graphen existieren also zwischen beliebigen Knoten mindestens zwei unabhängige Pfade.

2.1.2 Netzwerke

Def. 2.2 (Netzwerk) *Ein Netzwerk bezeichnet ganz allgemein ein System, dessen Verbindungsstruktur durch einen Graphen modelliert werden kann. Wird nur ein Teil eines Netzwerkes betrachtet, so spricht man von einem Teilnetz.*

Computernetzwerke

Computernetzwerke bestehen im aus „Hosts“ (v. engl. host „Veranstalter“, „Verarbeiter“), „Routern“ und (Daten-)„Links“.(v. engl. link, „Verbindung“). Ursprünglich bezeichnete man mit dem Begriff „Host“ sowohl Computer im Netzwerk, als auch beliebige andere Netzwerkperipherie. Zugunsten einer begrifflichen Unterscheidung zwischen Endstellen und Zwischenstellen bei der Datenübertragung ging man jedoch später dazu über die Zwischenstellen, also solche Hosts die zur Datenweiterleitung vorgesehen sind als *Router* und nur mehr die Endstellen als *Hosts* zu bezeichnen. Verbindungen zwischen Hosts/Routern, die der Übertragung von Daten dienen werden als (*Daten-*)*Links* bezeichnet. Links verallgemeinern damit Übertragungskanäle in dem Sinne, dass physikalische Eigenschaften vernachlässigt werden und schlichtweg die Möglichkeit Datenübertragung für einen Link zwischen zwei Teilnehmern ausreicht.

Bei der Modellierung von Computernetzwerken mittels Graphen ergeben sich damit folgende Zusammenhänge:

Computernetz-werk	Graph	Beschreibung
Host	*Endknoten*	Hosts bezeichnen Computer oder Netzwerkperipherie (wie Netzwerkdrucker) etc. und entsprechen Endknoten eines Graphen
Router	*Zwischenknoten*	Router sind Einrichtungen zur Weiterleitung von Daten und werden durch Zwischenknoten in einem Graph repräsentiert.
(*Daten-*)*Link*	*Kante*	Links abstrahieren Übertragungskanäle und werden mittels der Kanten des Graphen dargestellt. Dabei gibt die Richtung der Kante die Datenflussrichtung vom Sender zum Empfänger an. Gerichtete Kanten entsprechen also Kanälen im Simplex-Betrieb.
Route	*Pfad*	Pfade werden in Computernetzwerken als *Routen* bezeichnet. Ein Pfad, der eine Datenübertragung repräsentiert heißt auch *Übertragungsweg*.

Tbl. 2.1 *Modellierung von Computernetzwerken*

2.1.3 Topologien

Def. 2.3 (Topologie) *Die Verbindungsstruktur eines Netzwerkes / Teilnetzes wird als Topologie des Netzwerkes / Teilnetzes bezeichnet.*

In Computernetzwerken wird zwischen „physischer Topologie“ und „logischer Topologie“ unterschieden:

- Die physische Verbindungsstruktur der Übertragungskanäle in Form von Kabeln, oder WLAN-Verbindungen etc. wird als *physische Topologie* oder auch als *Signal-Topologie* bezeichnet. Bei der physischen Topologie entsprechen die Links also im Wesentlichen den (Signal-)Leitungen in Form von Übertragungskanälen.

- Die Verbindungsstruktur der Endstellen der Übertragungswege wird als *logische Topologie* bezeichnet. Bei dieser entsprechen die einzelnen Links den Übertragungswegen. Werden also Daten von *A* über *B* nach *C* übertragen besteht in der logischen Topologie des Netzwerkes ein Link zwischen *A* und *C*.

Sind die Knoten hinsichtlich ihrer strukturellen Aufgaben im Netzwerk unterscheidbar, so bezeichnet man eine Topologie als *asymmetrisch*, ansonsten als *symmetrisch*.

Insbesondere bei größeren Netzwerken ist es erforderlich Kenngrößen einzuführen, welche den Charakter des Netzwerkes beschreibt. Das Ziel dabei ist es Aussagen z.B. über den Aufwand, die Ausfallsicherheit oder die Leistungsfähigkeit des Netzwerkes aus diesen Kenngrößen abzuleiten. Andersrum ermöglichen solche Kenngrößen es ein Netzwerk nach vorgegebenen Anforderungen zu konstruieren.

Komplexität

Die *Komplexität* ist ein Maß für den *Aufwand einer Topologie*. Sie beschreibt die Größenordnung zwischen Kanten und Knoten bei einer zugrunde gelegten Struktur. Diese ergibt sich aufgrund der Beobachtung, wie sich bei einer schrittweisen Vergrößerung der Anzahl an Knoten die Anzahl der Kanten ändert.

Die Angabe der Komplexität erfolgt in der sog. O-*Notation*. Bezeichnet n die Anzahl der Knoten in einem Netzwerk, dann bedeutet: O(1) eine konstante Anzahl von Kanten, O(n) eine zu n linear anwachsende Anzahl von Kanten und O(n^2) eine zu n quadratisch anwachsende Anzahl von Kanten usw.

Konnektivität

Die *Konnektivität* ist ein Maß für die Stabilität einer Topologie. Dabei ist unter Stabilität die maximale Anzahl von tolerierbaren Ausfällen von Verbindungen zu verstehen. Die Konnektivität entspricht somit der Zusammenhangszahl des zugrunde liegenden Graphen. Bei einer Topologie mit der Konnektivität k lassen sich also zwei beliebige Knoten über mindestens k unabhängige Pfade verbinden.

Durchmesser

Der *Durchmesser* einer Topologie bezeichnet die maximale Entfernung, die zwischen Knoten in einem Netzwerk auftreten kann. Da die Entfernung ja immer dem kürzesten Pfad entspricht ist sie also die maximale Länge von kürzesten Pfaden zwischen beliebigen Knoten. Diese Kennzahl ist ein Maß für die maximale (tolerierbare) Länge von Übertragungswegen. Sie verallgemeinert damit in gewisser Weise den Begriff der Latenzzeit auf Netzwerke.

Bisektionsweite

Die *Bisektionsweite* ist ein Maß für die Leistungsfähigkeit einer Topologie. Unter der Leistungsfähigkeit ist dabei die Fähigkeit einer Topologie zur Lastverteilung zu verstehen. Bei Topologien mit Point-to-Point Verbindung entspricht sie der minimalen Anzahl von Kanten, die aus einem Graph mit n Knoten entfernt werden müssen, so dass zwei voneinander getrennte Graphen mit höchstens n / 2 Knoten entstehen. Bei Topologien mit Point-to-Multipoint Verbindungen entspricht sie der minimalen Summe über die anteiligen Kanalkapazitäten bei der Datenübertragung zwischen zwei beliebige Hälften (mit höchstens n / 2 Knoten). Sie verallgemeinert den Begriff der Kanalkapazität auf Netzwerke.

2.1.4 Elementare Topologien

Elementaren Topologien bezeichnen sehr einfache Verbindungsstrukturen, die als Grundbausteine für den Aufbau komplexerer Topologien dienen. Man unterscheidet zwischen

solchen elementaren Topologien, die auf Point-to-Multipoint Verbindungen basieren und solchen, die auf Point-to-Point Verbindungen basieren.

Bus

Die einfachste Topologie mit der geringsten Komplexität ist der *Bus* (Abb. 2.2a). Sie ist eine ist eine Point-to-Multipoint basierende Topologie und besitzt genau eine Kante, d.h. dass alle angeschlossenen Teilnehmer über einen gemeinsamen Übertragungskanal kommunizieren. Daher teilt sich die Kanalkapazität entsprechend dem eingesetzten Kanalteilungsverfahren auf die Teilnehmer auf.

Zelle

Die *Zelle* (Abb. 2.2b) ist eine Point-to-Multipoint basierende Topologie, die bei drahtlosen Netzwerken auftritt. Eine Zelle wird durch den Bereich um einen Knoten definiert, in welchem direkte Verbindungen zu anderen Knoten existieren. Dieser Bereich wird auch als *Versorgungsgebiet* bezeichnet. Wird ein zentraler Zugangsknoten (engl. *Wireless Access Point*) verwendet, so umfasst eine Zelle alle Knoten die in dessen Versorgungsgebiet liegen. Dabei entspricht die physische Topologie einem Bus und die logische Topologie einem Stern.

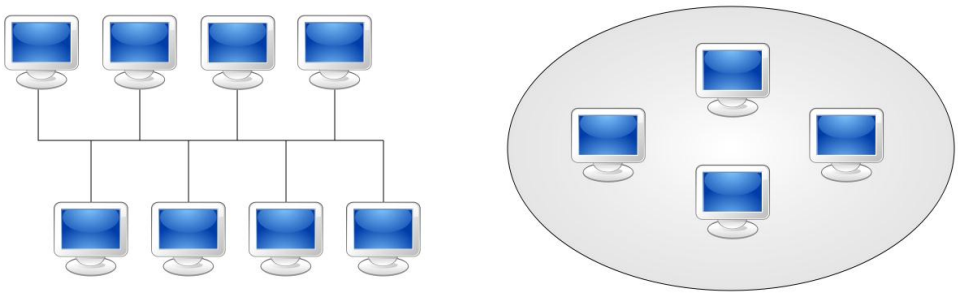

Abb. 2.2 *Topologien mit Point-to-Multipoint Verbindung: a) Bus, b) Zelle*

Linie

Als *Linie* (Abb. 2.3a) wird eine Point-to-Point basierende Topologie bezeichnet, in der ein einziger Pfad alle Knoten verbindet. Die zu übertragenden Daten werden dabei von Teilnehmer zu Teilnehmer weitergeleitet. Sie hat die geringste Komplexität unter den Point-to-Point Topologien und ist daher *aufwandseffizient*.

Ring

In einem *Ring* (Abb. 2.3b) ist jeder Knoten mit genau zwei anderen Verbunden. Diese Topologie entsteht zwingend aus der Forderung einer minimalen Anzahl von Verbindungen, so dass die Konnektivität 2 ist. Daher ist die Ring-Topologie *sicherheitseffizient*.

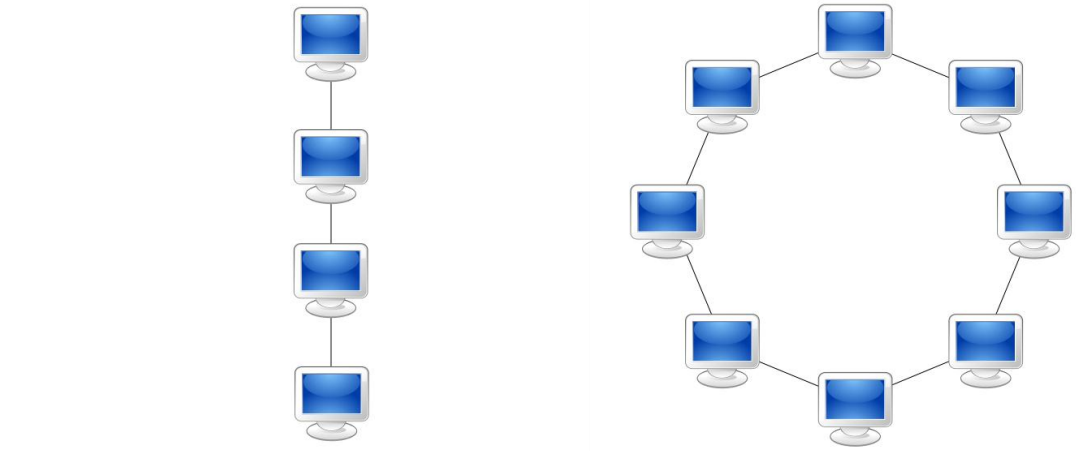

Abb. 2.3 *Topologien mit Point-to-Point Verbindung I*: *a) Linie, b) Ring*

Stern

Ein *Stern* (Abb. 2.4a) ist eine unsymmetrische Point-to-Point Topologie, da der zentrale Knoten (Verteiler) eine andere Klasse bildet als die mit ihm verbundenen Knoten. Die Stern-Topologie entsteht zwingend aus der Forderung einer minimalen Anzahl von Verbindungen bei der Bisektionsweite $n/2$ oder der Forderung einer minimalen Anzahl von Verbindungen bei einem Durchmesser von 2 und ist daher *leistungseffizient.*

Vermaschung

Die *Vermaschung* bezeichnet allgemein eine Point-to-Point basierende Topologie, bei der einzelnen Knoten mit mehr als einer Kante miteinander verbunden sind. Die Verbindung aller Knoten mit allen anderen wird als vollständige Vermaschung bzw. *Vollvermaschung* bezeichnet (Abb. 2.4b). Die Vollvermaschung ist zwar die aufwändigste (maximale Komplexität) Topologie, jedoch auch die sicherste (maximale Konnektivität), die schnellste (minimaler Durchmesser) und die leistungsstärkste (maximale Bisektionsweite).

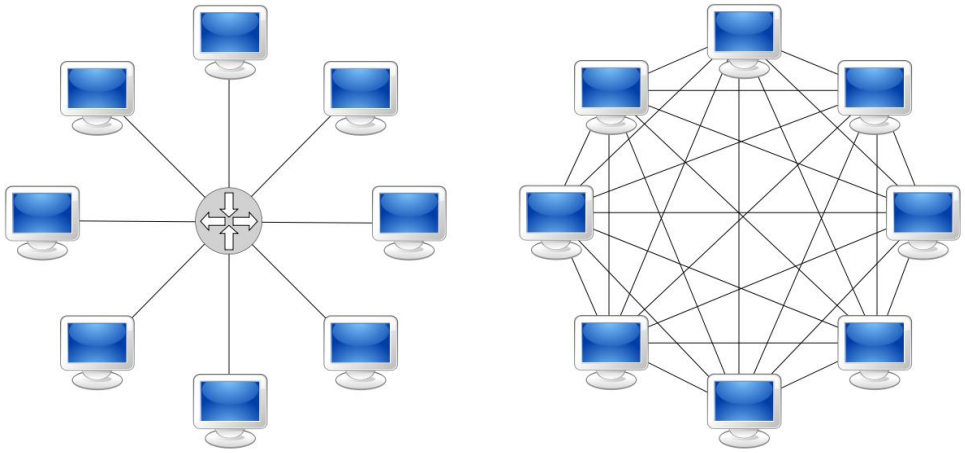

Abb. 2.4 *Topologien mit Point-to-Point Verbindung II: a) Stern, b) Vollvermaschung*

Vergleich elementarer Topologien

Topologie	Komplexität	Konnektivität	Durchmesser	Bisektionsweite
Bus	O(1)	1	1	1
Zelle	O(1)	1	1	1
Linie	O(n)	1	n-1	1
Ring	O(n)	2	n/2	2
Stern	O(n)	1	2	n/2
Vollständig	O(n^2)	n-1	1	n^2/4

Tbl. 2.2 *Vergleich elementarer Topologien*

2.1.5 Hierarchische Topologien

Elementare Topologien lassen sich nun als Grundbausteine zur Beschreibung beliebiger Netzwerke verwenden. Häufig haben diese einen hierarchischen Aufbau. In diesem Fall spricht man von einem *hierarchischen Netzwerk* bzw. einer *hierarchischen Topologie* des Netzwerkes.

Hierarchische Netzwerke zeichnen sich dadurch aus, dass aus der Sich eines bestimmten Teilnetzes immer zwischen übergeordneten Netzwerken und untergeordneten Netzwerken (insofern vorhanden) unterschieden werden kann. Dies ermöglicht es den Graphen aus der Sicht eines bestimmten Teilnetzes zu vereinfachen. Dabei wird je ein hierarchisch untergeordnetes Teilnetz durch einen einzelnen Knoten dargestellt. Umgekehrt kann aber auch das übergeordnete Netzwerk durch einen Knoten dargestellt wird

Nun ist es aber dennoch wünschenswert dem Graphen zumindest rein optisch zu entnehmen ob es sich bei einem Knoten um einen „normalen“ Knoten, um ein untergeordnetes Teilnetz oder um ein übergeordnetes Netzwerk handelt. Hierbei werden üblicherweise untergeordnete Teilnetzwerke als „Gruppe“ und übergeordnete Netzwerke als Wolke dargestellt, welche daher auch als *Cloud* (v. engl. „Wolke“) bezeichnet wird. Eine gerichtete Kante zu einem Knoten der ein untergeordnetes Netzwerk repräsentiert heißt dann *Downlink*, und eine gerichtete Kante zu einem Knoten, der ein übergeordnetes Netzwerk repräsentiert *Uplink*. Diese Bezeichnungen stammen ursprünglich aus der Satellitenkommunikation, bei der ein Uplink die Verbindung zu einem Satelliten und ein Downlink die Verbindung von einem Satelliten bezeichnet. Die Datenübertragung über einen Uplink wird als *Upstream* und über einen Downlink als *Downstream* bezeichnet.

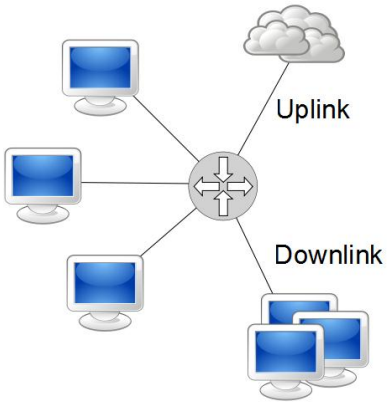

Abb. 2.5 *Uplink und Downlink in einem Teilnetz: Die Wolke (Cloud) repräsentiert das übergeordnete Netzwerk, die Gruppe ein Teilnetzwerk*

Zell-Topologie

Die *Zell-Topologie* ist eine hierarchische Topologie, bei dem die Teilnetze jeweils die Topologie einer Zelle besitzen.

Bei der Zell-Topologie erfolgt die Strukturierung der Zellen nach ihrer Reichweite (Tbl. 2.3). Diese Strukturierung findet im Mobilfunk Anwendung um einer Großzahl der Teilnehmer jeweils eine ähnliche Kanalkapazität zur Verfügung zu stellen. Hierbei besteht das Problem, dass sich die Teilnehmer innerhalb einer Zelle die Kanalkapazität teilen. Zur Lösung dieses Problems werden in Gebieten mit geringer Teilnehmerdichte Zellen mit großer Reichweite und in Ballungszentren sich überschneidende Zellen mit kleiner Reichweite eingesetzt.

Klasse	Beschreibung	Reichweite
Pico	*Picozellen* (v. lat.: pico, „klein“) sind „persönliche Funkzellen“ für Büros, Stockwerke oder Gebäude	< 50 m
Micro	*Mikrozellen* (v. gr.: mikrós, „klein“) umfassen Häuserblocks, Straßenzüge oder Stadtzentren	< 300 m
Macro	*Makrozellen* (v. gr.: makros, „groß“) umfassen ganze Stadtteile oder kleinere Regionen	< 20 km
World	*Weltweite Zellen* bezeichnen globale Funkzellen mit Satelliten-gestützen Verbindungen	-

Tbl. 2.3 *Strukturierung von Zellen nach Reichweite*

Baum-Topologie

Die *Baum-Topologie* ist eine hierarchische Topologie bei der die Teilnetze jeweils die Topologie eines Sternes besitzen. Diese kann anhand eines Baumes, dessen Äste sich immer weiter verzweigen gut veranschaulicht werden, weshalb hierbei auch von einem „organischen Aufbau“ gesprochen wird. In diesem Kontext erhält der Verteiler des hierarchisch höchsten Netzwerkes die Bezeichnung *Wurzel* und die Endknoten die Bezeichnung *Blätter*.

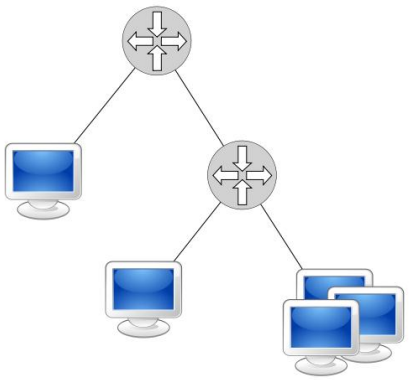

Abb. 2.6 *Baum-Topologie: Der obere Verteiler ist die Wurzel des Baumes, die einzelnen Hosts die Blätter.*

2.1.6 Dezentrale Netzwerke

Dezentrale Netzwerke bezeichnen beliebige Zusammenschlüsse von Teilnetzen. Insbesondere wird keine hierarchische Unterteilbarkeit zwischen diesen gefordert, so dass die Topologie eines dezentralen Netzwerkes keinem besonderen Schema unterliegen muss. Das besondere Interesse gilt hierbei nun der Vermaschung zwischen den einzelnen Teilnetzen. Die Intention dabei ist es trotz Ausfall einzelner Verbindungen eine Übertragung zu ermöglichen.

Häufig entsprechen dezentrale Netzwerke hierarchischen Netzwerken, bei denen zusätzliche Verbindungen zwischen den Teilnetzen bestehen. Je nachdem wie stark diese Vermaschungen ausgeprägt sind spricht man von einem hohen, oder von einem niedrigen *Zentralisierungsgrad*.

Strukturierung dezentraler Netzwerke

Aufgrund der komplexen Struktur von dezentralen Netzwerken und der schweren Trennbarkeit in Teilnetze ist es notwendig andere Kriterien zur Strukturierung anzuwenden. Eine einfache Möglichkeit bietet die Verwendung der Teilnetze (Tbl. 2.4).

Bemerkung *Neben den Klassen in Tbl. 2.4 werden zunehmend auch weitere Klassen, wie Body Area Networks (BAN), Controller Area Networks (CAN) oder Personal Area Networks (PAN) unterschieden. Des Weiteren wird auch häufig der Zusatz „Wireless“ verwendet, um diese Klassifizierung auf reine Funknetzwerke anzuwenden: WLAN, WMAN, WWAN, WBAN, WPAN.*

Klasse	Beschreibung
LAN	Lokale Netzwerke (engl.: *Local Area Networks*) umfassen zum Beispiel einzelne Büros, Gebäude oder Grundstücke und dienen der Vernetzung räumlich nicht getrennter Personengruppen (und Netzwerkperipherie). LANs werden vorwiegend leistungseffizient organisiert.
MAN	Regionale Netzwerke (engl.: *Metropolitan Area Networks*) bezeichnen Netzwerke in der Größenordnung eines größeren Unternehmens oder einer Stadt und dienen der Verbindung von einzelnen Zentren. Die Struktur ist meistens sicherheitseffizient.
WAN	Weitverkehrsnetze (engl.: *Wide Area Networks*) sind unbeschränkt hinsichtlich der geographischen Ausdehnung und vernetzen bis zu mehreren Kontinenten. Hierfür werden Strukturen mit hohen Datendurchsätzen (Leistung) und hoher Konnektivität (Sicherheit) verwendet.

Tbl. 2.4 *Strukturierung von Netzwerken nach Verwendung*

2.1.7 Struktur des Internet

Das Konzept dezentraler Netzwerke entstand in den frühen 60er Jahren und wurde 1969 im Vorläufer des heutigen Internet, dem sog. *ARPAnet* von einer Forschungsbehörde des US-Verteidigungsministeriums (ARPA, Advanced Research Projects Agency) umgesetzt. Mit dem Hintergrund des Kalten Krieges galt es ein Kommunikationsnetzwerk zu schaffen, welches möglichst sicher auf Ausfälle von Verbindungen reagieren sollte.

Organisatorische Struktur

Die Struktur des heutigen Internet entstand durch schrittweise Anbindung von Forschungsnetzwerken, militärischen Netzwerken, behördlichen und später auch Unternehmensnetzwerken an das ARPAnet. Die dabei entstandenen Verwaltungsbereiche werden als *autonome Systeme* bezeichnet. Die Registrierung autonomer Systeme und verschiedener anderer Internetressourcen (wie z.B. global eindeutiger „Netzwerkadressen“, siehe) unterliegt einer zentralen Organisation, der *IANA*. Diese delegiert wiederum *regionale Registraturen* (engl.: *Regional Internet Registries*, RIR) und diese schließlich *lokale Registraturen* (engl.: *Local Internet Registry*, LIR) mit der Verwaltung. Die IANA selbst ist eine Unterabteilung der ICANN, einer zentralen Organisation die zuweilen als „Internetregierung“ bezeichnet wird und viele zentrale (Verwaltungs-)Aufgaben in Bezug auf das Internet in sich vereint (Tbl. 2.5)

Internet Corporation for Assigned Names and Numbers (ICANN)	
Internet Assigned Numbers Authority (IANA)	
RIR	*African Network Information Centre* (AfriNIC)
	Asia Pacific Network Information Centre (APNIC)
	American Registry for Internet Numbers (ARIN)
	Latin American and Caribbean Network Information Centre (LACNIC)
	Réseaux IP Européens Network Coordination Centre (RIPE NCC)
LIR	*Internet Service Provider* (ISP), Behörden, Unternehmen, Universitäten

Tbl. 2.5 *Organisationsstruktur des Internet*

Jedem autonomen System wird eine eindeutige Nummer, die sog. *AS-Nummer (ASN)* zugewiesen. Ursprünglich war diese AS-Nummer eine 16-Bit lange Adresse, mit deren Hilfe somit 65536 autonome Systeme adressierbar waren, seit 2009 wurde die Länge allerdings auf 32-Bit erhöht. Dies ermöglicht es autonome Systeme bei Bedarf in kleinere Teilbereiche, sog. *Areas* (v. engl. area, „Bereich“) zu strukturieren.

Topologische Struktur

Da sich die Teilung in autonome Systeme in der Topologie des Internet wiederspiegelt, besitzt dieses einen hohen Zentralisierungsgrad, wobei jedoch zu beachten ist, dass einzelne Teilnetzwerke durchaus stark vermascht sein können. Die auf der Zentralisierung beruhende hierarchische Struktur erlaubt eine eher topologisch orientierte Strukturierung des Internet:

- Netzwerke die autonome System miteinander verbinden ordnet man dem sogenannten *Zentralbereich* zu. Dieser Bereich ist zwar auf einen hohen Datendurchsatz und auf eine hohe Ausfallssicherheit angewiesen, jedoch erstrecken sich die dabei auftretenden Verbindungen über sehr große Distanzen, weshalb hier vorwiegend kosteneffiziente Topologien, wie die Linien-Topologie eingesetzt werden. Aufgrund dieser Struktur wird der Zentralbereich häufig auch als *Backbone* (engl. f. Rückgrat) bezeichnet.

- In autonomen Systemen werden üblicherweise sicherheitseffiziente Topologien wie die Ring-Topologie zur Absicherung gegen Ausfälle der Infrastruktur eingesetzt.

- Netzwerke welche die Endknoten mit dem Internet verbinden werden dem sogenannten *Zugangsbereich* zugeordnet und üblicherweise durch leistungseffiziente Topologien wie Sterne realisiert, da hier Ausfälle meist nur geringe Auswirkungen haben.

2.2 Routing

2.2.1 Netzwerk Metrik

Im Folgenden wollen wir uns mit dem Problem der „Pfadsuche“ in Netzwerken beschäftigen. Insbesondere wollen wir hierbei einen möglichst guten Pfad für eine Datenübertragung finden. Nun spielen hierbei allerdings sehr viele Faktoren eine Rolle. Neben der Topologie des Netzwerkes sind dies z.B. auch die verschiedenen Eigenschaften der Übertragungskanäle. So kann es für eine Datenübertragung günstiger sein einen Umweg in Kauf zu nehmen, wenn dadurch ein Flaschenhals, z.B. ein Übertragungskanal mit niedriger Kanalkapazität umgangen wird. Hierfür ist es zunächst jedoch erforderlich, die Informationen über die Kanäle auf Graphen zu übertragen.

Gewichtung

Die Zuordnung von Informationen über die einzelnen Kanten eines Graphen wird als *Gewichtung* bezeichnet. Dabei erhält jede Kante e des Graphen jeweils einen ihr zugeordneten Wert $w(e)$, der die gewünschten Informationen enthält. Diese Werte werden als *Kantengewichte* bezeichnet. Dabei ist zu beachten, dass die Gewichtung im Allgemeinen richtungsabhängig ist, also verschiedene Werte für verschiedene Richtungen beinhalten kann. Des Weiteren kann $w(e)$ auch zusammengesetzte Informationen also z.B. sowohl über Kanalkapazität, als auch Latenzzeit beinhalten. Ein Graph G, der eine Gewichtung besitzt heißt *gewichteter Graph*. Im Weiteren werden wir für die Gewichte der Kanten e_1, e_2, e_3, etc. die abkürzende Schreibweise w_1, w_2, w_3, etc. verwenden.

Nun stellt sich die Frage inwiefern sich die Kantengewichte eines Graphen auf Pfade fortsetzen lassen. Entspricht die Gewichtungen zum Beispiel der (Pfad-) Länge dann ist die Länge eines Pfades e_1, e_2, e_3, etc. einfach die Summe der einzelnen Längen. In diesem Fall lässt sich die Gewichtung des Pfades mit $w_{1,2,3,\dots} = w_1 + w_2 + w_3 + \dots$ bestimmen. Andererseits kann die Gewichtung aber auch auf den Kanalkapazitäten, den Latenzzeiten oder ganz anderen Informationen beruhen. Auch in diesen Fällen wollen wir aber dass sich die Kantengewichte in natürlicher Form auf die Gewichtung von Pfaden übertragen. Damit dies möglich ist weicht die Rechenvorschrift jedoch u.U. von der obigen ab. Daher ist es notwendig neben der Gewichtung der einzelnen Kanten auch die Vorschrift zur Berechnung der Gewichtungen der Pfade zu kennen.

Netzwerk Metrik

Die Gewichtung eines Graphen (also der einzelnen Kanten) zusammen mit der zugehörigen Vorschrift zur Berechnung der Gewichtungen der Pfade wird als „Netzwerk Metrik“ bezeichnet.

Def. 2.4 (Netzwerk Metrik) *Die Metrik eines Netzwerkes bezeichnet ein Maß zur Bewertung von Kanten und Pfaden. Dabei ist die Bewertung der Pfade durch eine feste Rechenvorschrift mit den Bewertungen der einzelnen Kanten des Pfades verknüpft.*

Metriken lassen sich aufgrund der Rechenvorschrift in verschieden Klassen unterteilen. Einige wichtige Klassen von Metriken finden sich in Tbl. 2.6.

Klasse	Rechenvorschrift	Beispiele
Additive Metriken	$w_{1,2,3,\ldots} = w_1 + w_2 + w_3 + \ldots$	Latenzzeit, (Pfad-)Länge
Multiplikative Metriken	$w_{1,2,3,\ldots} = w_1 \cdot w_2 \cdot w_3 \cdot \ldots$	Ausfallswahrscheinlichkeit
Konkave Metriken	$w_{1,2,3,\ldots} = \min(w_1, w_2, w_3, \ldots)$	Kanalkapazität
Konvexe Metriken	$w_{1,2,3,\ldots} = \max(w_1, w_2, w_3, \ldots)$	Kostenmetrik der Kanalkapazität (Bsp. 2.3)

Tbl. 2.6 *Klassifizierung von Netzwerk Metriken*

Kosten

Häufig beruht das Interesse an der Metrik eines bestimmten Netzwerkes auf der Fragestellung, wie „gut“ ein bestimmter Pfad gegenüber einem anderen ist (z.B. bei der Datenübertragung). Ohne weitere Überlegungen ist es jedoch nicht möglich verschiedene Pfade miteinander zu vergleichen, da a priori nicht klar ist was unter „gut“ zu verstehen ist. So kann z.B. ein höherer Wert bei der Latenzzeit als negativ und ein höherer Wert Kanalkapazität als positiv interpretiert werden.

Um nun Pfade (und somit Übertragungswege) miteinander vergleichen zu können, muss zunächst irgendeine Form der Vereinheitlichung stattfinden. Diese erfolgt beim Übergang von einer gegebenen Gewichtung zu den sog. *Kosten*, bzw. einer gegebenen Metrik zu einer *Kostenmetrik*. Die Idee dabei ist, dass hohe Kosten grundsätzlich immer schlechter zu bewerten sind als geringe. Eine höhere Latenzzeit entspricht also höheren Kosten, andererseits entspricht eine höhere Kanalkapazität geringeren Kosten. Der Übergang erfolgt in zwei Schritten:

1. **Ermittlung der Kosten für einzelne Kanten**

   In diesem Schritt soll eine Funktion c gefunden werden, die jeder Kante aufgrund ihrer Gewichtung bestimmte Kosten zuordnet. Dabei soll c so gewählt werden, dass in Bezug auf die Gewichtung „bessere“ Kanten niedrigere Kosten verursachen. Um eine solche Funktion zu finden ist es hilfreich einige Überlegungen durchzuführen:

   Zunächst stellt sich die Frage was bei den Kosten unter „1“ zu verstehen ist. Hierfür wird eine Kante e_S mit der Gewichtung w_S als Maßstab ausgewählt. Für die Funktion c muss also gelten: $c(e_S) = 1$.

Als nächstes kann man sich die Frage stellen wie „gut“ andere Gewichtungen w gegenüber der Gewichtung w_S abschneiden, also welche Gewichtung ist zum Beispiel doppelt so „schlecht“ wie die von e_S. Für eine Kante e mit dieser Gewichtung muss dann ja gelten: $c(e) = 2$, da die Kosten von e somit doppelt sich hoch sein müssen, wie die von e_S.

2. Ermittlung der Kosten für Pfade und der Kostenmetrik

Nun verfügen wir über eine Funktion c, welche bei Kanten aufgrund des Gewichtes die Kosten ermittelt. Um nun die Kosten eines Pfades zu ermitteln reicht es aus mit Hilfe der bestehenden Metrik die Gewichtung des Pfades auszurechnen und in c einzusetzen.

Um die Kostenmetrik zu bestimmen, muss die ursprüngliche Metrik in c so eingesetzt werden, dass die Kosten eines Pfades in Abhängigkeit der Kosten der einzelnen Kanten vorliegen.

Im Allgemeinen kann es sehr schwierig sein eine geeignete Funktion c zur Bestimmung der Kosten auszuwählen. In vielen Fällen jedoch kann diese auch naheliegend sein, wie in den folgenden Beispielen Bsp. 2.1 bis Bsp. 2.3 zu sehen ist:

Bsp. 2.1 (Hop-Count) *Die Gewichtung mittels der (Pfad-)Länge wird als Hop-Count (engl. „Hop“: Hopser) bezeichnet. Als Maßstab wird zunächst eine einzelne Kante mit der Länge 1 gewählt. Da aber jede Kante per Definition die Länge 1 besitzt, sind die Kosten jeder einzelnen Kante gleich und somit 1. Nun ist eine Länge von 1 doppelt so „gut“ wie eine Pfadlänge von 2. Also versursacht ein Pfad aus zwei Kanten die Kosten 2 etc. Wenn man dies fortführt mit beliebigen Anzahlen von Kanten, so erkennt man, dass die Kosten eines* Pfade*s immer gleich der Anzahl der Kanten sind. Für die Kosten einer Pfades gilt also c = w, wobei w die Länge des Pfades ist. Für die Kostenmetrik gilt somit:*

$$c_{1,2,3,\ \ldots} = c_1 + c_2 + c_3 + \ldots$$

Bsp. 2.2 (Latenzzeit) *Wird die Latenzzeit zur Gewichtung verwendet, so ergibt sich ein ähnlicher Fall wie beim Hop-Count. Der Unterschied ist jedoch, dass zunächst ein Maßstab für die Latenzzeit gewählt werden muss. Um einen Maßstab zu finden könnte man beispielsweise die Kante mit der geringsten Latenzzeit wählen. Wurde ein Maßstab mit der Latenzzeit* w_S *festgelegt, so gilt für die Kosten eines Pfades* $c = w / w_S$*, wobei w die Latenzzeit des Pfades bezeichnet. Diese errechnet sich aus der Summe der Latenzzeiten der einzelnen Kanten des Pfades. Somit gilt:*

$$c_{1,2,3,\ \ldots} = c_1 + c_2 + c_3 + \ldots$$

Bsp. 2.3 (Kanalkapazität) *Bei der Kanalkapazität ergibt sich ein anderer Fall. Zunächst wollen wir uns aber wieder eine bestimmte Kanalkapazität w_S als Maßstab wählen. Wieder wäre hier eine mögliche Vorgehensweise die „beste" Kante, also die mit der höchsten Kanalkapazität zu wählen. Wurde nun ein w_S bestimmt, so stellt sich die Frage, welche Kanalkapazität den doppelten Kosten entspricht. Dies wäre ein Kanal mit der halben Kapazität. Insgesamt folgt: $c = w_S / w$. Dabei bezeichnet w die Kanalkapazität eines Pfades, also $w = min(w_1, w_2, w_3, \ldots)$. Somit gilt:*

$$c_{1,2,3,\ \ldots} = w_S / min(w_1, w_2, w_3, \ldots) \Rightarrow c_{1,2,3,\ \ldots} = max\ (c_1, c_2, c_3, \ldots)$$

*** Kostenmatrix**

Nun können beliebigen verbundenen Knoten aufgrund der sie verbindenden Pfade Kosten zugeordnet werden. Sind mehrere Pfade zwischen diesen vorhanden, dann wird der Pfad mit den geringsten Kosten ausgewählt: Umso geringer die Kosten, desto besser! Je nach der zugrunde liegenden Gewichtung kann es auch vorkommen, dass die Kosten c_{ij} (von Knoten v_i nach v_j) nicht die gleichen Kosten wie in die Gegenrichtung, also c_{ji} sind. Schließlich werden bei nicht verbundenen Knoten v_i und v_j die Kosten auf unendlich gesetzt: $c_{ij} = \infty$. Und die Kosten von einem Knoten zu sich selbst auf 0. Auf diese Weise können wir zwischen beliebigen Knoten im Netzwerk Kosten angeben.

In einem Netzwerk mit *n* Knoten lassen sich somit alle Kosten zwischen den Knoten durch eine $n \times n$-Matrix deren Eintrag in der *i*-ten Zeile und der *j*-ten Spalte den Wert c_{ij} hat darstellen. Diese Matrix wird als *Kostenmatrix* bezeichnet. Diese wird zur Erstellung sog. „Routing-Tabellen" verwendet.

2.2.2 Ermittlung kürzester Pfade

In Abschnitt 2.1.1 wurde ein „kürzester Pfad" als einen Pfad mit minimaler Anzahl von Kanten definiert. Dies entspricht in der Hop-Count Metrik einem Pfad, der minimale Kosten verursacht. Nun wollen wir aber den Begriff aber auf eine beliebige Metrik anwenden, und definieren ganz allgemein einen *kürzesten Pfad* als einen Pfad mit minimalen Kosten.

Bisher haben wir die Fragestellung, wie ein solcher kürzester Pfad zu finden ist völlig ausgeklammert. Wir sind davon ausgegangen, dass schlichtweg alle Pfade bekannt sind und unter diesen der beste mit den minimalen Kosten ausgewählt wird. Das diese Strategie schnell zu einem Problem werden kann ist jedoch schon bei relativ einfachen Topologien einzusehen, wenn es eine Unzahl von möglichen Pfaden zwischen verschiedenen Knoten gibt, welche miteinander verglichen werden müssten (Abb. 2.7)

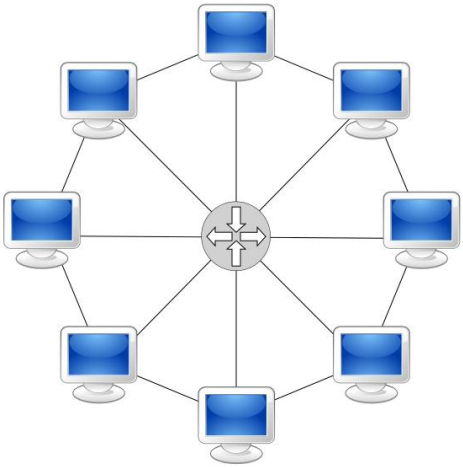

Abb. 2.7 *Stern-Ring Topologie: Bereits bei einfachen Topologien können sich zwischen zwei Knoten eine Unzahl an möglichen Pfaden ergeben. Daher ist eine Strategie zur Ermittlung kürzester Pfade erforderlich.*

Zur Ermittlung kürzester Pfade werden sog. *kürzester-Pfad-Algorithmen* (engl.: *Shortest Path*) eingesetzt. Diese können eingesetzt werden, um von einem Startknoten aus zu einem bestimmten Zielknoten den (bzw. einen) kürzesten Pfad zu finden (engl.: *Single-Pair Shortest Path*), oder aber auch um von einem Startknoten aus zu allen weiteren Knoten im Netzwerk kürzeste Pfade zu finden (engl.: *Single-Source Shortest Path*). Wichtige Vertreter von Kürzester-Pfad-Algorithmen sind der „Dijkstra-Algorithmus“ und der „Bellman-Ford-Algorithmus“.

Dijkstra-Algorithmus

Der sog. *Dijkstra-Algorithmus* (auch *Shortest Path First*) entspringt der Idee von einem Startknoten aus den nächsten Knoten (mit den geringsten Kosten) auszuwählen. Anschließen wird der Knoten, der den obigen zwei am nächsten steht ausgewählt und immer so weiter. Eine Voraussetzung zur Anwendung dieses Algorithmus ist, dass Kantengewichte nur positive Werte annehmen dürfen. Zudem müssen zusammengesetzte Pfade höhere Kosten besitzen, als die einzelnen Kanten. Diese Voraussetzungen werden beispielsweise durch die Metriken in 2.2.1erfüllt.

Die Umsetzung des Algorithmus bedient sich zweier Listen: Eine Liste von Knoten deren kürzester Pfad bekannt ist (bekannte Knoten) und eine Liste für die der kürzeste Pfad noch unbekannt ist (unbekannte Knoten). Zunächst befindet sich nur der Startknoten in der Liste der bekannten Knoten. Von diesem aus wird der benachbarte Knoten mit den minimalen Kosten ausgewählt. Für diesen ist der kürzeste Pfad dann offensichtlich die direkte Verbindung, da jeder Umweg über einen anderen Knoten aufgrund der Voraussetzung mehr Kosten verursachen würde. Somit kann dieser Knoten in die Liste der bekannten Knoten aufgenommen werden.

Bei jedem weiteren Schritt wird diese Verfahrensweise sukzessive fortgesetzt. Dabei werden jeweils die Kosten zu allen unbekannten Knoten die zu irgendeinem bekannten Knoten benachbart sind ermittelt. Dies ist sehr einfach, da zur Berechnung nur die Kosten jeweilig des bekannten kürzesten Pfades (zum bekannten Knoten) mit den Kosten der Kante zu dem unbekannten Knoten entsprechend der Berechnungsvorschrift zusammengesetzt werden muss. Falls ein solcher unbekannter Knoten zu mehreren bekannten Knoten benachbart ist, muss überprüft werden über welchen bekannten Knoten die geringeren Kosten anfallen.

Anschließend kann der unbekannte Knoten in die Liste der bekannten aufgenommen werden.

Da sich bei jedem Schritt die Liste der unbekannten Knoten um einen Knoten verringert, sind bei einer Gesamtzahl von n Knoten also $n - 1$ Schritte erforderlich um einen kürzesten Pfad zu einem bestimmten Zielknoten zu bestimmen. Sollen andererseits zu allen Knoten kürzeste Pfade ermittelt werden, sind genau $n - 1$ Schritte erforderlich. Eine wichtige Bedeutung hat der Dijkstra-Algorithmus für das Internet, da er aufgrund der niedrigen Anzahl notwendiger Schritte die Grundlage des Routing innerhalb autonomer Systeme bildet. (Abschnitt 2.2.5)

Bellman-Ford-Algorithmus

Der Bellman-Ford-Algorithmus wird verwendet, falls negative Kosten auftreten können. Dadurch, dass die Einschränkung auf positive Werte wegfällt, wird eine aufwändigere Berechnung der kürzesten Pfade erforderlich. Der Bellman-Ford-Algorithmus hat eine wichtige Bedeutung für das Routing zwischen autonomen Systemen (siehe 2.2.5).

2.2.3 Vermittlung und Routing

Sind Knoten über mehrere Pfade miteinander verbunden, so wir die Aufgabe aus diesen solche auszuwählen die für eine bestimmte Datenübertragung verwendet werden, als *Routing* bezeichnet. Die Knoten welche bei der Datenübertragung die Weiterleitung übernehmen heißen *Router* oder *Vermittlungsstellen*.

Grundsätzlich lassen sich Strategien zur Vermittlung aufgrund der Art des Datenaustausches zwischen den Endknoten klassifizieren:

- Ein *verbindungsorientierter Datenaustausch* beruht auf einer sog. *Verbindung* zwischen den Endknoten. Diese beruht auf drei Phasen: Den *Verbindungsaufbau*, den Datenaustausch und den *Verbindungsabbau*. Dabei besteht die Verbindung in der Zeit zwischen Verbindungsaufbau und Verbindungsabbau unabhängig davon, wann und ob Daten übertragen werden.

- Bei einem *verbindungslosen Datenaustausch* werden Daten ohne vorhergehende Verbindungsphase ausgetauscht.

Um diese Arten des Datenaustausches zu ermöglichen werden verschiedene Ansätze verwendet: Dies sind die *Leitungsvermittlung* und die *Paketvermittlung*:

Leitungsvermittlung

Die *Leitungsvermittlung* realisiert einen verbindungsorientierten Datenaustausch, bei dem den Teilnehmern über die gesamte Dauer der Verbindung ein fester Pfad (z.B. ein zuvor ermittelter kürzester Pfad) und die entsprechenden Übertragungskanäle exklusiv zur Verfügung gestellt werden. Dieses ermöglicht einerseits eine konstante Kanalkapazität und andererseits eine konstante Latenzzeit, da die Vermittlungsstellen alle benötigten Informationen bereits während dem Verbindungsaufbau erhalten und somit eingehende Daten direkt weiterleiten können. Tritt jedoch während der Übertragung ein Fehler auf, so gehen die Daten verloren und es muss eine neue Verbindung aufgebaut werden.

Paketvermittlung

Die *Paketvermittlung*. wurde in der Zeit des „Kalten Krieges“ mit dem Ziel entwickelt einen Datenaustausch zwischen beliebigen Knoten zu ermöglichen, der robust gegenüber Teilausfälle der Netz-Infrastruktur ist. Hierfür werden die Daten vom Sender in viele kleine Pakete aufgeteilt und über ein dezentrales Netzwerk an den Empfänger übertragen, welcher diese wieder zusammensetzt. Damit die Pakete unabhängig voneinander im Netzwerk übertragen werden können enthalten sie neben den zu übertragenden Daten auch Adressinformationen über den Empfänger und damit der Empfänger weiß woher die Pakete stammen auch vom Sender. Dabei entscheidet jede Vermittlungsstelle selbstständig wohin die Daten weitergeleitet werden sollen. Fällt im Netzwerk nun eine Verbindung aus, so kann dies von den betroffenen Vermittlungsstellen erkannt und bei der Übertragung berücksichtigt werden.

In seiner natürlichen Form realisiert die Paketvermittlung einen verbindungslosen Datenaustausch. Da verschiedene Anwendungen jedoch einen verbindungsorientierten Datenaustausch erfordern werden auch Mechanismen verwendet, die auf der Paketvermittlung aufbauend eine virtuelle Verbindung zwischen den Endknoten schaffen. In diesem Fall spricht man von einer *verbindungsorientierten Paketvermittlung*.

2.2.4 Routing Strategien

Wie sollen die Vermittlungsstellen nun aber entscheiden, an welchen benachbarten Knoten im Netzwerk eingehende Daten weiterzuleiten sind? Zur Lösung dieses Problems werden auf diesen alle benötigten Informationen in Form sog. *Routing-Tabellen* hinterlegt. Diese enthalten jeweils zu jedem Zielknoten Information darüber, über welche Kanten Pakete weitergeleitet werden können. Falls es mehrere Möglichkeiten der Weiterleitung gibt enthalten sie zudem die Kosten der zugehörigen kürzesten Pfade zum Zielknoten. Auf diese Weise kann der Router den kürzeren Pfad (mit den geringeren Kosten) wählen.

Nun stellt sich aber die Frage, woher die Router die Routing-Tabellen erhalten? In kleineren Netzwerken können diese Einträge evtl. manuell erzeugt werden, jedoch in größeren wäre dieser Ansatz nicht mehr durchführbar. Des Weiteren stellt sich noch die Frage was geschieht, wenn sich etwas am Netzwerk ändert, da in diesem Fall die Einträge nicht mehr stimmen müssen ... Um die verschiedenen Aspekte und dabei insbesondere auch den Aufwand zu berücksichtigen werden verschiedene Routing Strategien verfolgt. Man unterteilt diese in „statisches Routing“, „alternatives Routing“ und „adaptives Routing“:

Statisches Routing

Aufgrund der einfachen Handhabung werden vorwiegend in kleineren Netzwerken die Einträge der Routing-Tabellen einmalig, meist manuell erzeugt. Diese Strategie wird als *statisches Routing* bezeichnet. Gibt es nun für ein Ziel mehrere unabhängige Pfade, beziehungsweise Möglichkeiten der Weiterleitung, so entscheidet der Router per Zufall welchen er wählen soll. Die zufällige Entscheidung erfolgt dabei so, dass Weiterleitungen die zu einem kürzeren Pfad führen mit einer höheren Wahrscheinlichkeit ausgewählt werden als andere. Ein Nachteil hierbei ist, dass Ausfälle der Netz-Infrastruktur (Knoten oder Kanten) nicht berücksichtigt werden und zu einer Erhöhung der Paketverluste führen.

Alternatives Routing

Um Fehler im Netzwerk zu berücksichtigen werden beim *alternativen Routing* vor der jeweiligen Datenübertragung alle möglichen Weiterleitungen überprüft. Hierfür wird anhand der Kosten eine feste Reihenfolge der Weiterleitungen definiert, die nacheinander geprüft werden. Als erstes wird die Weiterleitung mit den niedrigsten Kosten, der sogenannte *Erst-Pfad* auf ihre Benutzbarkeit überprüft. Sollte diese Weiterleitung nicht benutzbar sein, so wird der *Zweit-Pfad* geprüft usw. bis hin zu einem *Letzt-Pfad*. Sollte auch diese Weiterleitung nicht möglich sein, so gehen die Datenpakete verloren. Insgesamt können aber Fehler erkannt werden, wodurch sich die Stabilität der Datenübertragung erhöht. Alternatives Routing führt bei Ausfällen bzw. Veränderungen im Netzwerk dazu, dass die einzelnen Router veraltete Informationen haben und dadurch evtl. keine kürzesten Pfade mehr, sondern Umwege verwendet werden. Ausfälle der Netz-Infrastruktur führen also zu einer (evtl.) deutlichen Erhöhung der Übertragungszeit.

Adaptives Routing

Adaptives Routing bezeichnet nun Verfahren bei denen mit Hilfe von sog. *Routingprotokollen* die Routing-Tabellen der einzelnen Knoten immer wieder aktualisiert werden, so dass auch bei Ausfällen kürzeste Pfade verwendet werden. Dabei tauschen die Router mittels der Routingprotokolle Informationen über diese Pfade im Netzwerk untereinander aus. Durch diese Kommunikation untereinander (sog. *horizontale Kommunikation*, Abschnitt 3.1.2) werden viele verschiedene Strategien ermöglicht, um flexibel auf Netzwerkausfälle zu reagieren und den Datenfluss gleichmäßig im Netzwerk zu verteilen:

- **Isoliertes Routing**

  Beim isolierten Routing findet kein Austausch von Informationen statt, so dass jeder Knoten seine Routingtabellen nur aufgrund der selbst gewonnenen Informationen erstellt

- **Zentrales Routing**

  Beim zentralen Routing übernehmen einzelne zentrale Router, sogenannte *Routing Control Center* (RCC) die Aufgabe kürzeste Pfade zwischen den Knoten zu finden, sowie die Routingtabellen der einzelnen Router zu erstellen und an diese zu übermitteln.

- **Verteiltes adaptives Routing**

  Das verteilte Adaptive Routing liegt zwischen dem isolierten und dem zentralen Routing. Dabei werden die Informationen über die Kosten zu den Nachbarknoten von jedem Knoten gesammelt und alle bekannten Informationen an die Nachbarn weitergegeben. Im Laufe der Zeit erhält somit jeder einzelne Knoten vollständige Information über das gesamte Netzwerk und kann selbstständig optimale Pfade ermitteln.

- **Hierarchisches Routing**

  Bei größeren Netzwerken ergibt sich beim verteilten adaptiven Routing das Problem, dass Informationen zwischen entfernten Knoten nur sehr langsam ausgetauscht

werden. Dieses Problem wird beim hierarchischen Routing dadurch gelöst, dass der Austausch der Routinginformationen die logische Topologie eines hierarchischen Netzwerkes abbildet. Das heißt, dass die einzelnen RCCs Informationen sammeln und diese an übergeordnete RCCs weitergeben. Die übergeordneten RCCs sammeln auf diese Weise sehr schnell Informationen über größere Teile des Netzwerkes und können diese wiederum an die untergeordneten RCCs zurückgeben, bzw. auch an noch höher stehende RCCs weiterleiten etc. Welche Informationen dabei zwischen den Knoten ausgetauscht werden, wird wiederum von den jeweilig eingesetzten Routingprotokollen festgelegt.

2.2.5 Routing im Internet

Das Routing im Internet ist ein hierarchisches Routing und ist im Wesentlichen durch die topologische Strukturierung in Zentralbereich und die einzelnen autonomen Systeme bestimmt. Dabei wird zwischen Intradomain-Routing und Interdomain-Routing unterschieden:

Intradomain-Routing

Intradomain-Routing bezeichnet das Routing innerhalb eines autonomen Systems und wird daher in den meisten Fällen so umgesetzt, dass die Datenlast gleichmäßig auf die Netzwerkkapazitäten verteilt wird, so dass die Gesamtmenge der übertragenen Daten maximiert wird. Die dabei eingesetzten Routingprotokolle werden als *Interior Gateway-Protokolle* (IGP) bezeichnet und verwenden meistens den Dijkstra-Algorithmus, um kürzeste Pfade zu ermitteln. Die Optimierung erfolgt also hinsichtlich der Gesamtdatenmenge.

Interdomain-Routing

Das *Interdomain-Routing* hingegen bezeichnet das Routing im Zentralbereich, also zwischen den autonomen Systemen. Hierbei muss nun folgende Überlegung berücksichtigt werden: Die Nutzung eines Netzwerkes verursacht Kosten. Da die verschiedenen autonomen Systeme verschiedene Betreiber haben ist es verständlich dass ein Betreiber, der Daten zwischen anderen autonomen Systemen transportiert dafür entlohnt werden will. Diese Entlohnung wird gemäß der freien Marktwirtschaft vertraglich ausgehandelt. So kann es sein dass bestimmte Daten die übertragen werden einen Gewinn und andere einen Verlust verursachen. Dieses wird bei der Ermittlung der Übertragung-Pfade berücksichtigt, indem die Bildung der Kosten aufgrund der tatsächlichen wirtschaftlichen Kosten erfolgt. Dabei entsprechen negative Kosten also den Gewinnen. Aufgrund der negativen Kosten wird zur Ermittlung kürzester Pfade üblicherweise der Bellman-Ford-Algorithmus verwendet. Dabei erfolgt die Optimierung hinsichtlich der Gesamtkosten. Die Routingprotokolle beim Interdomain-Routing werden als *Exterior-Gateway-Protokolle (EGP)* bezeichnet.

Bei der vertraglichen Aushandlung nehmen die Netzbetreiber, welche die Sender oder die Empfänger bei der Datenübertragung stellen die Rolle der *Netzbenutzer* ein. Die Netzbetreiber, welche den Netzbenutzern ihr Netzwerk zur Übertragung zur Verfügung stellen entsprechen den *Netzanbietern*. Vertraglich gleichgestellte autonome Systeme werden hierbei als *Peers*, sowie die Partnerschaft zum Transport der Daten von Teilnehmern zwischen diesen als *Peering* bezeichnet. Der Datenaustausch zwischen diesen Netzwerken erfolgt dann in der Regel ohne gegenseitige Zahlungsansprüche.

*** Netzneutralität**

Bei Peerings zwischen Netzwerken im Zugangsbereich und im Zentralbereich kann ein wirtschaftliches Ungleichgewicht zwischen den Betreibern der jeweiligen Netzwerke entstehen. Dies wird dadurch verursacht, dass zentralere Netzanbieter häufig aufgrund einer höheren Kanalkapazität von Dienstanbietern mit einem hohen Datenaufkommen (wie z.B. Youtube etc.) bevorzugt werden. Durch diese Dienste wiederum erhöht sich sowohl beim zentralen Netzanbieter, als auch beim Netzwerk im Zugangsbereich das Datenaufkommen, so dass die Netzwerke „nachgerüstet" werden müssen. Um nun das erwähnte wirtschaftliche Ungleichgewicht einzusehen müssen nur die Topologien der Netzwerke verglichen werden. Das zentralere Netzwerk kann günstiger nachgerüstet werden als das Netzwerk im Zugangsbereich (siehe 2.1.7).

Dieses wirtschaftliche Ungleichgewicht hat nun dazu geführt, dass verschiedene Netzbetreiber im Zugangsbereich dazu übergegangen sind, die zu übertragenden Datenpakete zu analysieren (*Deep Packet Inspection*) und bei bestimmten Inhalten wie Videostreams oder bei solchen von bestimmten Dienstanbietern künstliche Beschränkungen der Kanalkapazität vorzunehmen. *Netzneutralität* bezeichnet nun ein Dogma, welches vorsieht, dass alle Pakete im Internet wertneutral und gleichgestellt übertragen werden sollen.

2.3 Netzwerk Kommunikation

2.3.1 Verbindungstypen

Bei der Übertragung von Daten unterscheidet man zwischen verschiedenen Verbindungstypen (Abb. 2.8):

Unicast

Ein *Unicast* bzw. Direktverbindung oder Ende-zu-Ende-Verbindung bezeichnet die Nachrichtenübertragung bei einer Kommunikation, bei der genau zwei Kommunikationspartner beteiligt sind.

Multicast

Ein *Multicast* bezeichnet eine Datenübertragung von einem Punkt zu einer Gruppe ausgewählter Teilnehmer. Dabei müssen Daten nur einmalig übertragen werden, wodurch sich ein technischer Vorteil gegenüber vielen einzelnen Unicast Übertragungen ergibt. Bei einer paketvermittelte Datenübertragung findet die Vervielfältigung der Pakete an jedem Knoten auf der Route statt.

Broadcast

Ein *Broadcast* bezeichnet in einem Teilnetzwerk die Übertragung einer Nachricht von einem Host an alle weiteren Teilnehmer. Dieses Teilnetzwerk ist ein logischer Verbund von Hosts innerhalb eines anderen Netzwerkes und wird als *Broadcast-Domäne* (v. lat.: dominium, „Herrschaft, Herrschaftsbereich") bezeichnet. In der paketvermittelten Datenübertragung werden Broadcasts vorwiegend bei unbekannten Empfängeradressen verwendet. Ebenso wie bei der Multicast Übertragung müssen Nachrichten dabei nur einmalig übertragen werden.

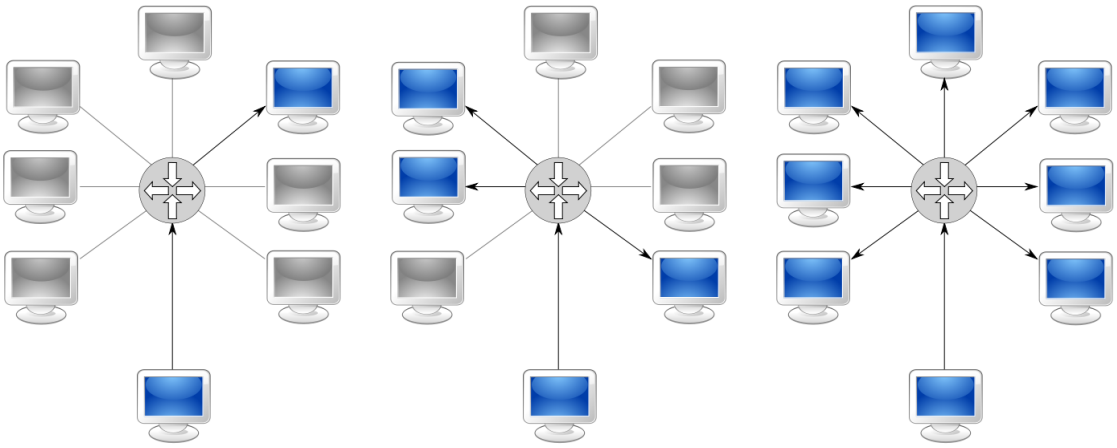

Abb. 2.8 *Verbindungstypen: a) Unicast, b) Multicast, c) Broadcast*

Anycast

Anycast bezeichnet eine Kommunikationsform bei der ein Gruppe von Kommunikationspartnern zusammengefasst wird und nach außen hin wie ein einziger Knoten erscheint. Dabei sind die einzelnen Hosts der Anycast-Gruppe über ein Netzwerk verteilt. Bei einer Kommunikationsanfrage eines Hosts mit der Anycast-Gruppe (welche sich als einzelner Knoten ausgibt) übernimmt ein Knoten der Gruppe die Kommunikation. Normalerweise ist dies der Knoten, der dem anfragenden Kommunikationspartner am nächsten ist (Abb. 2.9). Aus der Sicht des Anfragenden Kommunikationspartners entspricht die Anycast Verbindung dann einer Unicast Verbindung. Auf diese Weise können bestimmte Dienste flächendeckend in einem Netzwerk bereitgestellt werden.

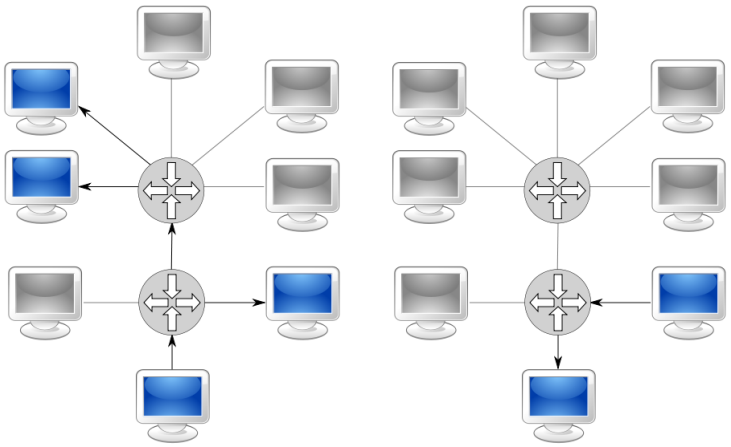

Abb. 2.9 *Anycast: a) Request, b) Response*

2.3.2 Aufgabenverteilung

Die Kommunikation in Netzwerken dient sowohl Informationsaustausch verschiedener Personen, als auch verschiedener Endstellen. Der Zweck einer solchen „automatisierten“ Kommunikation die unabhängig von beteiligten Personen stattfindet beruht auf einer Aufgabenverteilung im Netzwerk.

Hierfür bieten die einzelnen Knoten bestimmte Dienste, zum Beispiel Informationen oder Ressourcen an, welche andere Knoten in Anspruch nehmen können. Zur Strukturierung wurden zwei verschiedene Konzepte entwickelt: Das Peer-to-Peer Modell und das Client-Server Modell. Nach diesen beiden Modellen spricht man bei einer Kommunikation bezüglich eines bestimmten Dienstes auch von einem Peer-to-Peer bzw. Client-Server Netzwerk. Gemeint sind damit die einzelnen Kommunikation-Teilnehmer (als Knoten), sowie die logische Topologie der Kommunikation (als Kanten).

Peer-to-Peer Modell

Im *Peer-to-Peer Modell* (v. engl.: „peer“, Gleichgestellter) sind die je zwei Kommunikationspartner, sog. *Peers* in dem Sinne gleichgestellt, dass sie sich gegenseitig die gleichen Dienste anbieten und eine Kommunikation somit von beiden Seiten initiiert werden kann. Dieses Grundkonzept wird in modernen Umsetzungen derart erweitert, dass die einzelnen Kommunikationsteilnehmer in Gruppen aufgeteilt werden, die eine Hierarchie in der Kommunikation definieren. Dies wird beispielsweise verwendet, um zentrale Verwaltungsdienste auf wenige Hosts zu beschränken, welche sich untereinander synchronisieren. Die logische Topologie von Peer-to-Peer Netzwerken ist die von dezentralen Netzwerken (Vermaschung). Je nach Umsetzung einer hierarchischen Gruppierung ist das Netzwerk dabei mehr oder weniger stark zentralisiert. Aufgrund der logischen Topologie übernehmen Peer-to-Peer Netzwerke die Eigenschaft sehr stabil auf Ausfälle einzelner Knoten oder Verbindungen zu reagieren. Beispiele für Peer-to-Peer Netzwerke sind Filesharing-Netzwerke oder VoIP-Netzwerke.

Client-Server Modell

Das *Client-Server Modell* beschreibt eine hierarchische Aufgabenverteilung, bei der jeder Kommunikationsteilnehmer einen bestimmten Dienst zur Verfügung stellt, oder diesen Dienst nutzt. In diesem Kontext wird der Dienstleister als *Server* (v. engl. „to serve“, bedienen) und der Nutzer als *Client* (engl.: Kunde) bezeichnet. Eine Kommunikation wird dabei stets vom Client initiiert. Ein Server kann dabei gegenüber einem anderen Kommunikationsteilnehmer wiederum einen Client darstellen. Daher haben Client-Server Netzwerke eine logische Baum-Topologie bzw. im einfachsten Fall eine Stern-Topologie. Das Client-Server Modell vereint viele wichtige Stärken: 1) Ressourcen können zentralisiert werden (Speicherkapazität, Rechenzeit), 2) Organisation und Verwaltung kann zentralisierter werden, 3) Client-Server Netzwerke können leicht ausgebaut werden, da an einen zusätzlichen Client nur minimale Anforderungen gestellt werden.

Beispiele für die Aufgabenverteilung im Client-Server Modell sind DHCP (4.4.1) und DNS (4.4.2).

Gemischte Aufgabeverteilung

In größeren Kommunikationsnetzwerken werden häufig beide Modelle miteinander eingesetzt, um komplexe Beziehungen zwischen den Teilnehmern abzubilden. Dadurch gelingt es die logische Topologie an die physische Topologie anzupassen, um Beispielsweise eine Kommunikation mittels möglichst kurzer physikalischer Übertragungspfade zu gewährleisten. Ein Beispiel für diese gemischte Aufgabenverteilung ist NTP (4.4.3) wobei dies hier mittels verschiedener Modi umgesetzt wird.

2.4 Aufgaben zu Kapitel 2

Aufgabe 2.1

In Abschnitt 2.1.4 steht: „Bei einer einzigen Zelle entspricht die physische Topologie einer Bus-Topologie und die logische einer Stern-Topologie“. Begründen Sie diese Aussage.

Aufgabe 2.2

Skizzieren Sie einen minimalistischen Graphen für das Internet, bei dem anhand der Topologie Zentralbereich, autonome Systeme und Zugangsbereich erkennbar sind.

Aufgabe 2.3

Bestimmen sie zu diesem Graphen die Konnektivität, den Durchmesser und die Bisektionsweite. Schätzen Sie grob die tatsächlichen Größen für das Internet.

Aufgabe 2.4

Was versteht man unter Kosten? Was ist zur Bestimmung von Kosten erforderlich und wozu werden sie verwendet?

Aufgabe 2.5

Betrachten Sie den Graph in Abb. 2.1. Der Knoten v_1 möchte Daten zu v_7 übertragen. Als Metrik kommt die Anzahl der Kanten (Hop-Count) zum Einsatz. Bestimmen Sie die Kosten der kürzesten Pfade, die über die einzelnen Nachbarn von v_1 gehen. Erstellen Sie eine Routing-Tabelle für diese Übertragung, welche Routen über die einzelnen Nachbarn von v_1 berücksichtigt.

Aufgabe 2.6

Betrachten Sie wieder den Graph in Abb. 2.1 und erstellen Sie für den Knoten v_1 eine Routing-Tabelle unter Verwendung des Dijkstra-Algorithmus. Führen Sie hierfür zunächst schrittweise die Liste der bekannten Knoten $B_1 = (v_1)$, $B_2 = (...)$, usw.

Aufgabe 2.7

Warum eignet sich Anycast für eine flächendeckende Bereitstellung von Diensten? Was muss der Betreiber dabei berücksichtigen? Finden Sie jeweils ein konkretes Beispiel für die sinnvolle Anwendung von Broadcast und Multicast.

Aufgabe 2.8

Nennen Sie jeweils drei wichtige Vorteile für die Aufgabenverteilung nach dem Peer-to-Peer Modell und dem Client-Server Modell.

Aufgabe 2.9

Entwerfen Sie eine leistungseffiziente Topologie zur Bisektionsweite n. Welche Komplexität besitzt diese Topologie?

3 PROTOKOLLE UND SCHICHTENMODELLE

In den Kapiteln 1 und 2 haben wir uns nach und nach mit immer abstrakteren Aspekten der Datenübertragung in Netzwerken beschäftigt. Dabei wurden bestimmte Abstraktionsebenen zur Formulierung der auftretenden Probleme und Möglichkeiten angewandt. In den Kapiteln 3 und 4 wollen wir uns nun mit dem Ziel der technischen Anwendung dieses Wissens mit konkreten Umsetzungen beschäftigen. Hierfür ist es zunächst jedoch erforderlich die Abstraktionsebenen zu strukturieren und klare Vorgaben für Umsetzung zu formulieren.

In Abschnitt 3.1 werden wir uns sehr allgemein mit der strukturierten Umsetzung von Anforderungen in Computernetzwerken beschäftigen. Die Grundbegriffe hierfür sind „Protokolle“ und „Schichtenmodelle“, welche in den Abschnitten 3.1.1 und 3.1.2 erläutert werden. Protokolle sind Regelungen, die der Erfüllung spezieller Aufgaben dienen. Dabei bieten sie ihre Dienste anderen Protokollen sowohl innerhalb des gleichen, als auch auf anderen Computersystemen an. Auf diese Art und Weise entstehen Kommunikationen zwischen verschiedenen Protokollen. Je Nachdem, ob die Kommunikation zwischen gleichen Protokollen, also solchen auf der gleichen Abstraktionsebene oder verschiedenen Protokollen stattfindet, wird zwischen einer horizontalen und einer vertikalen Kommunikation unterschieden.

Im Abschnitt 3.2 werden wir uns mit dem sog. ISO-OSI-Modell beschäftigen. Das OSI-Modell stellt die Grundlage für alle auf Computernetzwerken basierenden Kommunikationssystemen dar. Wir wollen uns daher in Abschnitt 3.2.1 zunächst allgemein und in Abschnitt 3.2.2 im Speziellen mit den einzelnen Schichten auseinandersetzen, wobei wir feststellen können, dass die drei unteren Schichten des OSI-Modells bereits in den Kapiteln 1 und 2 diskutiert wurden.

3.1 Protokolle und Schichtenmodelle

3.1.1 Protokolle

Die Kommunikation in Netzwerken wird durch sog. „Protokolle“ geregelt, welche sowohl die Datenübertragung als auch die verwendeten Datenformate definieren.

Def. 3.1 (Protokoll) *Netzwerkprotokolle definieren Regeln und Verfahren zur Datenübertragung in einem Netzwerk. Die Umsetzung eines Protokolls durch einen Prozess (ein im Ablauf befindliches Computerprogramm) wird als Instanz bezeichnet.*

Damit Protokolle zur Datenübertragung angewandt werden können ist es erforderlich, dass diese untereinander Informationen austauschen. Beispielsweise ein Protokoll zur Fehlerkorrektur, verursacht auf der Senderseite zusätzliche Informationen, z.B. in Form von Paritätsbits. Diese Informationen werden auf der Empfängerseite wiederum benötigt um die Fehlerkorrektur anzuwenden. Diese zusätzlichen Informationen werden als *Steuerdaten* bezeichnet. Dem gegenüber heißen die ursprünglichen Daten *Nutzdaten*. Wird eine Teil der Steuerdaten vor den Nutzdaten übertragen, so wird dieser Teil als Kopf bzw. *Header* bezeichnet. Wird eine Teil der Steuerdaten nach den Nutzdaten übertragen so heißt dieser Teil Nachspann oder *Trailer*.

Hierarchie von Protokollen

Aufgrund einer Vielzahl an unterschiedlichen Anforderungen erfüllen einzelne Protokolle häufig nur Teilanforderungen bzw. Teilaufgaben. Daher ist häufig die gemeinsame Anwendung mehrere Protokolle erforderlich, um die Datenübertragung unter Berücksichtigung gegebener Anforderungen zu ermöglichen. Um dies zu ermöglichen, müssen die Protokolle hinsichtlich ihrer Schnittstellen aufeinander abgestimmt sein. Eine solche Sammlung von Protokollen wird dann auch als *Protokollfamilie* bezeichnet. Dabei herrschen innerhalb von Protokollfamilien aufgrund der Beziehung zwischen den einzelnen Protokollen strenge Hierarchien:

- Erfüllen Protokolle unabhängig voneinander bestimmte Teilaufgaben und sind somit nicht aufeinander angewiesen, so ist die gleichzeitige Verwendung z.B. auf verschiedenen Übertragungskanälen, oder abwechselnd möglich. In diesem Fall spricht man von einer *horizontalen Anordnung der Protokolle* und sie stehen auf einer Hierarchiestufe innerhalb der Protokollfamilie.

  Diese Anordnung von Protokollen tritt entweder auf, wenn verschiedene Teile der Daten verschiedene Anforderungen an die Übertragung stellen (z.B. Videostreaming und Texte), oder die Übertragung redundant erfolgt und die Daten mehrfach unter verschiedenen Anforderungen übertragen werden sollen (z.B. über WLAN und über Kabel).

- Wenn Protokolle zur Erfüllung der Aufgaben jedoch aufeinander angewiesen sind, so entsteht automatisch eine Hierarchie zwischen diesen. Die technische Umsetzung dieser Beziehung erfolgt durch jeweilige die Einbettung der hierarchisch höhergestellten Protokolle. Dabei werden die Steuer- und Nutzdaten des einen Protokolls als Nutzdaten des anderen aufgefasst. In diesem Fall spricht man von einer *vertikalen Anordnung der Protokolle*.

Protokollstapel

Ein *Protokollstapel* (engl.: *Protocol Stack*) bezeichnet die vertikale Anordnung mehrerer Protokolle. Dabei heißt ein Protokoll, welches ein anderes einbettet tieferliegender als das andere. Das niedrigste Protokoll im Stapel ist somit das Protokoll, welches von keinem anderen eingebettet wird und das höchste das, welches selbst kein Protokoll einbettet.

Wenn man nun die Protokolle des Protokollstapels nach unten läuft so ist zu beobachten, dass die von den Protokollen erfüllten Aufgaben immer grundlegender werden. Dies hat den Grund, dass höher liegende Protokolle die Aufgaben, welche von niedrigeren Protokollen zu erledigen sind als erfüllt betrachten können. So muss sich beispielsweise ein Protokoll, welches für die Fehlerkorrektur verantwortlich ist nicht mit der eigentlichen Übertragung beschäftigen. Ein niedrigeres Protokoll bietet also dem höheren bestimmte „Dienste“ an.

Dienste

Def. 3.2 (Dienst) *Ein Dienst (engl.: Service) bezeichnet definierte Daten und Aufgaben, die ein Protokoll in einem Protokollstapel den höheren zur Verfügung stellt. Der Anbieter eines Dienstes heißt Service Provider und der Benutzer Service User*.

Def. 3.3 (Service Access Point) *Die Schnittstelle, über welche ein Protokoll einen Dienst anbieten heißt Service Access Point und wird mittels der eindeutigen Zuordnung zu Service Provider „N“ und Service User „M“ auch als „N/M“ - Interface bezeichnet.*

In einem Protokollstapel sind die Begriffe „Nutzdaten“ und „Steuerdaten“ nicht eindeutig, sondern von der Sicht eines Protokolls abhängig. Aus der Sicht eins bestimmten Protokolls *N* wird die Einheit aus dessen Steuerdaten und Nutzdaten als *Protocol Data Unit* (PDU) bezeichnet. Hierfür wird die Schreibweise PDU(*N*) verwendet. Wenn dieses Protokoll nun in das tiefer liegende Protokoll *N*-1 eingebettet wird, so entspricht die PDU von *N* den Nutzdaten von *N*-1. Aus der Sicht des Protokolls *N*-1 bietet dieses dem Protokoll *N* nun bestimmte Dienste an. *N* heißt in diesem Zusammenhang dann *Service User* (v. engl. Dienstnutzer) und *N*-1 der *Service Provider* (v. engl. Dienstleister). Daher werden die Nutzdaten von *N*-1 als *Service Data Unit* von *N*-1 bezeichnet. Man schreibt hierfür SDU(*N*-1) und es gilt der Zusammenhang: PDU(*N*) = SDU(*N*-1)

In Bezug auf einen Protokollstapel bestehend aus *N* Protokollen ergibt sich nun folgende Situation bei einer Datenübertragung:

- **Sender**

  Beim Sender werden mit dem höchsten Protokoll *N* des Stapels beginnend die zu übertragenden Nutzdaten mit den Steuerdaten von *N* zur PDU(*N*) zusammengefasst und an das nächsttiefere Protokoll *N*-1 übergeben. Dieses fasst die Daten wiederum als Nutzdaten auf und ergänzt diese mit den Steuerdaten zu PDU(*N-1*). Dieser Vorgang wiederholt sich bis zum tiefest liegenden Protokoll 1. Schließlich wird die PDU(1) des tiefest liegenden Protokolls versendet.

- **Empfänger**

  Beim Empfänger werden die übertragenen Daten vom tiefest liegenden Protokoll im Protokollstapel angenommen. Anschließend werden die jeweiligen Steuerdaten entfernt und die SDU(1) an das nächsthöhere Protokoll gegeben. Dieses entfernt wiederum die eigenen Steuerdaten und gibt die SDU(2) an das nächst höhere. Dieser Vorgang setzt sich bis zu Protokoll *N* fort. Die SDU(*N*) entspricht den zu übertragenden Nutzdaten.

3.1.2 Schichtenmodelle

Ein bestimmter Protokollstapel erfüllt nun fest definierte Anforderungen bei der Datenübertragung. Da allerdings unterschiedliche Anforderungen bei der Datenübertragung auftreten können ist eine Vielzahl von verschiedenen Protokollstapeln erforderlich. Ein vernünftiger Ansatz zur Arbeitserleichterung ist die einzelnen Protokolle nun so zu gestalten dass die Anforderungen wie mit einem Baukasten aus Protokollen zusammengebaut werden können. Hierfür werden die Protokolle in sog. *Schichten* in Aufgabenbereiche zusammengefasst. Dabei sind die Protokolle einer Schicht horizontal und verschiedener Schichten vertikal angeordnet sind. Des Weiteren werden den Schnittstellen zwischen den Schichten definiert, so dass die Protokolle austauschbar sind.

Def. 3.4 (Schichtenmodell) *Ein Modell zur Datenübertragung in Netzwerken, welches einzelne Schichten definiert heißt Schichtenmodell. Ein Schichtenmodell, welches aus n Schichten besteht wird auch als n-Schichten Architektur bezeichnet.*

Innerhalb eines Schichtenmodells existieren nun zwei verschiedene Möglichkeiten wie Instanzen miteinander kommunizieren können.

Horizontale Kommunikation

Die Kommunikation zwischen Instanzen der gleichen Schicht wird *als horizontale Kommunikation* bezeichnet (Abb. 3.1). Hierfür können die Steuerdaten dieser Schicht verwendet werden, da beide Seiten Zugriff auf diese haben. Dabei kann die Kommunikation sowohl innerhalb eines Hosts, als auch über ein Netzwerk zwischen verschiedenen Hosts erfolgen.

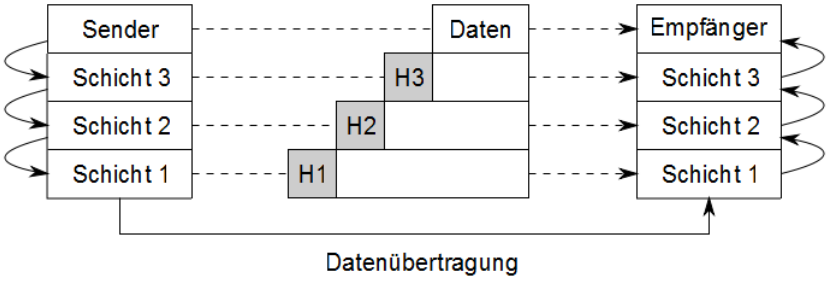

Abb. 3.1 *Horizontale Kommunikation*

Vertikale Kommunikation

Beim Senden können nur niedrigere Instanzen auf die Steuerdaten höherliegender und beim Empfang nur die höherliegenden Instanzen auf die Steuerdaten niedrigerer zugreifen. Daher ist zwischen Instanzen verschiedener Schichten von Natur aus keine Kommunikation möglich. Um diese dennoch zu ermöglichen wird eine sehr niedrige Schicht damit beauftragt die Daten zwischen den beiden höheren Schichten weiterzuleiten.

Die Kommunikation zwischen Service Provider und Service User beruht auf einfachen Steuerzeichen, sog. *Service Primitives*, die von den Protokollen „gesetzt“ werden. Gesetzt bedeutet hierbei, dass die Kommunikation passiv geführt wird, indem bei den ohnehin übertragenen Daten bestimmte Steuerdaten verändert werden. Diese lassen sich in vier Gruppen unterteilen: *Request*, *Confirm*, *Indication* und *Response*:

- Ein *Request Primitive* bezeichnet die Anforderung eines Dienstes durch einen Service User.

- Wurde ein Dienst angefordert, so setzt der Service Provider sobald die angeforderten Daten zur Verfügung stehen (bzw. die Aufgaben erledigt wurde) eine Mitteilung, ein sog. *Confirm Primitive* an den Service User.

- Ein *Indication Primitive* bezeichnet einen Hinweis, den der Service Provider unabhängig von einer konkreten Anforderung setzt.

- Um auf einen solchen Hinweis zu reagieren setzt der Service User ein *Response Primitive*.

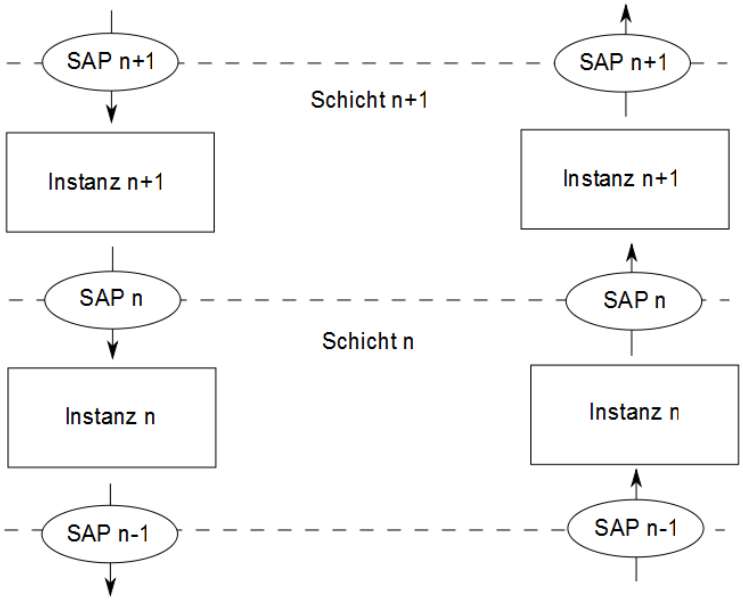

Abb. 3.2 *Vertikale Kommunikation*

3.2 ISO-OSI-Referenzmodell

3.2.1 Übersicht über das OSI-Modell

Das *ISO-OSI-(Referenz-) Modell* (kurz *OSI-Modell*) (v. engl. Open Systems Interconnection) ist ein Schichtenmodell der Internationalen Organisation für Normung (ISO) und beschreibt eine 7-Schichten Architektur zur Kommunikation sog. *offener Systeme*. Offene Systeme bezeichnen dabei Kommunikationssysteme, die auf offenen Schnittstellen und Spezifikationen basieren. D.h. dass die Informationen über Dienste und Service Access Points der einzelnen Instanzen öffentlich zugänglich sind. Dem gegenüber bezeichnen *geschlossene Systeme* solche Kommunikationssysteme, bei denen diese unveröffentlicht bleiben, wodurch insbesondere Fremdsysteme von einer Kommunikation ausgeschlossen werden.

Das OSI-Modell dient als Designgrundlage, sowie allgemein als Rahmen zur Beschreibung von Protokollcharakteristika und Funktionen. Zu Gunsten einer universellen Anwendbarkeit werden die Schichten und Dienste im OSI-Modell unabhängig von Herstellern und Übertragungsmedien und möglichst allgemein beschrieben. Die sieben Schichten des OSI-Modells lassen sich nach ihrer Funktion in transportorientierte Schichten (Schicht 1-4) und in anwendungsorientierte Schichten (Schicht 5-7) unterteilen (Tbl. 3.1). Zudem wird das Übertragungsmedium (bzw. das Signalübertragungssystem) selbst manchmal als „Schicht 0“ bezeichnet.

Innerhalb der transportorientierten Schichten 1 bis 4 werden die Nutzdaten zwischen den Schichten übergeben, wobei sie je nach Kommunikationsrichtung (Versand, Empfang) in

kleinere Blöcke, sogenannte Datagramme geteilt oder aus solchen wieder zusammengesetzt werden. Man unterscheidet drei Typen von Datagrammen: „Datenframes“ in Schicht 2 (Def. 3.5), „Datenpakete“ in Schicht 3 (Def. 3.6) und „Datensegmente“ in Schicht 4 (Def. 3.7). Die Daten höherer Schichten werden ja nach Typ als „Datenstrom“ (engl.: Stream) oder „Nachricht“ (engl.: Message) bezeichnet.

Des Weiteren werden bei der Übergabe von Daten zwischen den Schichten (und innerhalb von Schichten) Metainformation in Form von Headern und Trailern beigefügt oder entfernt. Dieser Vorgang wird als *Datenkapselung* bezeichnet.

3.2.2 OSI-Modell

<table>
<tr><th>Funktion</th><th>Schicht</th><th>Bezeichnung</th><th>Teilschichten</th></tr>
<tr><td rowspan="3">Anwendung</td><td>7</td><td colspan="2">Anwendungsschicht</td></tr>
<tr><td>6</td><td colspan="2">Darstellungsschicht</td></tr>
<tr><td>5</td><td colspan="2">Sitzungsschicht</td></tr>
<tr><td rowspan="5">Transport</td><td>4</td><td colspan="2">Transportschicht</td></tr>
<tr><td>3</td><td colspan="2">Vermittlungsschicht</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Sicherungsschicht</td><td>LLC-Schicht</td></tr>
<tr><td>MAC-Schicht</td></tr>
<tr><td>1</td><td colspan="2">Bitübertragungsschicht</td></tr>
<tr><td colspan="4">Übertragungsmedium (Signalübertragungssystem)</td></tr>
</table>

Tbl. 3.1 *OSI-Modell*

Bitübertragungsschicht

Die *Bitübertragungsschicht* (engl.: Physical Layer) ist für den physikalischen Transport der Daten verantwortlich und beschreibt somit die Funktion eins Datenübertragungssystems. Dabei wird die gemeinsame Nutzung eines Übertragungsmediums durch Multiplexing ermöglicht. Zur Überwachung der Datenübertragung werden in zyklischen Abständen die Steuerleitungen geprüft.

Sicherungsschicht

Die *Sicherungsschicht* bzw. *Datensicherungsschicht* (engl.: Data Link Layer) wird in zwei Teilschichten (engl.: Sublayer) unterteilt:

- **MAC-Schicht**

  Die *MAC-Schicht* (engl.: Medium Access Control) verwaltet die Zugriffsverfahren auf das physikalische Medium. Dabei werden auftretende Kollisionen behandelt und Daten in Datenframes gekapselt, welche mittels physikalischer Adressen, sog. *MAC-Adressen* (OSI Schicht-2 Adressen) adressiert werden. Zudem werden die zu Übertragenden Daten mit Prüfsummen zur Fehlererkennung versehen.

- **LLC-Schicht**

  Die *LLC-Schicht* (engl.: Logical Link Control) liegt über der MAC-Schicht und bietet einen Verbindungsdienst. (Def. 3.5). Die Broadcast-Domäne (also die Reichweite von Broadcasts) der Sicherungsschicht wird als *Link-Layer Broadcastdomäne* bezeichnet und umfasst alle Hosts eines Netzwerkes, bei denen eine horizontale Kommunikation innerhalb dieser Schicht möglich ist. Der Verbindungsdienst ergänzt schließlich die einzelnen Frames um Informationen über den Dienst welcher Daten empfängt (DSAP, Destination Service Access Point) und den Dienst, welcher die Daten verschickt (SSAP, Source Service Access Point).

Def. 3.5 (Datenframe) *Als Datenframe wird die Dateneinheit der Sicherungsschicht des OSI-Modells (bzw. der Netzzugangsschicht des TCP/IP-Modells) bezeichnet. Die Kapselung in Datenframes dient der Umsetzung von Zugriffsverfahren. Datenframes enthalten MAC-Adressen, SAP-Adressen und Fehlerkorrektur.*

Vermittlungsschicht

Da im Normalfall keine direkte Kommunikation zwischen Absender und Ziel möglich ist, wird auf der Vermittlungsschicht (engl.: Network Layer) die Wahl des Übertragungsweges vorgenommen. Dies erfolgt mittels logischer Adressierung der Hosts sowie horizontaler Kommunikation innerhalb der Vermittlungsschicht. Hierfür werden die von der Transportschicht erhaltenen Datensegmente (Def. 3.7) mit logischen Adressen (OSI Schicht-3 Adressen) zu sog. Datapaketen ergänzt (Def. 3.6). Weitervermittelte Datenpakete gelangen nicht in die höheren Schichten, sondern werden mit einem neuen Zwischenziel versehen und an den nächsten Host gesendet. Die Vermittlungsschicht stellt höheren Schichten somit eine vollständige Ende-zu-Ende Kommunikation (zwischen Sender und Empfänger) zur Verfügung, welche unabhängig von der Netzwerktopologie ist.

Def. 3.6 (Datenpaket) *Ein Datenpaket bezeichnet die Dateneinheit der Vermittlungsschicht des OSI-Modells (bzw. der Internetschicht des TCP/IP-Modells). Die Kapselung in Datenpakete dient der Paketvermittlung. Datenpakete enthalten u.a. logische Adressen der Start- und Endknoten und Steuerinformationen für das Routing.*

Transportschicht

Die *Transportschicht* (engl.: Transport Layer) bietet einen Datagrammdienst mit Datenflusskontrolle, sowie Mechanismen zur Kommunikationssteuerung zwischen den Kommu-

nikationsendpunkten. Hierbei erhält der Dienst aufgrund der Ende-zu-Ende Kommunikation jedoch ein Feedback vom Empfänger (Ende-zu-Ende Quittungen) und keiner Zwischenstation. Dies ermöglicht dem Dienst Verfahren anzuwenden um die Gefahr von Staus in der Datenübertragungskette zu vermindern. Beim Versenden werden die Daten der höheren Schichten in Datensegmente gekapselt, deren Größe dem kleinsten Puffer eines Zwischenknotens in der Übertragungskette entspricht. Dieser wird auch als MTU (engl.: Maximum Transmission Unit) bezeichnet. Beim Empfang werden die Datensegmente zu einem Datenstrom entkapselt. Dabei wird anhand einer zusätzlichen Adresse, der sog. *Portadresse* (OSI Schicht-4 Adresse) analysiert, an welchen Kommunikationsendpunkt der Inhalt des Datensegments adressiert ist. Dieser beschreibt eine Instanz eines höheren Protokolls. Dadurch ist es möglich dass in höheren Schichten gleichzeitig mehrere „Verbindungen“ bestehen können.

Def. 3.7 (Datensegment) *Ein Datensegment ist die Dateneinheit der Transportschicht des OSI-Modells (bzw. der Transportschicht des TCP/IP-Modells). Die Kapselung in Datensegmente wird als Segmentierung bezeichnet und dient der Flusskontrolle. Datensegmente enthalten u.A. die Adressen der Kommunikationsendpunkte, Verbindungsinformationen (Aufbau, Abbau, etc), Reihenfolge von Daten, Priorität von Daten, Informationen zur Bestimmung der MTU*

Sitzungsschicht

Die *Sitzungsschicht* (engl.: Session Layer) umfasst Dienste zur Interprozesskommunikation zwischen mehreren Systemen. Diese umfassen die Ordnung des Ablaufs einer Kommunikation mit mehreren Teilnehmern durch Berechtigungsmarken, die Synchronisierung des Datenaustauschs durch das Setzen von sogenannten *Fixpunkten*. Diese ermöglichen die Wiederaufnahme der Kommunikation, falls die Transportverbindung abbricht, ohne dass eine Datenübertragung neu gestartet werden muss.

Darstellungsschicht

Die *Darstellungsschicht* (engl.: Presentation Layer) stellt Dienste zu Verfügung, die es ermöglichen, dass Daten die von der Anwendungsschicht eines Systems gesendet wurden, von der Anwendungsschicht eines anderen Systems gelesen werden können. Diese Dienste umfassen z.B. die Ermittlung eines gemeinsam verwendbaren Zeichensatzes (zum Beispiel ASCII, EBCDIC). Sollte kein gemeinsamer Zeichensatz existieren, so werden die lokalen Zeichensätze in das für beide verständliche System ASN.1 (Abstract Syntax Notation One) übersetzt. Des Weiteren umfasst die Darstellungsschicht auch Dienste wie Datenkompression oder Verschlüsselung.

Anwendungsschicht

Auf der *Anwendungsschicht* (engl.: Application Layer) befinden sich die einzelnen Protokolle, welche Programme zu Erfüllung ihrer Dienste benötigen. Diese Schicht beschreibt die Verbindung der einzelnen Anwendungsprogramme innerhalb eines vernetzten Systems.

3.3 Aufgaben zu Kapitel 3

Aufgabe 3.1

Warum können Protokolle unterschiedlicher Schichten nicht „direkt“ (mittels Steuerdaten) miteinander kommunizieren?

Aufgabe 3.2

Wie wird dieses Problem im OSI-Modell gelöst? Verwenden Sie zur Beschreibung folgende Begriffe: „SSAP“, „DSAP“, „LLC-Schicht“, „Service Primitives“

Aufgabe 3.3

Ein Sender und ein Empfänger implementieren den gleichen Protokollstapel mit N Protokollen. Skizzieren Sie eine Datenübertragung unter Verwendung der Beschriftungen „PDU(...)“ und „SDU(...)“.

Aufgabe 3.4

Ordnen Sie folgende Begriffe den entsprechenden Schichten (bzw. Teilschichten) des OSI-Modells zu (Beschreibung der Begriffe ist nicht erforderlich): „Datenframe“, „Datenpaket“, „Datensegment“, „Bitstrom“, „DSAP“, „Flusskontrolle“, „Signal“, „SSAP“, „MTU“, „Multiplexing“, „Routing“, „Zugriffsverfahren“.

Aufgabe 3.5

Welche Arten der Adressierung treten in den transportorientierten Schichten des OSI-Modells auf und wofür werden sie benötigt?

4 TCP/IP NETZWERKE

TCP/IP bezeichnet eine Familie von Protokollen zur Realisierung der Kommunikation in Computernetzwerken. Das 1970 vom United States Department of Defense (DoD) entwickelte DoD-Modell stellte hierfür die Grundlage. Durch Weiterentwicklungen innerhalb der entsprechenden Protokollfamilie entstand schließlich das TCP/IP-Modell, welches nach seinen beiden wichtigsten Protokollen „Internet Protocol“ (IP) und „Transmission Control Protocol“ (TCP) benannt wurde. Da die TCP/IP Protokollfamilie die Basis für die Netzwerkkommunikation des Internet bildet, wird sie auch als *Internetprotokollfamilie* bezeichnet.

Im Folgenden wollen wir mit dieser Protokollfamilie näher beschäftigen. Das weit jüngere OSI-Modell (erste Standardisierung 1981, aktueller Standard von 1994) stellt eine Weiterentwicklung des TCP/IP-Modells dar und ist diesem im Allgemeinen auch vorzuziehen. Eine Ausnahme hier stellt jedoch die Frage, warum die Internetprotokollfamilie so strukturiert ist, denn moderne Protokolle basierend auf dem OSI-Modell sind oft modularer oder flexibler. Daher werden wir uns in Abschnitt 4.1 mit dem TCP/IP-Modell befassen, wobei wir immerhin die Sprache des OSI-Modells wiederfinden werden (Frames, Pakete, etc.). Im Wesentlichen beschreiben die transportorientierten Schichten im TCP/IP-Modell drei Aufgaben: Datenübertragung in der „Sicherungsschicht“, Netzwerkmanagement in der „Internetschicht“ und Datenflusskontrolle in der „Transportschicht“.

Mit den ersten beiden Aufgaben haben wir uns bereits in den ersten beiden Kapiteln beschäftigt. Da das TCP/IP-Modell jedoch keine Spezifikation der Datenübertragung umfasst werden wir uns in Abschnitt 4.2 direkt mit den Protokollen der Internetschicht, insbesondere „IPv4“ und „IPv6“ beschäftigen. Im Abschnitt 4.3 folgen die wichtigsten Protokolle der Transportschicht, „TCP“ und „UDP“ und abschließend in Abschnitt 4.4 mit „DHCP“, „DNS“ und „NTP“ einige wichtige Vertreter der „höheren“ Protokolle im Netzwerkmanagement.

4.1 TCP/IP Modell

<table>
<tr><th>Funktion</th><th>Schicht</th><th>Bezeichnung</th><th>OSI-Modell</th></tr>
<tr><td rowspan="3">Anwendung</td><td rowspan="3">4</td><td rowspan="3">Anwendungsschicht</td><td>7</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td></tr>
<tr><td rowspan="2">Transport</td><td>3</td><td>Transportschicht</td><td>4</td></tr>
<tr><td>2</td><td>Internetschicht</td><td>3</td></tr>
</table>

	1	Netzzugangsschicht	2
			1
Übertragungsmedium (Signalübertragungssystem)			

Tbl. 4.1 *TCP/IP-Modell*

Netzzugangsschicht

Die Netzzugangsschicht (engl.: Link-Layer) umfasst die Schichten 1 und 2 des OSI-Modells. Die Spezifikation im TCP/IP-Modell beschränkt sich dabei aber lediglich auf das Vorhandensein eines Datenübertragungssystems in Form von Datenframes (Def. 3.5). Diese allgemeine Formulierung wurde mit dem Ziel gewählt, auf möglichst vielen Technologien basierend (und damit möglichst vielen Herstellern) eine Implementierung des TCP/IP Protokollstacks zu ermöglichen.

Beispiele für Protokolle der Netzzugangsschicht:

- *Ethernet*: Standards zur kabelgebundenen Datenübertragung, die ursprünglich für lokale Datennetze (LANs) entwickelt wurden.
- *Fiber Distributed Data Interface* (FDDI): Technologie zum breitbandigen Datentransport über Glasfaser.
- *Asynchronous Transfer Mode* (ATM): Breitbandiger Datenverkehr mittels sehr kleiner Datenblöcke (Zellen, Slots) mit fester Länge.
- *IEEE 802.11*: Datenverkehr in lokalen Funknetzwerken (WLAN)

Internetschicht

Die *Internetschicht* (engl.: Internet-Layer) entspricht der Vermittlungsschicht des OSI-Modells und ist vom Datenübertragungssystem unabhängig. Die Aufgaben umfassen somit die Paketvermittlung, sowie das Routing. Zu diesem Zwecke werden den Hosts logische Adressen zugeordnet, die den Host innerhalb eines bestimmten Netzwerkes und dieses Netzwerk innerhalb des Internet eindeutig identifizieren. Diese Informationen ermöglichen es Routern Datenpakete entsprechend weiterzuleiten.

Beispiele für Protokolle der Internetschicht:

- *Internet Protocol* (IP, IPv4, IPv6): Protokolle zur Paketvermittlung in Netzwerken. (4.2.2 und 4.2.4)
- *Internet Control Message Protocol* (ICMP): Protokoll zum Austausch von einfachen Nachrichten und Fehlermeldungen

- *Open Shortest Path First* (OSPF): Wichtigster Vertreter der Interior Gateway-Protokolle. Verwendet zur Ermittlung der Kosten den Dijkstra-Algorithmus, der auch als „Shortest Path First“ bezeichnet wird.

- *Border Gateway Protocol* (BGP): Wichtigster Vertreter der Exterior Gateway Protokolle.

Transportschicht

Die *Transportschicht* (engl.: Transport Layer) des TCP/IP-Modells ist äquivalent zur Transportschicht des OSI-Modells. Aus der Sicht der Transportschicht entspricht die Kommunikation im Netzwerk immer einer direkten Verbindung zwischen den Teilnehmern. Die Aufgabe dieser Schicht ist die Organisation und Verwaltung von Verbindungen und der Datenübertragung.

Beispiele für Protokolle der Transportschicht:

- *Transmission Control Protocol* (TCP): Protokoll zur fehlerfreien, verbindungsorientierten Datenübertragung (4.3.2)

- *User Datagram Protocol* (UDP): Minimale Implementierung der Transportschicht zur verbindungslosen Datenübertragung (4.3.3)

Anwendungsschicht

Die *Anwendungsschicht* (engl.: Application Layer) umfasst Protokolle, die direkt mit den Anwendungsprogrammen zusammenarbeiten, oder weiterführende Organisationsaufgaben im Netzwerk übernehmen. Dabei werden in Abhängigkeit des Protokolls die einzelnen Schichten des OSI-Modells teilweise oder vollständig implementiert.

Beispiele für Protokolle der Anwendungsschicht:

- *Dynamic Host Configuration Protocol* (DHCP): Zentrale Zuweisung der Netzwerkkonfiguration an Hostes (Abschnitt 4.4.1)

- *Domain Name System* (DNS): Zuordnung zwischen Namen und IP-Adressen (Abschnitt 4.4.2)

- *Network Time Protocoll (*NTP*)*: Zeitsynchronisation (Abschnitt 4.4.3)

4.2 Internetschicht

4.2.1 Logische Adressierung und das Internet Protocol

Jeder Knoten im Internet hat eine eindeutige logische Adresse, die als *IP-Adresse* bezeichnet wird. Die Vergabe der IP-Adressen erfolgt teilweise zentral. Dabei werden Netzwerke mittels einer global eindeutigen sog. *Netzwerkadressen* (engl.: *Network-ID*) identifiziert. Die Aufgabe der Zuweisung solcher Netzwerkadressen wird zentral durch die *Internet Assigned Numbers Authority* (IANA) organisiert (siehe 2.1.7). Dadurch wird die Eindeutigkeit gewährleistet. Innerhalb der einzelnen Netzwerke müssen nun aber noch die Hosts

identifiziert werden. Diese Zuordnung, sog. *Hostadressen* (engl.: *Host-ID*) unterliegt nun dem jeweiligen Netzbetreiber.

Im Internet wird die logische Adressierung der einzelnen Hosts durch das *Internet Protocol* (IP) umgesetzt, weshalb logische Adressen hier als *IP-Adressen* bezeichnet werden. Eine IP-Adresse ist eine *N*-Bit lange Adresse. Die ersten *n* Bits identifizieren das Netzwerk, entsprechen also der Netzwerkadresse, und die restlichen *N*-*n* Bits identifizieren schließlich den Host innerhalb des Netzwerkes, entsprechen somit also der jeweiligen Hostadresse.

Aufgrund wachsender Anforderung entwickelte sich das Internet Protokoll in mehreren Version weiter, wobei sich der Adressraum der IP-Adressen (die Länge der Adressen, *N*) vergrößert hat. Heute wird mit der Bezeichnung „Internet Protocol“ (IP) zumindest in der Literatur zumeist explizit die erste Version bezeichnet. Dem gegenüber sind die heute wichtigsten Versionen das „Internet Protocol Version 4“ (IPv4) und das „Internet Protocol Version 6“ (IPv6).

4.2.2 Internet Protocol Version 4 (IPv4)

Das *Internet Protocol Version 4* (IPv4) ist ein weltweit verbreitetes Protokoll zur Implementierung der Internetschicht im TCP/IP-Modell. Es wurde in den späten 70er Jahren von der Defense Advanced Research Projects Agency (DARPA) entwickelt und schließlich 1981 als Internetstandard verabschiedet.

Notation und Aufbau von IPv4-Adressen

IPv4-Adressen haben eine Länge von 32-Bit und werden in einer *Dezimalschreibweise mit Punkt* (engl.: *Dotted Decimal Notation*) dargestellt. In dieser Notation werden die 32-Bit in 4 Bytes (8-Bit) unterteilt und jeweils durch einen Punkt getrennt als Dezimalzahlen zwischen 0 und 255 geschrieben (Bsp. 4.1)

Bsp. 4.1 (IPv4-Adresse) *Die 32-Bit lange IPv4-Adresse mit der binären Darstellung* `10000010 01011110 01111010 11000011` *wird in der „Dezimalschreibweise mit Punkt“ in 8-Bit große Blöcke unterteilt und durch Punkte voneinander getrennt durch Dezimalzahlen dargestellt:* `130.94.122.195`

Diesen Adressen unterliegt eine Hierarchie von links nach rechts. Das bedeutet, dass von links nach rechts hierarchisch immer kleinere Netzwerke bis zum einzelnen Host identifiziert werden. Der dahinterliegende Sinn ist im Routing zu suchen. Durch die beschriebene Hierarchie muss ein zentraler Router nur die ersten paar Bits der Empfängeradresse eines Datenpaketes auslesen und kann dieses dann an den entsprechenden Router weiterleiten, der diesen kleineren Adressbereich verwaltet. Dies wiederholt sich so oft, bis das Datenpaket beim Empfänger ankommt.

Dabei adressieren die ersten *n* Bits der IPv4-Adresse das Netzwerk und die restlichen 32-*n* Bits schließlich den Host innerhalb des Netzwerkes. Nun stellt sich aber die Frage, wie lange die Netzwerkadresse und wie lange die Hostadresse sein soll!? Das Problem soll kurz an zwei extremen Beispielen veranschaulicht werden: Angenommen die Netzwerkadresse wäre 24-Bit lang und die Hostadresse 8-Bit. Dann hätte jedes Netzwerk 256 Hostadressen zur Verfügung. Jedoch reicht diese Anzahl für viele Netzwerke nicht aus. Ange-

nommen die Netzwerkadresse wäre hingegen 8-Bit lang und die Hostadresse somit 24-Bit. In diesem Fall könnten nur 256 Netzwerke weltweit adressiert werden.

Um eine Lösung für dieses Problem zu finden behalf man sich in der Anfangszeit, indem man Netzwerke verschiedener Größen anbot und in sog. „Netzklassen“ einteilte.

Netzklassen

Zur Aufteilung der Netzwerke nach Größen wurden von 1981 bis 1993 sog. *Netzklassen* verwendet. Diese teilten die Netzwerke in „Klasse A“, „Klasse B“ und „Klasse C“. Für jede Netzklasse wurden innerhalb des IPv4 Adressraumes eigene Adressbereiche eingerichtet, innerhalb derer Netzwerke der entsprechenden Klasse registriert werden konnten (Tbl. 4.2). Des Weiteren wurden die weiteren Adressbereiche „Klasse D“ und „Klasse E“ für besondere Zwecke reserviert.

Die feste Zuordnung in Netzklassen sollte in erster Linie das Routing dadurch vereinfachen, da die Größe der Netzwerkadresse und der Hostadresse sehr einfach anhand der IP-Adresse ausgelesen werden konnte. Diese Form des Routings wird als *Classful Routing* bezeichnet.

Netzklasse	# Netzwerke	# Hosts	Adressbereich	
Klasse A	126	16.777.214	`0.0.0.0`	`127.255.255.255`
Klasse B	16.384	65.534	`128.0.0.0`	`191.255.255.255`
Klasse C	2.097.152	254	`192.0.0.0`	`223.255.255.255`
Klasse D	*Multicast*		`224.0.0.0`	`239.255.255.255`
Klasse E	*experimentell*		`240.0.0.0`	`255.255.255.255`

Tbl. 4.2 *Netzklassen*

Subnetting und Supernetting

Bereits sehr früh wurden an dieser Klassifikation konzeptionelle Schwächen festgestellt: Manche den Netzwerken zugeordnete Adressbereiche wurden nur teilweise genutzt und andere wiederum waren zu klein. Dies resultierte in einer ineffektiven Nutzung des IPv4-Adressraumes und somit zu einer Knappheit der IPv4-Adressen. Um das Problem zu umgehen wurde 1985 die Möglichkeit eingeführt Netzwerke mittels sog. „Subnetting“ in kleinere Teilnetze aufzuteilen. Des Weiteren wurde 1992 durch sog. „Supernetting“ auch die umgekehrte Richtung, also das Zusammenführen von Netzwerken zu einem größeren ermöglicht.

- Beim *Subnetting* wird ein Adressbereich der alle Hosts des Netzwerkes umfasst eingeschränkt, indem die führenden Bits der Hostadresse, die nicht benötigt werden für eine sog. *Teilnetzadresse* zu verwenden (Abb. 4.1, Abb. 4.2). Innerhalb der Netzwerkadresse können dann mehrere kleinere Netzwerke mittels der Teilnetzadresse identifiziert werden. Somit kann mittels Subnetting der Adressbereich eines Netzwerkes auf mehrere kleine Netzwerke aufgeteilt werden.

- Beim *Supernetting* werden mehrere Netzwerke zu einem größeren zusammengefasst. Dabei wird ein größerer Adressbereich gewählt, der die Teilnetzwerke umfasst und dann wie beim Subnetting vorgegangen.

Da bei diesen Verfahren aber die Netzwerkadresse um die Teilnetzadresse erweitert wird, ist aus der IP-Adresse nicht mehr ersichtlich, wie lange die Netzwerkadresse, bzw. die Hostadresse ist. Daher wird der IP-Adresse die Information über die Länge der Netzwerkadresse beigefügt. Dies erfolgt mittels der „Subnet-Mask“.

Die *Subnet Mask* ist eine binäre „Maske“ bzw. Schablone für die IP-Adresse und hat daher die gleiche Länge von 32-Bit. Jede Stelle der IP-Adresse, die das Netzwerk adressiert hat in der Subnet Mask den Wert 1, solche die den Host adressieren den Wert 0. Die Subnet Mask wird dabei ebenso wie die IP-Adresse in der Dezimalschreibweise mit Punkt notiert. Im Beispiel von Abb. 4.2 ist die Subnet Mask `255.255.252.0`. Schließlich werden die IP-Adresse und die Subnet Mask durch einen Schrägstrich getrennt. Beispiel: `172.16.4.0/255.255.252.0`

Mit Hilfe dieser zusätzlichen Information über die Länge der Netzwerkadresse (und somit auch der Länge der Hostadresse) ist es praktisch nicht mehr notwendig eine Klassifizierung mittels „Klasse A“, „Klasse B“ und „Klasse C“ durchzuführen.

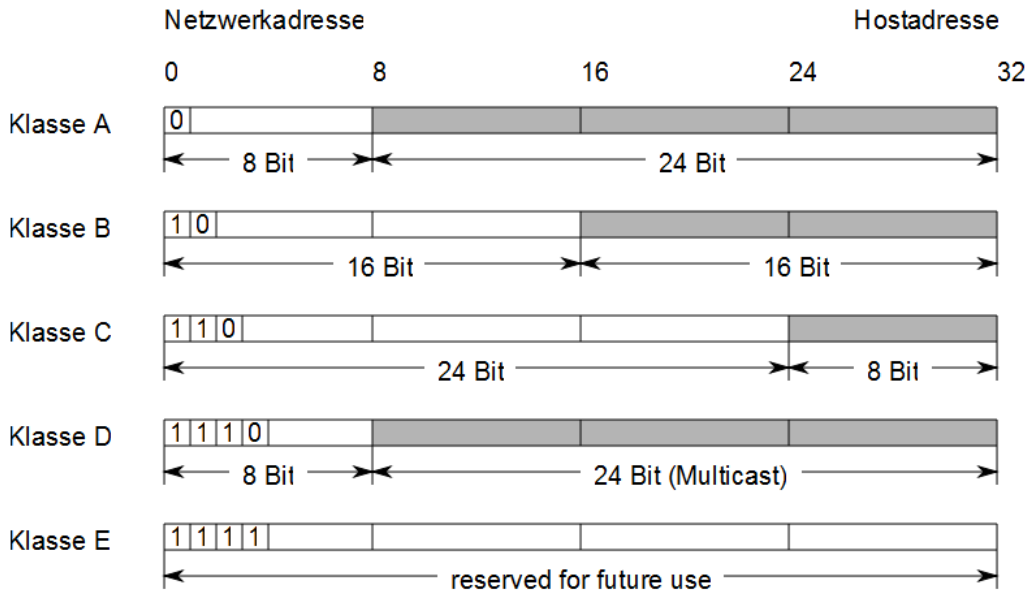

Abb. 4.1 *Übersicht der IPv4-Adressen nach Netzklassen*

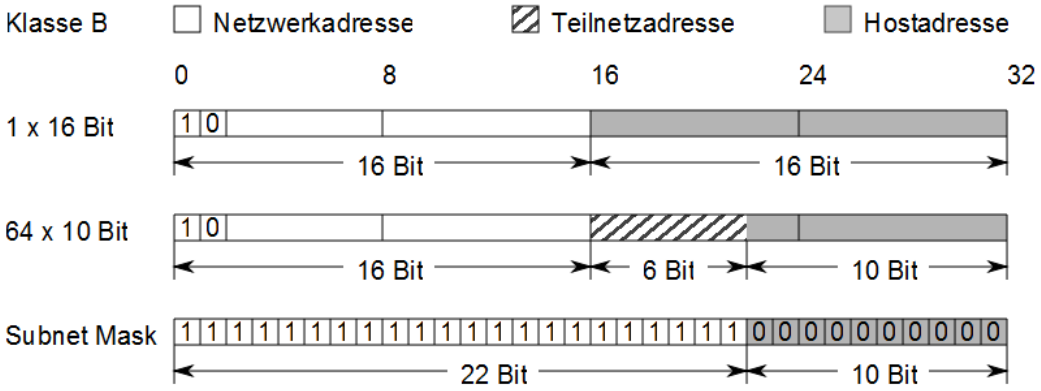

Abb. 4.2 *Aufteilung eines Klasse B Netzes in 64 Teilnetze*

Classless Inter-Domain Routing (CIDR)

Die Aufteilung in Netzklassen wurde 1993 mit der Einführung des *Classless Inter-Domain Routing* (CIDR) schließlich endgültig abgeschafft. Eine grundlegende Idee beim CIDR ist die Länge der Netzadresse hinter der IP-Adresse anzugeben. Diese Angabe wird auch als *Präfixlänge* bezeichnet und die Darstellung einer IP-Adresse mittels Präfixlänge daher als *CIDR-Notation*. Analog zur Schreibweise mittels Subnet Mask werden dabei die IP-Adresse und die Präfixlänge durch einen Schrägstrich voneinander getrennt dargestellt. Beispiel: `172.16.4.0/22`

Diese Notation ermöglicht eine kürzere Darstellung der IP-Adresse. Zu Gunsten kleinerer Routingtabellen, sowie einer effizienteren Adressvergabe wurde die Klassifikation in Netzklassen daher beim Routing vollständig aufgehoben. Jedoch wurden einigen Adressbereiche noch zu Zeiten der Netzklassen spezielle Funktionen zugeordnet, weshalb diese Adressbereiche weiterhin über Netzklassen erklärt werden können. Dies sind: „Private Netzwerke“, „Multicast-Adressen“ (früher Klasse D) und „für zukünftige Zwecke reservierte Adressen“ (früher Klasse E). Darüber hinaus werden die Bezeichnungen „Klasse A“-, „Klasse B“- und „Klasse C“-Netz teilweise (umgangssprach) dazu verwendet um Netzwerke der Größe `/8`, `/16` und `/24` zu benennen.

Private Adressbereiche

Def. 4.1 (Privates Netzwerk) *Ein Privates Netzwerk bezeichnen ein Netzwerk, welches unabhängig vom Internet verwaltet wird*.

Unabhängig bedeutet in dieser Definition insbesondere, dass keine Adressenkonflikte mit anderen Netzwerken (Eindeutigkeit der IP-Adressen) im Internet auftreten können. Dies ist der Fall, wenn das Netzwerk nicht mit dem Internet verbunden ist, oder die Hosts bezüglich des Internets eine weitere eindeutige IP-Adresse haben. Man unterscheidet dann zwischen der *privaten IP-Adresse* und der *öffentlichen IP-Adresse*. Bei der Einwahl eines Hosts eines privaten Netzwerkes in das Internet wird durch den ISP beispielsweise automatisch eine öffentliche Adresse aus dessen Adressbereich zugewiesen.

Da bei privaten Netzwerken somit nicht die Gefahr eines Adressenkonfliktes besteht, können verschiedene private Netzwerke den gleichen Adressbereich verwenden. Hierfür wur-

den Adressbereiche in verschiedenen Größen definiert (Tbl. 4.3). Zur Gewährleistung dieses Mechanismus, werden Pakete mit privaten Adressen nicht über die Grenzen des privaten Netzwerkes weitergeleitet. Man spricht daher (insbesondere im IPv6 Chargon) von Site-Local Adressen.

Zur Kommunikation zwischen Hosts aus privaten Netzwerken und beliebigen Hosts im Internet werden von den ISPs Pools mit öffentlichen Adressen verwaltet. Beim Verbindungsaufbau wird nun dem Host im privaten Netzwerk eine öffentliche IP-Adresse zugewiesen.

Netzklasse	# Netzwerke	# Hosts	Privater Adressbereich	
Klasse A	1	16.777.214	`10.0.0.0`	`10.255.255.255`
Klasse B	16	65.534	`172.16.0.0`	`172.31.255.255`
Klasse C	256	254	`192.168.0.0`	`192.168.255.255`

Tbl. 4.3 *Private Netzwerke*

Die Verwendung privater Adressbereiche hat zwei grundlegende Vorteile:

1. Die Verwaltung privater Netzwerke ist wesentlich einfacher, da die IP-Adressen nicht registriert werden müssen

2. Trotz Knappheit von IPv4-Adressen können den Hosts beliebig vieler privater Netzwerke IP-Adressen zugewiesen werden

Multicasting

Der IPv4-Adressbereich von `224.0.0.0` bis `239.255.255.255` umfasst Multicast-Adressen (früher Klasse D), die es in TCP/IP Netzwerken ermöglichen mehrere Empfänger eines Datenpaketes mit einer IP-Adresse zu adressieren. Die Empfänger, die einer bestimmten Multicast-Adresse zugeordnet werden heißen Multicast-Gruppe. Anwendung findet dieses IP-Multicasting beispielsweise für einzelne Protokolle wie das Network Time Protocol (NTP), bestimmten Routingprotokollen, sowie beim Video- oder Audiostreaming.

Die Registrierung von Multicast Adressen unterliegt direkt der IANA und nicht regionalen Behörden. Diese unterteilt dabei in Abhängigkeit von der Reichweite (durch das Routing) sowie dem Verwendungszweck in mehrere Teilbereiche (Tbl. 4.4).

Bezeichnung / Beschreibung	Adressbereich

Local Network Control: Routing Protokolle und lokale Multicast-Gruppen z.B. Multicast DNS: `224.0.0.251`	`224.0.0.0`	`224.0.0.255`
Internetwork Control: Globale Multicasting-Protokolle z.B. Network Time Protocol: `224.0.1.1`	`224.0.1.0`	`224.0.1.255`

AD-HOC I-III:
Globale Multicast-Gruppen
z.B. *Walt Disney Company*:
`224.0.19.0` - `224.0.19.063`
Deutsche Börse:
`224.0.46.0` - `224.0.50.255`
	`224.0.2.0`	`224.0.255.255`

	`224.3.0.0`	`224.4.255.255`
	`233.252.0.0`	`233.255.255.255`
Private Multicast Domains: Multicast in privaten Netzwerken	`239.0.0.0`	`239.255.255.255`

Tbl. 4.4 *IPv4 Multicast-Adressbereiche*

Zur Verwaltung einer Multicast-Gruppe wird das *Internet Group Management Protocol* (IGMP) verwendet. Dieses kennt im Wesentlichen die beiden Mechanismen „Mitgliedschaft anmelden“, und „Mitgliedschaft beenden“. Diese Anfragen in Form von IGMP-Paketen werden dabei in IP-Pakete gekapselt.

Besondere IPv4-Adressen

Neben der Adressierung von einzelnen Hosts ist es auch möglich ganze Netzwerke durch IP-Adressen, oder alle in einem Netzwerk befindlichen Hosts (broadcast) zu adressieren. Des Weiteren werden bestimmte IP-Adressen zur relativen Adressierung, beispielsweise des aktuellen Netzwerkes oder des aktuellen Hosts verwendet. Diese sind:

- **Netzwerk**

  IP-Adressen bei denen alle Bits der Hostadresse auf `0` gesetzt sind bezeichnen das Netzwerk mit der entsprechenden Netzwerkadresse. Beispiel: Die IP-Adresse `192.168.1.0/24` bezeichnet das Klasse C Netzwerk mit der Netzwerkadresse `192.168.1`.

- **Lokales Netzwerk**

  IP-Adressen bei denen alle Bits der Netzwerkadresse auf 0 gesetzt sind bezeichnen das aktuelle Netzwerk.

- **Broadcast**

  IP-Adressen bei denen alle Bits der Hostadresse auf 1 gesetzt sind bezeichnen die Broadcastadresse eines Netzwerkes. Sie ist somit die höchste einem Netzwerk zugeordnete IP-Adresse. Diese wird verwendet, um alle Host in diesem Netzwerk durch eine einzige IP-Adresse zu adressieren. Broadcasts dieser Form werden als *Directed Broadcasts* (engl.: gesteuerter Broadcast) bezeichnet.

- **Lokaler Broadcast**

  Die IP-Adresse `255.255.255.255` adressiert alle Hosts des lokalen Netzwerkes. Solche Broadcasts werden als *Limited Broadcast* (engl.: eingeschränkter Broadcast) bezeichnet und beispielsweise von DHCP verwendet.

- **Local Host**

  IP-Adressen aus dem Adressraum von `127.0.0.1` bis `127.255.255.254`, also solche deren erstes Byte den dezimalen Wert 127 hat umfassen das sogenannte Loopback-Netzwerk (dt.: Schleifenschaltung). Hierbei werden alle Pakete die an eine Adresse aus diesem Bereich gesendet werden, ohne den Host zu verlassen an sich selbst weitergeleitet. Daher werden Adressen aus diesem Bereich als *local host* Adressen bezeichnet.

4.2.3 Address Resolution Protocol (ARP)

Die Schnittstelle zwischen IPv4 und dem Link-Layer der Netzzugangsschicht erfolgt mittels des *Address Resolution Protocols* (ARP). Dieses ermittelt zu einer IP-Adresse die zugehörige MAC-Adresse und speichert es anschießend in einer Tabelle (ARP-Tabelle). Um die MAC-Adresse zu ermitteln wird in der Link-Layer Broadcastdomäne eine Anfrage (ARP-Request) durchgeführt, wobei die gesuchte IP-Adresse als Empfängeradresse und die eigene IP-Adresse als Sender-Adresse eingetragen wird. Ein Host der einen ARP-Request erhält dessen Empfängeradresse seiner eigenen entspricht antwortet dem Sender mit seiner MAC-Adresse. Kann zu einer IP-Adresse keine MAC-Adresse ermittelt werden, so wird der ursprüngliche Sender der IP-Pakete benachrichtigt und diese verworfen.

4.2.4 Internet Protocol Version 6 (IPv6)

Nachdem man Anfang der 90er Jahre auf das exponentielle Wachstum des Internet und somit der Zunahme registrierter IP-Adressen aufmerksam wurde erkannte man zwei grundsätzliche Probleme, die sich aus der Verwendung von IPv4 ergaben:

- Durch die großzügige Vergabe von Adressräumen mittels Netzklassen wurde der Adressraum zu schnell aufgebraucht, so dass sich eine Knappheit an IPv4-Adressen anbahnen würde. Dieses Problem konnte aber weitestgehend durch Classless Internet Domain Routing, sowie der Einrichtung privater Adressbereiche und NAT umgangen werden.

- Die Routingtabellen umfassten immer mehr Einträge, wodurch bei jeder Weiterleitung aufgrund der längeren Suche innerhalb der Tabelle Verzögerungen entstehen, die schließlich zu immer größeren Latenzzeiten bei der Datenübertragung führen.

Daher begann die *Internet Engineering Task Force* (IETF), eine Organisation zur Weiterentwicklung das Internet 1995 an der Arbeit eines Nachfolgers für IPv4. Schließlich wurde 1998 das Internet Protocol Version 6 (IPv6) (früher *Internet Protocol next Generation*, IPnG) als neuer Internetstandard verabschiedet.

Notation und Aufbau von IPv6-Adressen

IPv6-Adressen haben eine Länge von 128-Bit. Da bei einer Schreibweise ähnlich IPv4 sehr lange (16 aufeinander folgende Blöcke) und somit unhandliche Adressen entstehen würden, wird bei IPv6-Adressen eine Verkürzung der Notation angewandt:

1. IPv6-Adressen verwendet eine hexadezimale Notation. Dabei werden die 128-Bit in acht 16-Bit große Blöcke unterteilt, die durch einen Doppelpunkt voneinander getrennt notiert werden.

2. Innerhalb eines Blockes werden führende Nullen weggelassen.

3. Die längste Aneinanderreihung von Blöcken die den Wert 0 haben wird durch zwei Doppelpunkte (`::`) ersetzt.

Bsp. 4.2 (IPv6-Adresse)
Hexadezimalschreibweise: `2001:0470:1f06:011f:0000:0000:0000:0002.`
Weglassen führender Nullen : `2001:470:1f06:11f:0:0:0:2`
Ersetzen der längsten Aneinanderreihung von Nullen: `2001:470:1f06:11f::2`

Diese Schreibweise wird als colon hexadecimal Notation bezeichnet. Um umgekehrt nun aus einer solchen IPv6-Adresse wieder die 128-Bit zu rekonstruieren, werden die „::“ (falls vorhanden) durch die fehlende Anzahl von Blöcken mit dem Wert „0“ ersetzt und schließlich die einzelnen Blöcke um führende Nullen zu 4 hexadezimalen Ziffern ergänzt.

Zur Angabe von Adressbereichen wird in IPv6 die CIDR-Notation verwendet. Beispiel: `2001::/16` umfasst alle IPv6 Adressen, deren erster Block den hexadezimalen Wert 2001 hat.

IPv6-Adressen werden durch die ersten Bits der Netzwerkadresse klassifiziert. Diese definieren sog. Adress-Typen (Tbl. 4.5).

Präfix (binär)	IPv6 Adr.	Adress-Typ / Beschreibung
0000 0000	0::/8	Reserviert für spezielle Adressierungen. u.a.: 0:0:0:0:0:0:0:0 die „unspezifizierte Adresse“ 0:0:0:0:0:0:0:1 die local host Adresse
0000 001	200::/7	Reservierter Bereich zur Integration von NSAP-Adressen (engl.: Network Service Access Point)
0000 010	400::/7	Reservierter Bereich zur Integration von IPX-Adressen (engl.: Internetwork Packet eXchange)
001	2000::/3	*Global Scope Unicast*: Global gültige aggregierbare (bündelbare) Unicast-Adressen
1111 1110 10	FE80::/10	*Link-Local Unicast*: Private Unicast-Adressen mit Gültigkeitsbereich innerhalb einer Link-Layer Broadcastdomäne
1111 1110 11	FEC0::/10	*Site-Local Unicast*: Private Unicast-Adressen mit Gültigkeitsbereich innerhalb eines privaten Netzwerkes (Def. 4.1)
1111 1111	FE00::/8	Multicast-Adressen

Tbl. 4.5 *IPv6 Adress-Typen*

Dabei werden die Verbindungsformen Unicast, Anycast und Multicast. Broadcasts entfallen und werden in der Funktion durch sogenannte All-Nodes Multicast-Adressen ersetzt. Aggregierbare (bündelbare) Unicast-Adressen bieten die Möglichkeit der Bündelung mehrerer Knoten zu einer Anycast-Gruppe.

Des Weiteren werden Adressen nach ihrer Gültigkeitsbereiche (Scope) unterschieden. Diese bezeichnet den Bereich einer Adresse innerhalb welcher diese eindeutig ist und führt somit das Konzept der privaten Netzwerke von IPv4 fort. Dabei werden IP-Pakete nur innerhalb ihrer Bereiche weitergeleitet. IPv6 verwendet hierfür folgende Klassifikation:

- *Node-local Scope*: Bereich eines Hosts oder einer Anycast-Gruppe (zur Adressierung von Netzwerkschnittstellen). Wird nur in Verbindung mit Multicast-Adressen verwendet.

- *Link-local Scope*: Bereich einer Link-Layer Broadcastdomänen (OSI-Schicht 2)

- *Site-local Scope*: Bereich eines privaten Netzwerkes (entsprechen funktional den privaten IPv4-Adressen)

- *Organisation-local Scope*: Bereich innerhalb einer LIR (z.B. ISP). Wird nur in Verbindung mit Multicast-Adressen verwendet.

- *Global Scope*: Global eindeutige IP-Adressen (entsprechen funktional den öffentlichen IPv4- Adressen)

Unicast-Adressen

IPv6-Unicast-Adressen beziehen sich auf Netzwerk-Schnittstellen (Interfaces). Einer Schnittstelle können dabei mehrere Adressen beliebigen Typs und umgekehrt einer Adresse mehrere Schnittstellen zugeordnet werden (Anycast-Gruppe).

Hierfür umfassen Unicast-Adressen zwei Teiladressen. Dies ist die aus IPv4 bekannte Netzwerkadresse (engl.: Network-ID), welche jedoch im Gegensatz zu IPv4 immer die feste Länge von 64-Bit hat, und die ebenfalls 64-Bit lange Interfaceadresse (engl.: Interface-ID) die zur eindeutigen Identifikation der Netzwerkschnittstelle dient. Verschiedene IPv6-Adressen (z.B. eine globale und eine lokale) mit der gleichen Interface-ID beziehen somit sich auf die gleiche Netzwerkschnittstelle.

Da auf diese Art und Weise eine eindeutige Zuordnung von IP-Paketen zu Interfaces und somit zu einzelnen Personen möglich ist, wurden die *Privacy Extensions* entwickelt. Dabei wird die Interface-ID innerhalb des aktuellen Subnetzes durch eine freie (zufällig gewählte) Interface-ID ersetzt. Um schließlich zu verhindern, dass Einzelpersonen in kleineren Netzwerken (z.B. Privatpersonen) über die Network-ID identifiziert werden können, wird dieses Verfahren bei der Einwahl in den ISP auch für die Network-ID durchgeführt.

Bei Unicast-Adressen werden ausschließlich die Gültigkeitsbereiche „Link-local“, „Site-local“ und „Global“ verwendet.

Link-local Unicast-Adressen

Link-local Unicast-Adressen ermöglichen die automatische Selbstkonfiguration von Hosts. Hierfür wird mittels der Protokolle *Neighbor Discovery Protocol* (NDP) und *Stateless Address Autoconfiguration* (SAA) innerhalb der Link-Layer Broadcastdomäne vom nächstgelegenen Router eine Adresse bezogen. Diese ist entweder eine Site-local Unicast-Adresse oder eine globale Unicast-Adresse und wird von Router anhand der MAC-Adresse ermittelt. Diese Mechanismen ersetzen damit das Dynamic Host Configuration Protocol (DHCP).

Site-local Unicast-Adressen

Site-local Unicast-Adressen dienen zur Vergabe eindeutiger Adressen innerhalb eines privaten Netzwerkes. Im Gegensatz zu Link-Local-Unicast Adressen müssen Site-local Adressen manuell konfiguriert werden.

Globale Unicast-Adressen

Die globalen (aggregierbaren) Unicast-Adressen sind Internet weit eindeutig und ersetzen die öffentlichen IPv4-Adressen. Um mittels einer solchen Adresse flächendeckende Dienste anbieten zu können sind sie bündelbar, so dass sie als globale Anycast-Adressen verwendet werden können.

Zugunsten eines effizienten Routings teilt sich die Netzwerkadresse in einen hierarchisch aufgebauten *Routingpräfix* (engl.: *global routing prefix*), eine *Subnet-ID* (Abb. 4.3) und eine *Interface-ID*. Der Routingpräfix dient der Zuweisung einer Organisation oder eines Netzwerkes und umfasst zum Beispiel Identifikatoren für die RIRs (für APNIC), LIRs oder ISPs. Der Aufbau hierfür kann ähnlich einer Postleitzahl gesehen werden. Die Subnet-ID wird verwendet um innerhalb einer Organisation das Netzwerk zu strukturieren. Die Interface-ID schließlich identifiziert die Schnittstelle eines Hosts anhand ihrer Hardware-Adresse. Diese ist im *EUI-64* Format, bei dem die 48-Bit lange MAC-Adresse in die 64-Bit lange Interface-ID abgebildet wird.

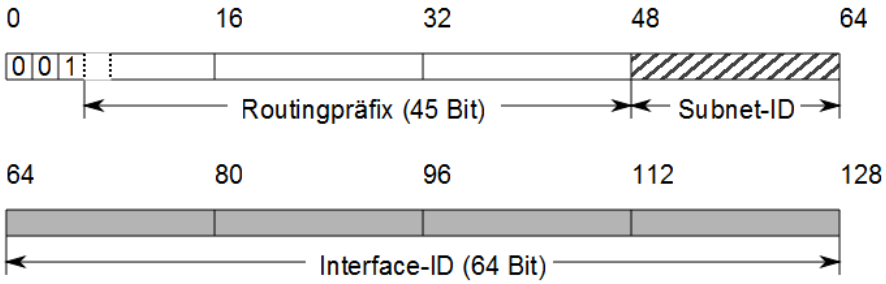

Abb. 4.3 *Strukturierung einer globalen Unicast-Adresse*

Innerhalb des Adressbereiches der globalen Unicast-Adressen (`2000::/3`) werden einige Teilbereiche für spezielle Anwendungen insbesondere für den Übergang von IPv4 zu IPv6 verwendet (Tbl. 4.6).

IPv6 Adr.	Beschreibung
`2001::/16`	Adressen die durch RIRs/LIRs verwaltet werden
`2002::/16`	Adressen um mittels des sog. *6to4*-Mechanismus IPv6 Pakete über ein IPv4 Netzwerk zu übertragen
`3FFE::/16`	Adressen eines früheren IPv6 Testnetzwerkes, dem sog. *6Bone*

Tbl. 4.6 *Reservierte globale Unicast-Adressen*

Multicasting

Ähnlich wie bei globalen Unicast-Adressen teilen sich auch Multicast-Adressen in mehrere Teiladressen auf. Diese umfassen einen Bereich für Adresseigenschaften (Flags), einen Identifikator für den Gültigkeitsbereich (Scope-ID) und die Multicast Group-ID, welches die Multicast-Gruppe identifiziert (Abb. 4.4).

Um Gegensatz zu globalen Unicast Adressen werden bei Multicast-Adressen alle Gültigkeitsbereiche von Node-local bis global verwendet. Zudem wird innerhalb der Adresseigenschaften (Flags) festgelegt, ob sich die Bedeutung einer Multicast-Gruppe über alle Gültigkeitsbereiche (engl.: permanent) erstreckt, oder nur auf einen bestimmten (engl.: transient). Beispielsweise werden für das Network Time Protocol (NTP) permanente Multicast-Gruppen verwendet. Ein Absender eines Paketes an eine solche NTP Multicast-Gruppe spricht dabei alle NTP-Server an, die im gleichen Gültigkeitsbereich wir die der Absenderadresse sind.

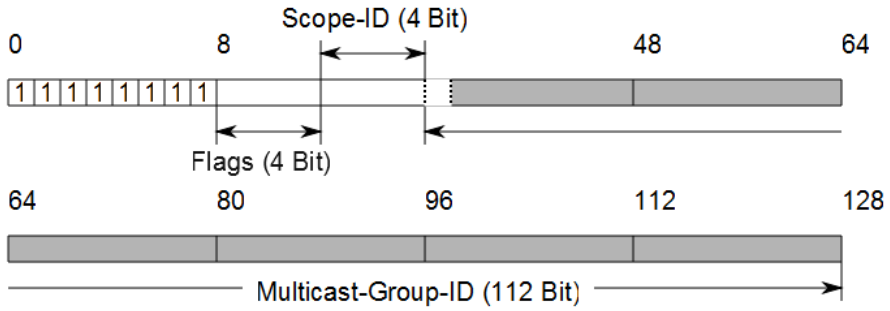

Abb. 4.4 *Struktur einer Multicast-Adresse*

Einige Multicast-Adressen wurden für spezielle Zwecke reserviert, z.B. um die die Funktion der in IPv4 verwendeten Broadcast-Adressen zu ersetzen (Tbl. 4.7). Ein weiterer wichtiger Typ von Multicast-Adressen wird als Solicited-Node Addresse bezeichnet. Diese haben den Adressbereich `FF02::1:FF00:0/104`. Dabei werden die letzten 24-Bit, der 128-Bit Adresse um die letzten 24-Bit einer Unicast-Adresse ergänzt. Sie werden verwendet, um innerhalb einer Link- Layer Broadcastdomäne mit einem Knoten zu kommunizieren, ohne die Netzwerksdresse zu kennen. Eine Solicited-Node Addresse spricht somit alle Hosts innerhalb einer Link-Layer Broadcastdomäne an, deren IPv6-Adressen in den letzten 24-Bits mit ihr übereinstimmen. Diese Adressen werden u.a. vom NDP verwendet.

Autokonfiguration von IPv6-Adressen

In IPv6-Netzwerken werden zwei Mechanismen zur automatischen Konfiguration von Hosts verwendet: Die zustandsbehaftete und die zustandslose Konfiguration:

Die *zustandsbehaftete Konfiguration* (engl.: Statefull Address Autoconfiguration) bedient sich des Protokolls DHCPv6 und entspricht in seiner Funktionalität der Autokonfiguration von IPv4-Adressen. Insbesondere können hierbei an zentraler Stelle (DHCP-Server) feste IP-Adressen für Hosts (DHCP-Clients) vergeben werden.

IPv6 Adr.	Beschreibung
`FF0x::`	Alle Multicast-Adressen mit der Multicast Group-ID „0“ sind reserviert
`FF01::1`	(Node Scope) Alle Interfaces des Hosts
`FF02::1`	(Link-local Scope) Alle Hosts (engl.: All Nodes) in der Link-Layer Broadcastdomäne. Wird u.a. vom NDP zur Ermittlung einer Link-local Adresse eines Hosts verwendet.
`FF01::2`	(Node Scope)
`FF02::2`	(Link-local Scope) Alle Router in der Link-Layer Broadcastdomäne. Wird u.a. vom NDP zur Ermittlung der Router verwendet.
`FF05::2`	(Site-local Scope) Alle Router im privaten Netzwerk

Tbl. 4.7 *Reservierte Multicast-Adressen*

Die *zustandslose Konfiguration* (engl.: *Stateless Address Autoconfiguration*, SAA) ermöglicht es ohne zentrale Konfiguration (Server) eine automatische Adresszuordnung zu ermöglichen. Hierfür werden mit Hilfe des Neighbor Discovery Protocols alle notwendigen Informationen ermittelt.

Neighbor Discovery Protocol (NDP)

In IPv6 Netzwerken übernimmt das *Neighbor Discovery Protocol* (NDP) die Aufgaben des Adress Resolution Protocols (ARP), sowie einiger Aufgaben des Internet Control Message Protocol (ICMP). Diese Aufgaben umfassen u.a.:

- Automatische Erkennung der Parameter zur Übertragung von Link-Layer Nachrichten
- Ermittlung aller verfügbaren Router innerhalb der Link-Layer Broadcastdomäne
- Ermittlung des Adress-Präfixes, sowie der Link-Layer-Adresse eines Hosts innerhalb der Link-Layer Broadcastdomäne
- Überprüfung, ob eine IP-Adresse bereits vergeben ist
- Überprüfung, ob ein bestimmter Host innerhalb der Link-Layer Broadcastdomäne verfügbar ist

4.2.5 Migration von IPv4 zu IPv6

Als Basis aller höheren Protokolle hat das Internetprotokoll innerhalb der TCP/IP-Familie eine zentrale Bedeutung. Die teilweise grundlegenden Unterschiede zwischen IPv4 und IPv6 erschweren dabei eine einfache Umstellung. Da diese in der Vermittlungsschicht des OSI-Modells angesiedelt sind, betrifft dies neben den einzelnen Hosts auch und insbesondere die Routinginfrastruktur. Eine schnelle Umstellungsphase ist daher nicht möglich. Um dennoch eine reibungslose Umstellung zu ermöglichen, wurden im Laufe der Zeit verschiedene Mechanismen entwickelt und eingeführt.

Dual-Stack Architekturen

Häufig sind innerhalb eines Knotens beide TCP/IP Protokollstapel (IPv4 und IPv6) implementiert und werden parallel als sog. *Dual-Stack* betrieben. Den einzelnen Schnittstellen werden dann sowohl IPv4-Adressen, als auch IPv6-Adressen zugewiesen. Dieses Verfahren ermöglicht es über beide Protokolle unabhängig voneinander zu kommunizieren.

IPv6-Tunnel in IPv4-Netzwerken

Für eine IPv6-Verbindung zwischen zwei Hosts ist es notwendig, dass alle Zwischenknoten auf dem Übertragungsweg das IPv6 (z.B. durch eine Dual-Stack Architektur) unterstützen. Da dies aber nicht immer gegeben ist, besteht die Notwendigkeit nach einer Methode zur Überbrückung von IPv4-Netzwerken.

Die Überbrückung von Routern (oder Netzwerken), die keine IPv6 Unterstützung bieten, erfolgt durch sogenannte *Tunnel*. Hierfür werden Knoten mit Dual-Stack Architektur, die sowohl mit dem IPv6-Netzwerk, als auch mit dem IPv4-Netzwerk verbunden sind als Tunnelstellen (engl.: Gateways) eingesetzt. Diese kapseln IPv6-Pakete in IPv4-Pakete bzw. entkapseln sie wieder.

Die Umsetzung einer getunnelten Nachrichtenübertragung kann nun in zwei Varianten erfolgen: Durch *konfigurierte Tunnel* und durch *automatische Tunnel*. Ein konfigurierter Tunnel erfordert die Bekanntgabe aller nötigen Informationen an die Gateways, welche zum Beispiel die eigene Adresse sowie die entsprechende Gegenstelle des Tunnels umfassen. Bei einem automatischen Tunnel entnehmen die Gateways die benötigten Informationen direkt aus den IP-Paketen.

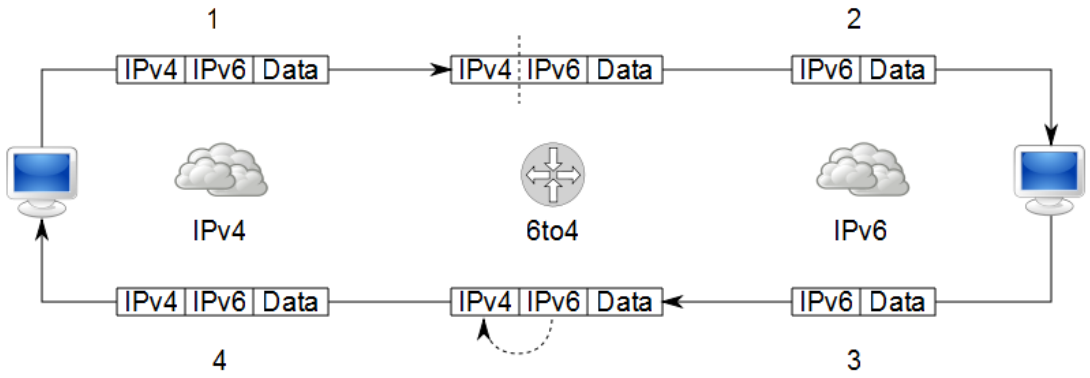

Abb. 4.5 *"6to4"-Tunnel*

Ausgehend von einem Host, der IPv6-Pakete über ein IPv4-Netzwerk übertragen möchte ergibt sich aufgrund der fehlenden Verbindung zum IPv6-Internet das Problem, das dem Sender keine globale Unicast Adresse zugewiesen werden kann. IPv6 umfasst hierfür einen Mechanismus, der automatische Tunnelung ermöglicht. Dieser 6to4 genannte Mechanismus umfasst mehrere Schritte (Abb. 4.5):

1. **IPv6 in IPv4 Kapselung**

   Hierfür wird die eigene 32-Bit lange öffentliche IPv4-Adresse an den 16-Bit langen Präfix `2002::/16` angehängt, wodurch sich die globale Eindeutigkeit der öffentlichen IPv4-Adresse auf ein ganzes `/48` IPv6-Subnet überträgt. (Bsp. 4.3)

   Diese IPv6-Adresse wird nun in den IPv6-Paketen als Absender eingetragen und die IPv6 Adresse des Empfängers übernommen. Anschließend wird das IPv6-Paket in ein IPv4-Paket gekapselt, welches an einen 6to4-Gateway geschickt wird.

2. **Entkapselung und Weiterleitung in IPv6**

   Der 6to4-Gateway entkapselt die IPv6-Pakete und schickt diese im IPv6-Netzwerk an den Empfänger weiter

3. **Antwort und Rücksendung**

   Der Empfänger schickt seine Antwort an die Absenderadresse (die automatisch an einen beliebigen 6to-Gateway geleitet wird)

4. **IPv6 in IPv4 Kapselung**

   Der 6to4-Gateway generiert aus der neuen Empfängeradresse die ursprüngliche öffentliche IPv4-Adresse und schickt das IPv4-Paket an diese.

Dieses Verfahren hat jedoch einige Nachteile. Da nun beim Routing immer die entsprechenden Tunnelstellen berücksichtigt werden müssen ist eine optimale Wegsteuerung nicht mehr gewährleistet. Hinzu kommt, dass sich im IPv4-Netzwerk die Datenlast aufgrund der zusätzlichen Header erhöht. Um nicht die MTU zu überschreiten müssen entweder kleinere Pakete verschickt werden (wodurch sich die Last an zusätzlichen Headern pro Nutzdaten zusätzlich erhöht), oder bei der Einkapselung die inneren IP-Pakete aufgeteilt werden.

Bsp. 4.3 (Ermittlung einer IPv6-Adresse zu einer IPv4-Adresse)
Zur Berechnung werden die IPv4-Blöcke in je zwei hexadezimale Ziffern umgewandelt:
IPv4: `192.0.32.8` *Umwandlung:* `192` → `0xC0`, `0` → `0x00`, `32` → `0x20`, `8` → `0x08`
IPv6: `2002:C000:2008::/48`

Adressübersetzung zwischen IPv4 und IPv6-Netzwerken

Ein weiteres Verfahren um IPv6-Pakete über ein IPv4-Netzwerk zu übertragen und auch umgekehrt, bieten Übersetzungsverfahren (engl.: Translator). Dabei wird eine Verbindung zu einem Gateway aufgebaut, welches eine eindeutige Zuordnung zwischen der aktuellen Adresse und einer entsprechend dem anderen Protokoll vornimmt. Dieser Gateway über-

nimmt dann während dem gesamten Kommunikationsablauf die Übersetzung der Adressen, sowie die Umsetzung der Header.

4.3 Transportschicht

4.3.1 Interprozesskommunikation in TCP/IP Netzwerken

Die nächsthöhere Schicht über der Internetschicht ist die Transportschicht. Diese ermöglicht die Kommunikation zwischen einzelnen Prozessen auf gleichen oder auch auf verschiedenen Hosts. Hierfür werden die Prozesse innerhalb eines Hosts durch *Portnummern* (nach dem OSI-Modell auch „OSI Schicht-4 Adressen“) adressiert. Genauso wie die Netzwerkadressen werden die Portnummern von der IANA zentral organisiert.

Notation und Aufbau von Portnummern

Portnummern haben eine Länge von 16-Bit und werden dezimal notiert. Der Adressbereich von umfasst also die Adressen von 0 bis 65635. In Verbindung mit einer IP-Adresse können mittels einer Portnummer einzelne Prozesse auf einzelnen Hosts im Internet adressiert werden. Hierfür werden IP-Adresse und Portnummer durch einen Doppelpunkt voneinander getrennt dargestellt. Da der Doppelpunkt in IPv6-Adressen jedoch bereits die hexadezimalen Blöcke voneinander trennt, werden diese deshalb in eckige Klammern eingeschlossen. (Bsp. 4.4)

Von der IANA werden die Portnummern in folgende Kategorien eingeteilt:

- **Privilegierte Portnummern** (engl.: *Well Known Ports*)

  Diese haben den Adressbereich von 0 bis 1023 und werden von der IANA für privilegierte Prozesse wie DHCP, DNS oder NTP vergeben.

- **Registrierte Portnummern** (engl.: *Registered Ports*)

  Registrierte Portnummern umfassen den Adressbereich von 1024 bis 49151. Anwendungshersteller können hier bei Bedarf Portnummern für eigene Protokolle bei der IANA registrieren lassen. Dadurch ist die Portnummer eines solchen Prozesses schon bei der Verbindung zwischen zwei Hosts bekannt. Beispiele: Instant-Messaging (ICQ), Chatserver (IRC) oder Datenbankserver

- **Private Portnummern** (engl.: *Private Ports*)

  Diese verfügen über den Adressbereich von 49152 bis 65535 und werden zumeist für Clientprozesse verwendet. Für die Kommunikation über private Portnummern müssen diese bei der Gegenstelle bekannt gegeben werden.

Bsp. 4.4 (Portnummer)
Adressierung des Port 80 (HTTP) auf verschiedenen Hosts in IPv4 und IPv6 Netzwerken:
IPv4: `192.0.32.8:80`, IPv6: `[2002:C000:2008::1]:80`

Sockets

Insbesondere bei Serverprozessen erfolgt die Kommunikation häufig parallel über mehrere Kommunikationspartner gleichzeitig. Für eine eindeutige Zuordnung der Datenpakete zu einer bestimmten Kommunikation sind somit sowohl die Adressinformationen (IP, Port) für den Sender, als auch den Empfänger, sowie Information über das verwendete Protokoll der Transportschicht notwendig. Da bezüglich der Kommunikation mit einem Prozess immer das verwendete Protokoll angegeben wird, können einzelne Ports für verschiedene Protokolle gleichzeitig benutzt werden.

Um die Verwaltung der für die Kommunikation benötigten Informationen zu erleichtern wurde das Konzept sog. *Sockets* (dt.: Sockel, Steckverbindung) eingeführt. Will ein lokaler Prozess nun eine Verbindung mit einem entfernten Prozess aufnehmen, so fordert er beim Betriebssystem einen Socket an (Allokation) und übergibt diesem die für die Verbindung benötigten Informationen (Initialisierung).

Das Betriebssystem verwaltet die Verbindungs-Informationen und ordnet sie einer (innerhalb des lokalen Systems eindeutigen) Socket-Adresse (engl.: Socket-ID) zu. Über diese können Protokolle der Transportschicht mit anderen Prozessen über sog. virtuelle Verbindungen kommunizieren. Virtuelle Verbindungen sind dabei unabhängig von der physikalischen Verbindung und werden Beispielsweise auch bei einer Unterbrechung dieser aufrechterhalten.

4.3.2 Transmission Control Protocol (TCP)

Das *Transmission Control Protocol* (TCP) ist ein verbindungsorientiertes, zuverlässiges (Rückwärtsfehlerkorrektur) Protokoll der Transportschicht. Mit dem Ziel eines Transportprotokolls, welches beliebige Datenströme verlustfrei und fehlerfrei bidirektional übertragen kann, wurde nach fast acht Jahren Entwicklungszeit 1981 TCP als Internetstandard eingeführt.

Die wichtigsten Mechanismen von TCP umfassen:

- **Zuverlässige Übertragung**

  Übertragungskontrolle durch Empfangsbestätigung mittels einer Quittung

- **Verbindungsorientierte Vermittlung**

  Kommunikation mittels einer logischen Verbindung, um Steuerinformationen auszutauschen. Der Auf- und Abbau erfolgt dabei über einen sog. *Drei-Wege-Handschlag*.

- **Stromorientierte Übertragung**

  Datagrammdienst zur Verarbeitung von Datenströmen

Zur Umsetzung dieser Mechanismen werden im TCP-Headers verschiedene Steuerinformationen übertragen (Tbl. 4.8).

Feld(er)	Beschreibung
Adressen	Portnummern des Senders und des Empfängers
Flags	Mittels binärer Variablen, sog. *Flags* werden bestimmte Eigenschaften von Datensegmenten gekennzeichnet
SequenzNr (SN)	Diese 32-Bit lange Zahl gibt bei einer Datenübermittlung die Byte-Position eines Datensegments innerhalb eines Datenstromes an. Haben die Datensegmente beispielsweise eine Größe von 512 Bytes, so ist die Sequenznummer eines Datensegmentes jeweils um 512 größer als die des vorhergehenden. Wird dabei der Höchstwert von 2^{32} überschritten, so wird bei 0 weiter gezählt. Diese Nummer dient dem Empfänger um die Datensegmente in der richtigen Reihenfolge zusammenzusetzen und den Datenstrom auf fehlende Datensegmente zu überprüfen
BestätigungsNr (BN)	Gibt die Sequenznummer des nächsten Datensegments an, dessen Empfang quittiert wird. Diese entspricht der Byteposition, bis zu welcher die Daten vollständig übertragen wurden + 1. Beispiel: Das letzte empfangene Datensegment hat die Länge 32 Byte und die SN 1000, dann ist die BN 1033
Fenstergröße	Maximale Datenmenge die vom Sender zum Empfänger versandt wird ohne, dass dieser den Empfang bestätigt

Tbl. 4.8 *Informationen im TCP-Header*

Zur Erklärung wird im Folgenden eine Kommunikation zwischen den zwei Partnern Host A und Host B betrachtet, wobei Host A der Initiator des Verbindungsaufbaus, sowie des Verbindungsabbaus ist:

Phase 1: Verbindungsaufbau

Der Verbindungsaufbau erfolgt mittels des sog. Drei-Wege-Handschlags (Abb. 4.6)

1. **Host A schickt Verbindungsanfrage an Host B**

   Mit den vollständigen Absender- und Empfänger-Adressen erzeugt Host A einen Socket. Daraufhin generiert Host A eine zufällige Sequenznummer x, die sog. *Initial Sequence Number* (ISN) und schickt diese in einem Datensegment an Host B. Dabei ist das SYN-FLAG (v. engl.: Synchronize „Synchronisieren“) gesetzt, welches die Absicht zum Verbindungsaufbau zu einem Prozess repräsentiert.

2. **Host B schickt eine Quittung an Host A**

   Nach dem Empfang dieses Datensegmentes überprüft Host B, ob der aufgeforderte Prozess im Annahmemodus (listen) ist und informiert seinerseits Host A mittels eines zweiten Datensegments. Eine positive Bestätigung wird dabei mittels eines gesetzten ACK-FLAG (v. engl.: acknowledgment „Bestätigung“), sowie der Bestätigungsnummer x+1 repräsentiert. In diesem Falle generiert Host B ebenfalls eine eigene Sequenznummer y und fügt diese dem Datensegment bei.

3. **Host A schickt eine Quittung der Quittung an Host B**

   Schließlich schickt Host A nach dem Empfang des Datensegments ein abschließendes Datensegment, welches ein gesetztes ACK-FLAG, sowie die Sequenznummer x+1, sowie die Bestätigungsnummer y+1 enthält.

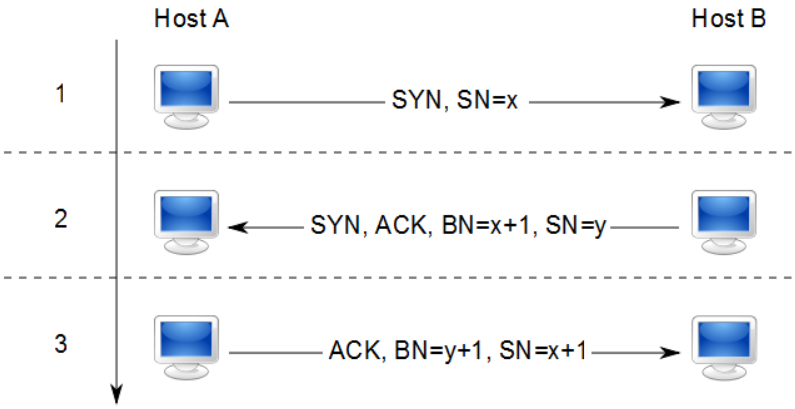

Abb. 4.6 *Drei-Wege-Handschlag*

Phase 2: Datenübertragung

Die Grundlage von TCP bildet das Konzept der *Übertragungskontrolle durch Empfangsbestätigung* (engl.: Positive Acknowledgement with Retransmission, PAR):

Dabei wird für jedes verschickte Segment ein abwärts zählender Zeitmesser, der sog. *Retransmission Timer* gestartet. Innerhalb dieser Zeitspanne kann der Kommunikationspartner eine Empfangsbestätigung durch ein Segment mit gesetzten ACK-FLAG und der entsprechenden Bestätigungsnummer zurück schicken. Läuft die Zeit ab, ohne das eine Empfangsbestätigung erhalten wurde, so wird das letzte Segment, welches nicht mehr quittiert wurde erneut gesendet.

Dabei wird nicht bei jedem Segment gewartet, bis eine Quittung eingetroffen ist bzw. der Timer abgelaufen ist, da dies ineffizient wäre. Dies ist auch nicht notwendig, da auf der Empfängerseite schließlich immer nur eine Quittung für die Sequenznummer ausgestellt wird, bis zu welcher die Daten vollständig übertragen wurden.

Falls nun mehrere Datensegmente (nicht notwendigerweise in der richtigen Reihenfolge) eintreffen, die jedoch einen zusammenhängenden („lückenlosen“) Datenstrom rekonstruieren lassen, so reicht eine Quittung auf die höchste Sequenznummer. Insbesondere wird

aber auch erst dann eine Quittung ausgestellt, wenn ein zusammenhängender Datenabschnitt übertragen wurde (Abb. 4.7).

Angenommen Host A schickt nun Host B immer mehr Datensegmente, obwohl die vorhergehenden noch nicht quittiert wurden (weil z.B. ein Datensegment verloren gegangen ist), so kann es passieren, dass es bei Host B zu einem Überlauf des Empfangspuffers kommt. Aus diesem Grund wird diese Datenmenge durch eine *Fenstergröße* beschränkt. Dieses Verfahren wird *Sliding Window* (zu dt.: Schiebefenster) genannt. Am Beispiel in Abb. 4.7 beträgt die Fenstergröße 4.

TCP optimiert dabei die Fenstergröße und den Retransmission Timer hinsichtlich einer maximalen Datenübertragung. Einerseits erfolgt dies über die Empfangsbestätigungen, über welche der Empfänger dem Sender die Fenstergröße mitteilt, die er wünscht (abhängig vom Puffer). Andererseits misst TCP die Zeitdauer zwischen dem Versenden eines Datensegments und dem Erhalt der Quittung (engl.: *Round Trip Time*, RTT) und reguliert danach die Retransmission Time.

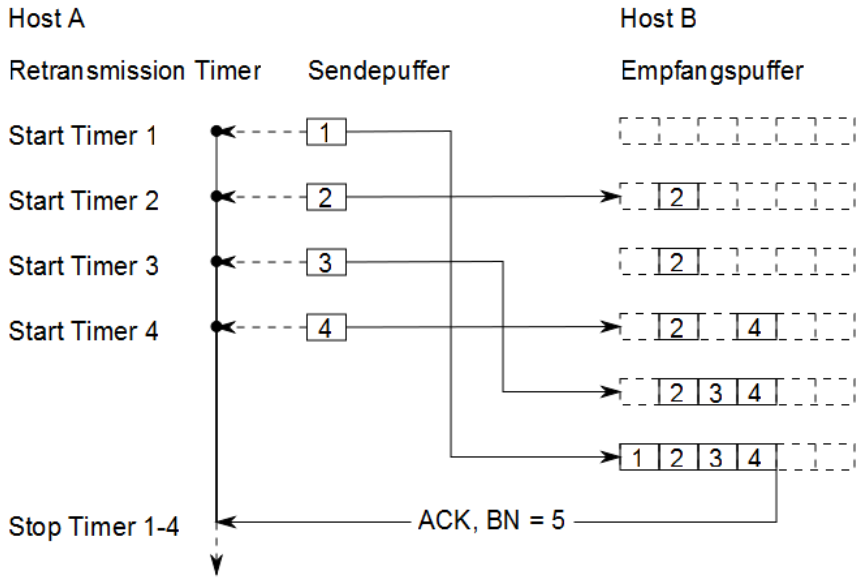

Abb. 4.7 *Übertragungskontrolle in TCP*

Phase 3: Verbindungsabbau

Der Verbindungsabbau erfolgt mittels eines (veränderten) Drei-Wege-Handschlags:

1. **Host A schickt eine Abmeldung an Host B**

   Host A schickt ein Datensegment mit gesetztem FIN-FLAG (engl.: finnish, Abschluss) und der aktuell von Host A verwendeten Sequenznummer x an Host B.

2. **Host B schickt eine Quittung und eine Abmeldung an Host A**

   Host B schickt eine Empfangsbestätigung mit gesetztem ACK-FLAG und der BN x+1 an Host A. Zudem schickt Host B ein Datensegment mit gesetztem FIN-FLAG

und der aktuell von Host B verwendeten Sequenznummer y an Host A. Diese beiden Datensegmente werden dabei je nach Implementierung auch zu einem einzigen zusammengefasst.

3. **Host A schickt eine Quittung an Host B**

   Schließlich schickt Host A eine Empfangsbestätigung mit gesetztem ACK-FLAG und der BN y+1 an HOST B. Nachdem Host A die Bestätigung gesendet hat, wechselt er in einen Wartezustand. Innerhalb dieses Wartezustands werden alle auf dem Socket ankommenden Datensegmente verworfen. Dieses soll verhindern, dass verspätet eintreffende Datensegmente die Kommunikation bei einer neuen Verbindung mit der gleichen Socket-ID stören. Schließlich werden Sende- und Empfangspuffer geleert und die Socket-ID freigegeben.

4.3.3 User Datagram Protocol (UDP)

Nachdem 1973 die Entwicklung von TCP begann, stellte man bereits früh fest, dass dieses für die Übertragung von Sprache ungeeignet ist. Bei dieser Art der Datenübertragung spielen Verluste einzelner Pakete gegenüber einer langen Verzögerung bei der Übertragung eine untergeordnete Rolle. Daher begann man 1977 mit der Entwicklung des verbindungslosen, minimalen *User Datagram Protocols* (UDP) welches z.B. sog. IP-Telefonie ermöglichen sollte.

Feld(er)	Beschreibung
Adressen	Portnummern des Senders und des Empfängers
Länge	Länge der transportierten Nutzdaten

Tbl. 4.9 *Informationen im UDP-Header*

Im Wesentlichen umfasst UDP nur die Adressinformationen in Form der 16-Bit langen Portnummern des Senders und des Empfängers, die Längenangabe der transportierten Nutzdaten, sowie eine Prüfsumme. Im Gegensatz zum darunter liegenden IP werden zur Berechnung der Prüfsumme jedoch die Nutzdaten miteinbezogen, wodurch fehlerhaft übertragene Daten erkannt werden können.

Dieser minimalistische Ansatz macht UDP für Audio- und Videoübertragungen aber auch für Protokolle zur Netzwerkkonfiguration interessant, da hiermit die stetige Netzwerklast gering gehalten werden kann. Beispiele hierfür sind DHCP, DNS und NTP.

4.3.4 Network Adress Translation (NAT)

In den Abschnitten 3.2.1 und 3.2.2 wurden private (bzw. lokale) IP-Adressen eingeführt. Diese stellen einen Ansatz dar, mit der global eindeutige IP-Adressen nur noch für Verbindungen zum Internet und nicht innerhalb privater Netzwerke verwendet werden müssen. Da private Adressen jedoch nicht über die Netzwerkgrenzen geroutet werden und ohnehin

darüber hinaus nicht gültig sind, müssen für Verbindungen zu Hosts außerhalb des privaten Netzwerkes wiederum globale Adressen verwendet werden. Diese werden beim Verbindungsaufbau vom ISP aus dessen Adresspool entnommen und dem Host im privaten Netzwerk für die Zeitdauer der Verbindung zur Verfügung gestellt.

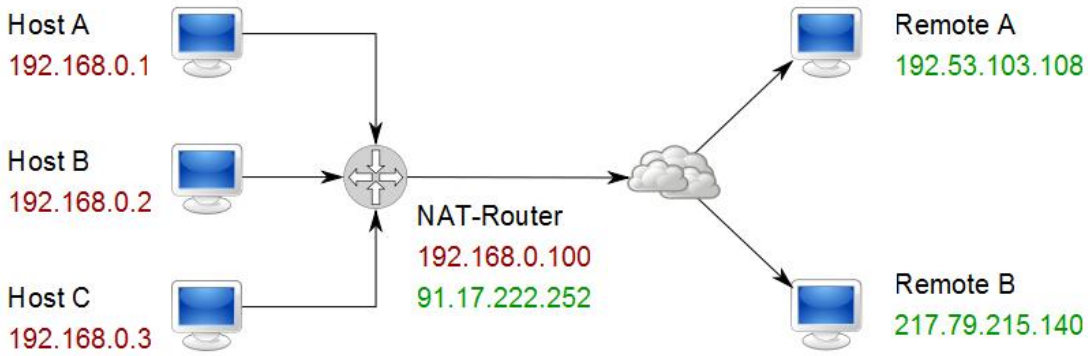

Abb. 4.1 *NAT-Router: rot: lokales Netz, grün: Internet*

Die Zuweisung der globalen Adressen zu der lokalen Adresse in einem Knoten der sowohl mit dem privaten Netzwerk, als auch mit dem Internet verbunden ist. Dieser führt während der Verbindung eine Übersetzung der IP-Adressen durch und leitet sie entsprechend weiter. Dieses Verfahren wird als *Network Adress Translation* (NAT) und der Knoten als *NAT-Router* bezeichnet. Außerhalb des privaten Netzwerkes ist nur der NAT-Router mit seiner globalen Adresse sichtbar. Dabei ist es auch möglich, dass mehrere Hosts des privaten Netzwerkes über den NAT-Router kommunizieren und sich somit eine globale Adresse teilen (Abb. 4.1).

Hierfür ist es notwendig, dass die Beziehung zwischen den lokalen Verbindung und den globalen Verbindungen eindeutig ist. Zu diesem Zweck identifiziert der NAT-Router lokale Verbindungen eindeutig durch die IP-Adressen und Portnummern des lokalen Hosts (Lokale Adresse) und des Hosts im Internet (Remoteadresse). Für diese Verbindung wählt er eine freie Portnummer ab 61000 (private Portnummern) und leitet die Verbindung des Hosts im Internet über diese.

Bei einem Verbindungsaufbau werden die entsprechenden Einträge in einer Verknüpfungstabelle des NAT-Routers hinterlegt und beim Verbindungsabbau wieder aus dieser entfernt (Tbl. 4.10).

Lokales Netzwerk		Internet	
Lokale Adresse	Remoteadresse	Lokale Adresse	Remoteadresse
192.168.0.1 :49587	92.53.103.108 :123	91.17.222.252 :61000	192.53.103.108 :123
192.168.0.1 :49588	217.79.215.140 :80	91.17.222.252 :61001	217.79.215.140 :80
192.168.0.2 :49842	217.79.215.140 :21	91.17.222.252 :61002	217.79.215.140 :21
192.168.0.3 :49121	192.53.103.108 :123	91.17.222.252 :61003	192.53.103.108 :123

Tbl. 4.10 *Beispiel für die Verknüpfungstabelle eines NAT-Routers*

Ein Problem, welches sich jedoch aus der Verwendung von NAT ergibt, ist das von einem Host außerhalb des privaten Netzwerkes keine Verbindung zu einem Host innerhalb des privaten Netzwerkes aufgebaut werden kann.

4.4 Anwendungsschicht

Die Protokolle der Anwendungsschicht dienen als Schnittstelle zwischen dem lokalen Host und dem Netzwerk. Ihre Aufgaben umfassen zum Einen das Bereitstellen von Grundfunktionen wie den Dateitransfer, die von lokalen Anwendungen genutzt werden können, und zum anderen aber auch Netzwerkmanagement und Verwaltungsaufgaben die unabhängig vom Transport (TCP/IP Schicht 1-3) sind.

4.4.1 Dynamic Host Configuration Protocol (DHCP)

Das *Dynamic Host Configuration Protocol* (DHCP) ist ein Client-Server basierendes Netzwerkmanagement Protokoll. Es ermöglicht die zentrale Zuweisung einer Netzwerkkonfiguration der Internetschicht. Dadurch ist es beispielsweise möglich, dass einem Host beim Betreten einer WLAN-Funkzelle automatisch eine freie IP-Adresse und die des zuständigen Routers, sowie die des Nameservers übermittelt wird, ohne dass eine manuelle Konfiguration durch den Anwender notwendig wird.

Hierfür versendet der DHCP-Client per UDP eine DHCP Anfrage, einer sog. DHCPDISCOVER-Nachricht mittels eines Broadcasts im lokalen Netzwerk. Beispielsweise in IPv4 hat der Absender dieser Anfrage dann die Adresse `0.0.0.0` und der Empfänger `255.255.255.255`. Erhält nun ein DHCP-Server eine solche Nachricht, so entnimmt er den Datenframes die MAC-Adresse des Senders und ermittelt anhand einer Tabelle, wie

mit dieser MAC-Adresse zu verfahren ist. Zu diesem Zweck wird beim DHCP-Server ein Pool mit IP-Adressen hinterlegt, die anhand festgelegter Reglements vergeben werden.

DHCP kennt drei Arten der Adressvergabe:

- **Manuelle statische Adressvergabe**

  Einer bestimmten MAC-Adresse wird eine feste IP-Adresse zugeordnet. Dies erfolgt mittels eines manuellen tabellarischen Eintrags. Damit ist es z.B. möglich einzelne Serverprozesse mit einer festen IP-Adresse zu verknüpfen.

- **Automatische statische Adressvergabe**

  Aus einem Adresspool wird eine freie IP-Adresse gewählt und dem Client zugewiesen. Die Zuordnung wird in einem tabellarischen Eintrag automatisch festgehalten, so dass sie zukünftig ausschließlich an diesen Client erfolgt.

- **Dynamische Adressvergabe**

  Bei einer Anfrage wird aus einem Adresspool eine freie IP-Adresse gewählt und dem Client zugewiesen. Diese Zuweisung erfolgt zeitlich beschränkt. Hierfür wird eine sog. Lease- Time gesetzt, vor deren Ablauf der Client mittels einer DHCPREQUEST-Nachricht eine Verlängerung beantragen muss. Ansonsten ist die IP-Adresse wieder frei.

Konnte der DHCP-Server eine entsprechende Konfiguration für den Client ermitteln, so antwortet er mit einer DHCPOFFER-Nachricht, welche diese enthält. Konnte er hingegen keine Konfiguration ermitteln, da z.B. alle IP-Adressen belegt sind so lehnt er die Anfrage mit einer DHCPNAK-Nachricht ab.

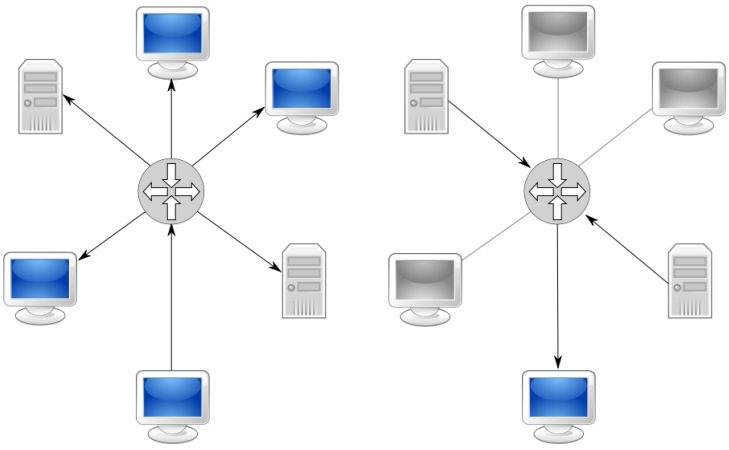

Abb. 4.8 *DHCP-Anfrage: a) DHCPDISCOVER, b) DHCPOFFER*

Da sich nun aber auch mehrere DHCP-Server im lokalen Netzwerk befinden können ist es möglich, dass der DHCP-Client mehrere Angebote erhält. In der Regel wählt er das erste Angebot und bestätigt dem entsprechenden DHCP-Server die Annahme mit einer DHCPREQUEST-Nachricht. Dabei wird diese Nachricht wieder per Broadcast in das loka-

le Netzwerk geschickt, wodurch alle weiteren DHCP-Server, die nicht ausgewählt wurden automatisch eine Absage erhalten (Abb. 4.8).

Schließlich sendet der DHCP-Server alle weiteren für die Konfiguration nötigen Information mittels einer DHCPACK-Nachricht an den Client. Bevor nun der Client die ihm zugeteilte IP-Adresse übernimmt, sichert er sich noch einmal ab, indem er mittels eines ARP-Requests überprüft, ob die IP-Adresse im lokalen Netzwerk bereits vergeben ist.

Handelt es sich um eine dynamische Zuordnung, so muss der Client vor Ablauf der Lease-Time eine Verlängerung mittels einer DHCPREQUEST-Nachricht anfragen ansonsten wird die IP-Adresse neu vergeben. Nach dem der Client schließlich nicht mehr über eine IP-Adresse benötigt, kann er dies dem DHCP-Server mittels einer DHCPRELEASE-Nachricht mitteilen. Unabhängig ob die Lease-Time abgelaufen ist, wird die IP-Adresse ab diesem Zeitpunkt neu vergeben.

Aufgrund der festen Verbindung mit der Internetschicht folgt DHCP bei der Versionsnummer dem Internet Protocol. Man unterscheidet daher aktuell zwischen DHCPv4 (für IPv4) und DHCPv6 (für IPv6). Da mit der Einführung von IPv6 die Funktionen des DHCPv4 durch das Neighbor Discovery Protocol und ICMPv6 größtenteils bereits in der Internetschicht erfolgen, werden bei DH- CPv6 die Konfigurationen weiterer Netzwerkprotokolle wie NTP ausgeliefert. DHCPv6 unterstützt unter dem Begriff „Stateless DHCPv6“ die in IPv6 eingeführte Zustands-lose Adresskonfiguration, indem nach einer DHCP-Anfrage ausschließlich die nicht IP-Konfigurationen ausgeliefert werden. Dem entgegen entspricht die grundsätzliche Vorgehensweise bei der Zustands-behafteten Adresskonfiguration der von DHCPv4 und wird als „statefull DHCPv6“ bezeichnet.

4.4.2 Domain Name System (DNS)

Die Identifikation eines Knoten mittels einer eindeutigen IP-Adresse ist eine effiziente Verfahrensweise, zur Handhabung der Kommunikation zwischen mehreren maschinellen Endstellen. Für Personen jedoch, welche diese Endstellen als Schnittstellen benutzen ist es wesentlich einfacher andere Knoten durch sprechbare Namen zu identifizieren und sich zu merken. Zu diesem Zweck wurde 1983 mit dem *Domain Name System* (DNS) ein Mechanismus entworfen, um solche Namen in IP-Adressen aufzulösen.

Hierfür wird die Zuordnung zwischen Namen und IP-Adressen in einer verteilten hierarchischen Datenbank hinterlegt, welche einen bestimmten sog. Namensraum zur Verfügung stellt. Die Hierarchie innerhalb des DNS-Namensraumes folgt einer Aufteilung in *Domänen* (engl.: *Domain*), *Unterdomänen* (engl.: *Subdomain*) und *Zonen*. Ein Domäne (bzw. Unterdomäne) bezeichnet einen zusammengehörigen Bereich, der durch gleiche Namensendung hervorgeht. Eine Zone bezeichnet einen Teilbereich einer Domäne, der durch einen DNS-Server (Nameserver) verwaltet wird.

Allgemeine Namensräume

Im Allgemeinen ist ein Namensraum eine Menge von Bezeichnungen zur Identifikation von beliebigen Objekten. Dabei unterliegen die Bezeichnungen bestimmten Regeln, so dass bei einer willkürlichen Bezeichnung einfach überprüft werden kann, ob diese den Regeln des Namensraumes unterliegt und die Bezeichnung somit im Namensraum enthalten ist.

Typischerweise werden Namensräume hierarchisch wie Bäume strukturiert. Dabei erhalten untergeordnete Objekte eigene kurze Bezeichnungen, sog. *Labels*. Einzelne Objekte können nun durch die Zusammensetzung der Labels identifiziert werden. Diese Zusammensetzung wird dann im Allgemeinen als *Pfad* (nicht mit Pfad ein einem Netzwerk verwechseln) bezeichnet. Ein einfaches Beispiel hierfür ist die Identifikation einer Person mittels Familienname und Vorname. Dabei ist „Familienname" das Label für die Familie und „Vorname" das Label für die Person innerhalb einer Familie. Nun muss aber auch geregelt werden, wie diese Labels zu einem Pfad zusammengesetzt werden. Im Beispiel ist die Zusammensetzung bekannt: „Vorname Nachname".

DNS-Namensraum

Im DNS-Namensraum werden Labels durch Punkte voneinander getrennt in der *Punktnotation* zu Pfaden zusammengesetzt. Die Hierarchie ist dabei von links nach rechts aufsteigend, so dass die höchste Ebene ganz rechts steht.

Der DNS-Namensraum unterliegt folgenden Einschränkungen:

- Labels können aus alphanumerischen Zeichen sowie dem Sonderzeichen „-" gebildet werden, wobei die Groß und Kleinschreibung nicht berücksichtigt wird. Daraus ergibt sich ein Zeichenvorrat von lediglich 37 verschiedenen Zeichen. Die Länge eines Labels darf zwischen einem und bis zu maximal 63 Zeichen umfassen. Das erste Zeichen muss stets alphabetisch und das letzte darf nicht „-" sein. Zur Ermöglichung internationaler Zeichen (z.B. äöüé etc.) werden diese mittels des sog. *Punycodes* in erlaubte Zeichen kodiert.

- Ein Pfad darf inklusive aller Punkte eine maximale Länge von 255 Zeichen haben

Im DNS Namensraum haben die einzelnen Ebenen teilweise besondere Bedeutungen:

Da der DNS Namensraum zur Identifikation einzelner Hosts im Internet dient und die Hierarchie von links nach rechts aufsteigend ist, bezeichnet der volle Pfad immer einen Host, sowie das erste Label den *Hostnamen*. Im Beispiel „www.iana.org" ist der Name des Hosts „www". Häufig orientieren sich die Hostnamen an der Aufgabe eines Hosts. So erhalten Webserver üblicherweise den Namen „www" und Mailserver den Namen „mail".

Die Labels der höheren Ebenen bezeichnen einzelne Domänen. Dabei muss ein gültiger *Domänenname* die Labels aller höheren Ebenen umfassen. Im Beispiel „www.iana.org" sind die gültigen Domänennamen „org" und „iana.org" enthalten, kein gültiger Domänenname ist jedoch „iana" ohne „.org". Eine Domäne die aufgrund der Bezeichnung einer anderen untergeordnet werden kann heißt bezüglich dieser auch *Unterdomäne* oder *Subdomain*, so ist „iana.org" eine Unterdomäne von der Domäne „org".

Ein gültiger Pfad wird auch *Fully Qualified Domain Name* (FQDN) genannt. Die Eindeutigkeit eines solchen ergibt sich aus der Zusammensetzung des Hostnamen und des Domänennamens, in welchem der Host vorhanden ist.

Die Domänen der höchsten Ebene heißen *Top-Level-Domains* (TLD) und werden wie die IP-Adressen, die Portnummern und die AS-Nummern von der IANA verwaltet. Diese teilt die TLDs in mehrere Gruppen (Tbl. 4.11).

Klasse		Beispiele	Beschreibung
gTLD	uTLD	com, net, org	Allgemeine TLDs (engl.: generic, gTLD) werden in nicht gesponserte (engl.: unsponsored, uTLD), von der IANA kontrollierte und gesponserte (engl.: sponsored, sTLD), von unabhängigen Organisationen und Dachverbänden kontrollierte unterschieden.
	sTLD	edu, gov, mil, tel, travel	
ccTDL		de, fr, uk	Länderspezifische TLDs (engl.: country-code, ccTLD) werden für eine geographische Zuordnung per Ländercode nach ISO 3166 Standard verwendet
iTDL		arpa, root	Infrastruktur-TLDs (iTLD) werden für spezielle Auskünfte bzw. technische Infrastrukturzwecke des Internet verwendet

Tbl. 4.11 *Klassifikation der Top-Level-Domains durch die IANA*

DNS-Namensauflösung

Ausgehend von einem Host der zu einem DNS-Namen die entsprechende IP-Adresse ermitteln will, erfolgt die Namensauflösung in mehreren Schritten. In allen Schritten wird jeweils unabhängig vom darunterliegenden Netzwerk eine Auflösung in eine IPv6 Adresse bevorzugt. Wird dies jedoch vom Host, bzw. eine am Host beteiligten Komponenten nicht unterstützt, so wird in eine IPv4 Adresse aufgelöst.

1. **Suche in eigenem DNS-Cache**

   Der DNS-Client der einen Namen auflösen will sucht den Eintrag zuerst in einer lokalen Tabelle, dem sog. *DNS-Cache*. Diese enthält Beispielsweise die Ergebnisse früherer Suchanfragen oder die Hostnamen im lokalen Netzwerk.

2. **Rekursive Anfrage an DNS-Server**

   Kann der Name nicht in dieser Tabelle gefunden werden, so wird ein *DNS-Server* (auch *Nameserver*) in Form einer Anfrage, eines sog. *DNS-Requests* befragt. Bei einem DNS-Request wird zwischen zwei Formen unterschieden: Bei einer *rekursiven Anfrage* kümmert sich der angefragte Server vollständig um die Namensauflösung. Bei einer *iterative Anfrage* hingegen gibt der Server in dem Fall das er den Namen nicht selber auflösen kann eine Liste mit Serveradressen zurück, welche „mehr Informationen“ besitzen.

   Da der DNS-Client einen DNS-Server um die Namensauflösung bemühen will handelt es sich hierbei um eine rekursive Anfrage. Hierfür ist es jedoch erforderlich, dass bereits vorher die IP-Adresse (mindestens) eines DNS-Servers manuell oder per DHCP beim DNS-Client eingetragen wurde.

Kann der DNS-Server den Namen selbst auflösen so antwortet er mit der entsprechenden IP-Adresse des gewünschten Hosts und die Anfrage ist beendet. Wenn der DNS-Server nun keinen Eintrag findet, der gesuchte Host aber in dessen Zuständigkeit (also nicht in einer anderen Domäne oder einer Subdomain) fällt, so antwortet er dem Client, dass der Host nicht bekannt ist bzw. mit der Nichtauflösbarkeit des Namens.

3. Iterative Anfrage durch den DHCP-Servers (eigene Domäne)

Kann der DNS-Server die Anfrage nicht selbst auflösen, jedoch liegt der Name in der Zuständigkeit einer (direkten) Unterdomäne, so befragt er den zuständigen Server dieser (direkten) Unterdomäne mittels einer iterativen Anfrage. Hierauf erhält er entweder die gewünschte IP-Adresse des Hosts, die Antwort, dass der Host unbekannt ist, oder eine Liste von wiederum untergeordneten zuständigen Servern. In diesem Fall wird die Anfrage so lange an den jeweils nächsten untergeordneten Server mittels einer iterativen Anfrage weitergegeben, bis irgendwann entweder die IP-Adresse des Host bekannt ist oder der Host als unbekannt identifiziert wurde. Schließlich kann der DNS-Server dem Client eine Antwort geben und die Anfrage ist beendet (Abb. 4.9a).

4. Iterative Anfrage durch den DHCP-Servers (andere Domäne)

Kann der DNS-Server die Anfrage nicht selbst auflösen und zudem liegt der Name nicht in seiner eigenen Zuständigkeit, noch in der einer untergeordneten Domäne, so befragt er zunächst einen sog. *Rootserver* (auch *Root-Nameserver*) mittels einer iterativen Anfrage. Die Rootserver dienen der Verwaltung der höchsten Ebene im DNS-Namensraum und verfügen hierfür über eine vollständige Liste der zuständigen Server der einzelnen TLDs. Somit wird sichergestellt, dass der ursprüngliche DNS-Server bei einer gültigen TLD eine Liste mit zuständigen Servern erhält. Aus diesen wiederum wählt er einen Server und gibt die iterative Anfrage des Namens weiter. Nun gibt es wieder nur die drei Möglichkeiten, dass der Name entweder bekannt oder unbekannt ist, oder in einer Unterdomäne verwaltet wird. Mittels weiterer iterativen Anfragen erhält der DNS-Server schließlich die gewünschte Information und kann diese an den Client weitergeben. (Abb. 4.9b)

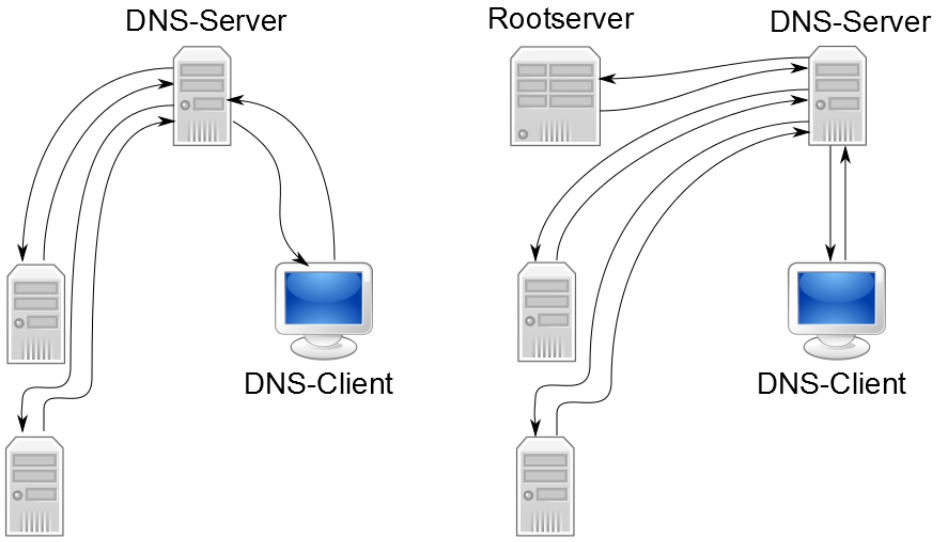

Abb. 4.9 *DNS-Namensauflösung: a) Eigene Domäne, b) Andere Domäne*

Sicherung der Rootserver

Die beschriebene Verfahrensweise verursacht jedoch eine Flut an DNS-Anfragen an die zentralen Rootserver. Um einen Zusammenbruch des DNS zu verhindern wurden verschiedene Mechanismen umgesetzt:

- Die Rootserver haben mit den ca. 3000 Einträgen zuständiger DNS-Server einen sehr eingeschränkten Aufgabenbereich. Des Weiteren werden nur iterative Anfragen bearbeitet und zugunsten einer schnelleren Verarbeitung auf Fehlerüberprüfung verzichtet.

- Es stehen momentan 13 Rootserver zur Verfügung. Diese sind größtenteils selbst keine einzelnen Hosts, sondern Anycast-Gruppen aus mehreren weltweit verteilten Hosts.

- Jeder einzelne der Rootserver muss nach Voraussetzung die dreifache Menge an Anfragen bearbeiten können wie der am stärksten belastete Rootserver. Dadurch kann der Betrieb auch bei Ausfall von 8 der 13 Rootserver gewährleistet werden.

Reverse DNS

Da in einigen Fällen auch die Ermittlung eines DNS-Namens zu einer bestimmten IP-Adresse erforderlich ist, unterscheidet man in *forward lookups* (v. engl. „direkte Suche“) für die Namensauflösung und *reverse lookups* (v. engl. „umgekehrte Suche“) für die Umkehrung. Diese umgekehrte Suche wird auch als *Reverse DNS* bezeichnet.

Die technische Umsetzung von Reverse DNS benutzt eine herkömmliche Namensauflösung, wobei der Name jedoch aus der IP-Adresse gebildet wird. Als Antwort erhält der Client schließlich den gesuchten DNS-Namen zu der IP-Adresse. Um IP-Adressen in den

DNS-Namensraum abzubilden wurden die Domänen in-addr.arpa (für IPv4-Adressen) und ipv6.arpa (für IPv6- Adressen) eingerichtet.

Die Bildung des DNS-Namens erfolgt nun sukzessive durch Bildung von Unterdomänen aus den einzelnen Blöcken der IP-Adresse. Hierbei wird für die jeweilig nächste Unterdomäne der hierarchisch höchste Block der IP-Adresse aus dieser entfernt und als Label verwendet. Dies führt aufgrund der umgekehrten Richtung der Hierarchie zu einer Umkehrung der IP-Adresse in den DNS-Namen. Die Größe der Blöcke umfasst bei IPv4 acht Bit und bei IPv6 vier Bit.

Bsp. 4.5 (Reverse DNS)
IPv4: `130.94.122.195` wird zu `195.122.94.130`.in-addr.arpa
IPv6: `2001:470:1f06:11f::2` wird zu
`2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.f.1.1.0.6.0.f.1.0.7.4.0.1.0.0.2`.ipv6.arpa

Der aus der IP-Adresse generierte DNS-Name wird nun wie bei der herkömmlichen Namensauflösung an einen DNS-Server übergeben, welcher diesen an einen Rootserver weiterleitet. Der Rootserver wiederum leitet die Anfrage an den Nameserver, der für die „arpa“ Domäne zuständig ist weiter etc. Nun tritt aber das Problem auf, dass die Weiterleitungen nicht mehr über herkömmliche Domänennamen, sondern über Adressbereiche zu ermitteln sind, allerdings sollen für reverse lookups die gleichen Server wie für forward lokkups verwendet werden. Aus diesem Grund sind in Nameservern häufig mehrere Eintragungen vorhanden, die einerseits den Namen der Unterdomänen, als auch ihren Adressbereichen den entsprechenden Nameserver zuordnen.

Bei der Auflösung übergibt der entsprechende Nameserver schließlich als Antwort den Eintrag welcher unter dem IP-generierten DNS-Namen zu finden ist. Da die Reverse DNS Einträge jedoch nicht für die Erreichbarkeit der einzelnen Knoten benötigt werden, sondern lediglich als Auskunft über den DNS-Namen fungieren müssen diese weder vollständig noch korrekt sein. Beispielsweise kann eine Reverse DNS Anfrage auf eine bestimmte IP-Adresse als Antwort den DNS-Namen eines anderen Hosts innerhalb der Domäne (z.B. ein Web-Server) zur Folge haben.

Lastverteilung per DNS

Das Domain Name System ermöglicht die Zuordnung verschiedener IP-Adressen zum gleichen DNS-Namen. Bei einer DNS-Anfrage umfasst die Antwort dann alle diese Einträge, wobei diese jedoch in einer Reihenfolge übergeben werden. Der Client wird in der Regel dann die erste IP-Adresse auswählen.

Dadurch kann eine einfache Lastverteilung auf mehrere Systeme, die den gleichen Dienst anbieten realisiert werden. In Abhängigkeit des DNS-Servers ist es beispielsweise möglich die Reihenfolge mit einer Gewichtung zu versehen, so dass eine bestimmte IP-Adresse mit einer vorgegebenen Wahrscheinlichkeit die Reihenfolge anführt. Je nach den Möglichkeiten des DNS-Servers können auch Standort bezogene oder andere Informationen zur automatischen Generierung der Gewichtung herangezogen werden. Diese Vorgehensweise lässt sich mit der Verwendung einer Metrik im Routing vergleichen, findet jedoch auf einer ganz anderen Abstraktionsebene statt.

Ein Nachteil dieser Methode ist, dass keine Kontrolle über die tatsächliche Lastverteilung möglich ist: Ein Client der einen Namen in eine IP-Adresse auflöst hat, wird diese lokal speichern und ab diesem Zeitpunkt stets die Dienste des gleichen Systems verwenden. Da die Umsetzung jedoch sehr einfach und unabhängig von der Netzwerkinfrastruktur ist wird dieses Verfahren in der Praxis sehr häufig verwendet.

4.4.3 Network Time Protocol (NTP)

Bei der Kommunikation in Netzwerken tritt häufig die Notwendigkeit einer zeitlichen Synchronisation der Kommunikationspartner ein. Dies ist beispielsweise der Fall, wenn ein Server einen zeitlich beschränkten Dienst (z.B. Authentifizierung mit Zeitstempel und Ablaufzeitpunkt) anbietet. Will ein Benutzer des Dienstes (Client) diesen auch über den Ablaufzeitpunkt weiterhin benutzen so ist eine rechtzeitige „Erneuerung“ erforderlich. Dafür müssen Server und Client jedoch Zeit-synchron arbeiten.

Das *Network Time Protocol* (NTP) ist ein Client-Server basierendes Protokoll zur Zeit-Synchronisation in Netzwerken. Mittels zentraler öffentlicher Server ermöglicht es in aktueller Version (NTPv4) eine globale Zeitkoordination mit einer Genauigkeit von ca. 10 Millisekunden. Hierfür wird das internationale Zeitformat UTC (v. engl. Universal Time Coordinated) verwendet.

NTP strukturiert die Kommunikation in hierarchische Ebenen, welche sich aufgrund der jeweiligen Zeitquelle ergeben. Ein NTP-Server mit einer vom TCP/IP Netzwerk unabhängigen hochpräzisen Zeitquelle, wie z.B. eine Atomuhr wird in dieser Hierarchie als „Stratum 0“ bezeichnet. Ein NTP-Client der die Zeit von einem Server mit Stratum 0 bezieht als Stratum 1. Wenn dieser Host wiederum NTP-Server-Dienste anbietet und ein weiterer Host diese mittels eines NTP-Clients nutzt, so wird dieser weitere als „Stratum 2“ bezeichnet usw. Des Weiteren kennt NTPv4 verschiedene Modi:

- **Client / Server Modus**

  NTP-Clients schicken einen NTP-Request an den NTP-Server. Dieser antwortet und der Client synchronisiert seine lokale Uhrzeit entsprechend dem Ergebnis.

- **Symmetrischer Modus**

  In diesem Modus umfassen die Hosts sowohl NTP-Client, als auch NTP-Server. Durch gegenseitige Bekanntgabe der eigenen lokalen Zeit, synchronisieren sich mehrere Hosts, so dass eine einheitliche Zeitgebung ermöglicht wird.

- **Broadcast Modus**

  Im Broadcast Modus schicken NTP-Server regelmäßig Broadcast Pakete mit den Zeitinformationen. NTP-Clients synchronisieren die lokale Zeit der Hosts dann automatisch nach diesen Informationen.

Diese Modi ermöglichen es die Zeitsynchronisation bis zu einem gewissen Maß an die gegebene Netzwerktopologie anzupassen.

4.5 Aufgaben zu Kapitel 4

Aufgabe 4.1

Worin besteht die Verbindung zwischen Netzwerkadresse, Hostadresse und IP-Adresse und wofür werden diese benötigt?

Aufgabe 4.2

Beschreiben Sie die historische Entwicklung der Netzwerkadressierung (Netzklassen ⇒ Subnet-Mask & Subnetting ⇒ CIDR)

Aufgabe 4.3

Welche Netzwerkadressen und Broadcast-Adressen gehören zu folgenden IPv4-Adressen: (1) `192.168.0.1/24` (2) `130.94.122.195/16` (3) `172.16.4.0/20`

Aufgabe 4.4

Nennen Sie drei Vorteile von IPv6 gegenüber IPv4

Aufgabe 4.5

Welche Netzwerkadresse, Interfaceadresse und welcher Adress-Typ gehört zu folgender IPv6-Adresse: `2001:470:1f06:11f::2/16`

Aufgabe 4.6

Welche Gültigkeitsbereiche kennt IPv4, welche IPv6? (inkl. kurze Beschreibung)

Aufgabe 4.7

In welchen Gültigkeitsbereichen ist in IPv4 und IPv6 Multicasting & Broadcasting möglich?

Aufgabe 4.8

Welche Vorteile besitzen UDP und TCP und wofür eignen sie sich daher besonders?

Aufgabe 4.9

Beschreiben Sie eine DHCP Anfrage und deren Antwort wenn mehrere DHCP-Server zur Verfügung stehen

Aufgabe 4.10

Warum sind die Rootserver theoretisch (im Sinne des DNS-Protokolls) interessante Ziele eines terroristischen Anschlags? Warum sind sie es nicht in der Praxis?

LÖSUNGEN ZU DEN AUFGABEN

Aufgaben zu Kapitel 1

Aufgabe 1.1 *Erklären und unterscheiden Sie die Begriffe „Bitrate“, „Übertragungsrate“ und „Durchsatzrate“.*

Bei der Datenübertragung bezeichnet die Bitrate, die gesamte übertragende Datenmenge pro Zeiteinheit, die Datenübertragungsrate die fehlerkorrigierte Datenmenge pro Zeiteinheit und die Datendurchsatzrate die unkomprimierte, fehlerkorrierte Datenmenge (und somit die Nutzdatenmenge) pro Zeiteinheit.

Aufgabe 1.2 *Welche Fehlerkorrektur, Kompression und Verschlüsselung eignen sich zur Übertragung von Dateien, welche zur Übertragung von Sprachsignalen?*

- Zur Übertragung von Dateien eignen sich die Rückwärtsfehlerkorrektur, sowie beliebige Kompressionen und Verschlüsselungen.

- Zur Übertragung von Sprachsignalen eignen sich die Faltungscodierung, die Stromverschlüsselung und die Irrelevanzreduktion.

Aufgabe 1.3 *Tauschen Sie in dem Wort „IT“ jeden Buchstaben durch die entsprechende Nummer im Alphabet. Bei einstelligen Nummern ergänzen Sie eine führende Null. Führen Sie anschließend eine Codierung in den BCD-Code durch. Schreiben Sie die Bitfolge dabei in vier Blöcken untereinander und ermitteln Sie die Paritätsbits für die Zeilen und die Spalten.*

„IT“ $\Rightarrow$ 0920 $\Rightarrow$

0	0	0	0	0
1	0	0	1	0
0	0	1	0	1
0	0	0	0	0
1	0	1	1	

Aufgabe 1.4 *Erklären Sie unter Verwendung der Tabelle in Aufgabe 1.3 warum der Empfänger bei Empfang des Wortes „ET“ mit Hilfe der zuvor ermittelten Paritätsbits den Fehler zwar erkennt, aber nicht automatisch korrigieren kann. Um was für eine Art von Übertragungsfehler handelt es sich? Durch welches zusätzliche Verfahren hätte der Fehler korrigiert werden können?*

Der Empfänger erhält das Wort „ET“, und damit sie Bitfolge: `0000 01`01 0010 0000

Der Vergleich mit der Tabelle in Aufgabe 1.3 zeigt, dass zwei Bits in Folge fehlerhaft übertragen wurden. Der Empfänger kann durch die Paritätsbits nur noch feststellen in welchen Spalten, aber nicht in welcher Zeile die Fehler aufgetreten sind. Es handelt sich bei dem Fehler um einen Burst-Fehler, der durch den Einsatz von Interleaving hätte korrigiert werden können.

Aufgabe 1.5 *Welche Zusammenhänge bestehen zwischen der realen Kanalkapazität und der Bitfehlerrate?*

$\Rightarrow$ Wird eine Vorwärtsfehlerkorrektur mit entsprechender Korrekturfähigkeit eingesetzt, so ist die reale Kanalkapazität unabhängig von der Bitfehlerrate.

⇒ Wird jedoch eine Rückwärtsfehlerkorrektur eingesetzt, so führt eine Erhöhung der Bitfehlerrate zu einer Absenkung der realen Kanalkapazität

Aufgabe 1.6 *Erklären Sie die Begriffe „Öffentlicher Schlüssel“ und „Privater Schlüssel“.*

Öffentlicher und Privater Schlüssel werden beim Public-Key Verfahren angewandt, um den Austausch eines sicheren Schlüssels zu umgehen. Dabei verschlüsselt der Sender die Daten mit dem Öffentlichen Schlüssel des Empfängers. Der Empfänger entschlüsselt diese mit dem privaten Schlüssel.

Aufgabe 1.7 *Eine Code besitzt die beiden Codewörter: c_1 = (1,1,-1,-1), c_2 = (1,-1,-1,1). Eignet sich dieser Code für Codemultiplexing? Wie viele Sender sind dabei möglich?*

$$c_1 \circ c_1 = \frac{1 \cdot 1 + 1 \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1)}{4} = 1$$

$$c_1 \circ c_2 = \frac{1 \cdot 1 + 1 \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot 1}{4} = 0$$

$$c_2 \circ c_2 = \frac{1 \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot 1}{4} = 1$$

⇒ c_1 und c_2 sind also orthogonal und eignen sich für das Codemultiplexing mit zwei Sendern

Aufgabe 1.8 *Ein Auto fährt auf eine Straßenkreuzung zu, erkennt dass bereits andere Autos auf der Straße sind und hält an. Sobald die Straße frei ist beschleunigt der Fahrer den Wagen. Welchem Zugriffsverfahren entspricht diese Vorgehensweise? Beschreiben sie ein weiteres Zugriffsverfahren in diesem Kontext.*

Die vorgehendweise entspricht persistentem CSMA.

Ein ALOHA Fahrer würde nicht schauen, ob die Kreuzung belegt ist sondern in jedem Fall fahren. Kommt es zu einem Unfall, so lässt der Fahrer seinen Wagen reparieren, wartet eine zufällige Zeitdauer und geht wieder genauso vor.

Aufgaben zu Kapitel 2

Aufgabe 2.1 *In Abschnitt 2.1.4 steht: „Bei einer einzigen Zelle entspricht die physische Topologie einer Bus-Topologie und die logische einer Stern-Topologie“. Begründen Sie diese Aussage.*

Da in einer Zelle alle Teilnehmer das gleiche Übertragungsmedium und damit den gleichen Übertragungskanal verwenden handelt es sich hier um eine physische Bus-Topologie. Insbesondere sind hier Zugriffsverfahren erforderlich. Eine Kommunikation zwischen zwei Teilnehmern der Zelle findet jedoch über den zentralen Verteiler, den Zugangsknoten statt, weshalb es sich um eine logische Stern-Topologie handelt.

Aufgabe 2.2 *Skizzieren Sie einen minimalistischen Graphen als Modell für das Internet, bei dem anhand der Topologie Zentralbereich, autonome Systeme und Zugangsbereich erkennbar sind.*

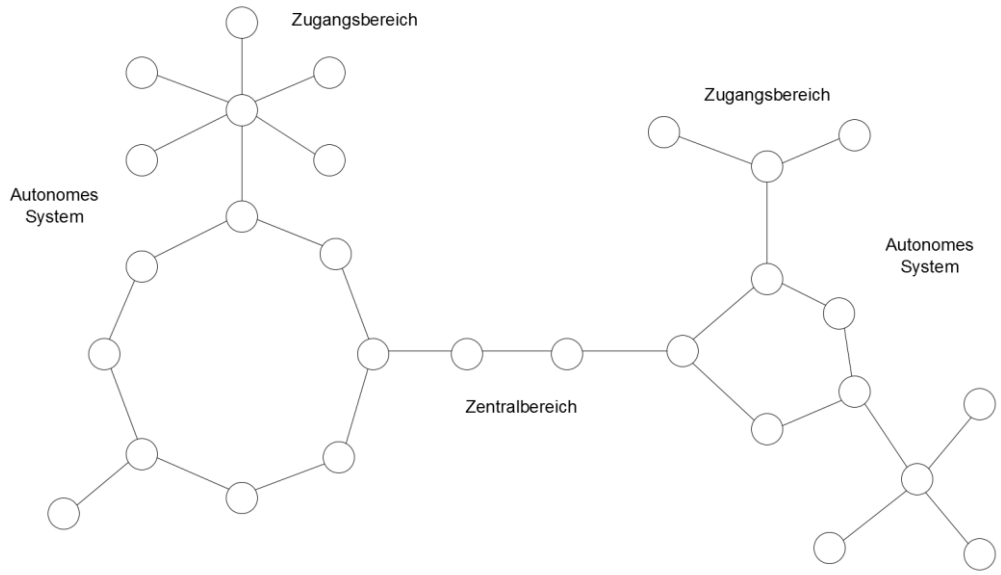

Abb. *Skizze - Struktur des Internet*

Aufgabe 2.3 *Bestimmen sie zu diesem Graphen die Konnektivität, den Durchmesser und die Bisektionsweite. Schätzen Sie grob die tatsächlichen Größen für das Internet ab.*

- Kenngrößen für den Graph aus Aufgabe 2.2:
  Konnektivität: 1, Durchmesser: 17, Bisektionsweite: 1

- Geschätzte Kenngrößen für das Internet:

  - Konnektivität: 1

  - Durchmesser: sehr groß

  - Bisektionsweite: Vielleicht zwischen 5 - 20

    Die Bisektionsweite des Internets hängt vorwiegend von der Beschaffenheit des Zentralbereichs ab. Tatsächlich sind hier die Signalleitungen redundant vorhanden. Zudem müssen Satellitenverbindungen berücksichtigt werden, weshalb von keiner sehr kleinen Bisektionsweite auszugehen ist. Andererseits wäre eine sehr große Bisektionsweite wirtschaftlich nicht tragbar.

Aufgabe 2.4 *Was versteht man unter Kosten? Was ist zur Bestimmung von Kosten erforderlich und wozu werden sie verwendet?*

In einem Graphen bezeichnen „Kosten“ Bewertungen von Kanten und Wegen die es ermöglichen diese zu vergleichen. Dabei sind Pfade mit niedrigen Kosten besser zu bewerten als solche mit hohen Kosten. Die Bestimmung der Kosten in einem Netzwerk ist eine Gewichtung des Graphen sowie eine gegebene Metrik erforderlich. Kosten werden z.B. ver-

wendet um in einem Computernetzwerk bei der Datenübertragung günstige Übertragungswege zu bestimmen.

Aufgabe 2.5 *Betrachten Sie den Graph in Abb. 2.1. Der Knoten v_1 möchte Daten zu v_7 übertragen. Als Metrik kommt die Anzahl der Kanten (Hop-Count) zum Einsatz. Bestimmen Sie die Kosten der kürzesten Pfade, die über die einzelnen Nachbarn von v_1 gehen. Erstellen Sie eine Routing-Tabelle für diese Übertragung, welche Routen über die einzelnen Nachbarn von v_1 berücksichtigt.*

Der Knoten v_1 besitzt drei Nachbarn: v_3, v_4 und v_7

- Über v_3 ist der Kürzeste Pfad: $\rightarrow v_3 \rightarrow v_2 \rightarrow v_4 \rightarrow v_7$ mit der Pfadlänge 4
- Über v_4 ist der Kürzeste Pfad: $\rightarrow v_4 \rightarrow v_7$ mit der Pfadlänge 2
- Über v_6 ist der Kürzeste Pfad: $\rightarrow v_6 \rightarrow v_5 \rightarrow v_7$ mit der Pfadlänge 3

Da bei der Hop-Count Metrik die Kosten identisch mit der Pfadlänge sind, ergibt sich folgende Tabelle:

Zielknoten	Nachbar	Kosten
v_7	v_3	4
v_7	v_4	2
v_7	v_6	3

Aufgabe 2.6 *Betrachten Sie wieder den Graph in Abb. 2.1 und erstellen Sie für den Knoten v_1 eine Routing-Tabelle unter Verwendung des Dijkstra-Algorithmus. Führen Sie hierfür zunächst schrittweise die Liste der bekannten Knoten $B_1 = (v_1)$, $B_2 = (\ldots)$, usw.*

$B_1 = (v_1)$, $B_2 = (v_1, v_3, v_4, v_6)$, $B_3 = (v_1, v_3, v_4, v_6, v_2, v_5, v_7)$

Zielknoten	Nachbar	Kosten
v_3	v_3	1
v_4	v_4	1
v_6	v_6	1
v_2	v_3	2
v_5	v_6	2
v_7	v_4	3

Aufgabe 2.7 *Warum eignet sich Anycast für eine flächendeckende Bereitstellung von Diensten? Was muss der Betreiber dabei berücksichtigen? Finden Sie jeweils ein konkretes Beispiel für die sinnvolle Anwendung von Broadcast und Multicast.*

Da bei einer Anfrage jeweils nur ein Knoten der Anycast-Gruppe antwortet, führt dies zu einer Entlastung der anderen Knoten. Werden die einzelnen Knoten über eine große Fläche gleichmäßig verteilt, so können Dienste flächendeckend angeboten werden. Dabei ist zu berücksichtigen, dass jeder Knoten der Anycast-Gruppe mit den anderen identisch ist. Ein Beispiel für Multicast ist der Versandt einer E-Mail an eine Liste von Adressen. Ein Beispiel für Broadcast ist der Rundfunk.

Aufgabe 2.8 *Nennen Sie jeweils drei wichtige Vorteile für die Aufgabenverteilung nach dem Peer-to-Peer Modell und dem Client-Server Modell.*

- Vorteile des Client-Server Modells:
    - Gute Skalierbarkeit und geringere Kosten, da an zusätzliche Hosts nur minimale Anforderungen gestellt werden
    - Zentrale Wartung und zentrale Verwaltung verringern den Wartungs- / Verwaltungsaufwand
    - Vereinfachte Kontrolle / Überwachbarkeit durch befugte Dienstleiter

- Vorteile des Peer-to-Peer Modells:
    - Höhere Stabilität gegenüber Ausfällen
    - Höhere Leistungsfähigkeit in Bezug auf den Gesamtdatendurchsatz
    - Erschwerte Kontrolle / Überwachbarkeit durch Unbefugte

Aufgabe 2.9 *Entwerfen Sie eine leistungseffiziente Topologie zur Bisektionsweite n. Welche Komplexität besitzt diese Topologie?*

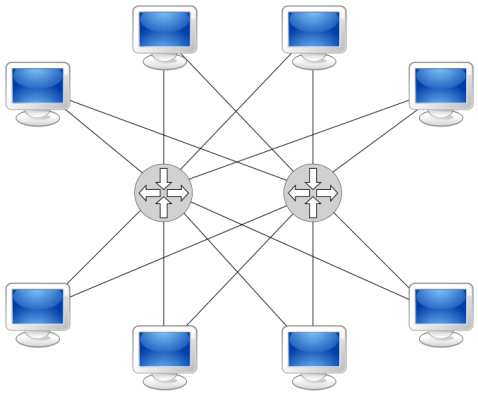

Abb. *Doppelstern*

Ein Doppelstern besitzt die Bisektionsweite *n* und die Komplexität *O(n)*, da die Anzahl der Verbindungen linear zur Anzahl der Knoten wächst.

Ein typisches Einsatzgebiet für Doppelsterne sind Netzwerke, deren Funktion in ein „Steuernetzwerk“ und ein „Arbeitsnetzwerk“ getrennt wird. Dadurch können im Arbeitsnetzwerk sehr leistungsfähige und im Steuernetzwerk sehr ausfallssichere Komponenten eingesetzt werden.

Aufgaben zu Kapitel 3

Aufgabe 3.1 *Warum können Protokolle unterschiedlicher Schichten nicht „direkt“ (mittels Steuerdaten) miteinander kommunizieren?*

Angenommen Protokoll „A“ liegt höher als Protokoll „B“. Dann ist es Protokoll „A“ möglich Informationen an „B“ in seine Steuerdaten aufzunehmen und „B“ kann diese dann auslesen. Umgekehrt kann „B“ zwar Informationen an „A“ in die Steuerdaten aufnehmen allerdings werden diese Steuerdaten bereits entfernt, bevor sie „A“ erreichen.

Aufgabe 3.2 *Wie wird dieses Problem im OSI-Modell gelöst? Verwenden Sie zur Beschreibung folgende Begriffe: „SSAP“, „DSAP“, „LLC-Schicht“, „Service Primitives“*

Sie wieder angenommen, Protokoll „A“ liegt höher im Protokollstapel als „B“. Protokoll „B“ möchte nun mit „A“ kommunizieren. Hierfür werden in der tiefliegenden LLC-Schicht von „B“ (bei Host von „B“) Informationen in deren Steuerdaten aufgenommen, die enthal-

ten, wer (SSAP) mit wem (DSAP) zu welchem Zweck (Service Primitive) in Kontakt treten will. Auf der Empfängerseite erhält die LLC-Schicht von „A“ nun diese Informationen und leitet sie direkt an „A“ weiter. Auf diese Art und Weise können „A“ und „B“ miteinander unabhängig von deren Position im Stapel kommunizieren.

Aufgabe 3.3 *Ein Sender und ein Empfänger implementieren den gleichen Protokollstapel mit N Protokollen. Skizzieren Sie eine Datenübertragung unter Verwendung der Beschriftungen „PDU(...)“ und „SDU(...)“.*

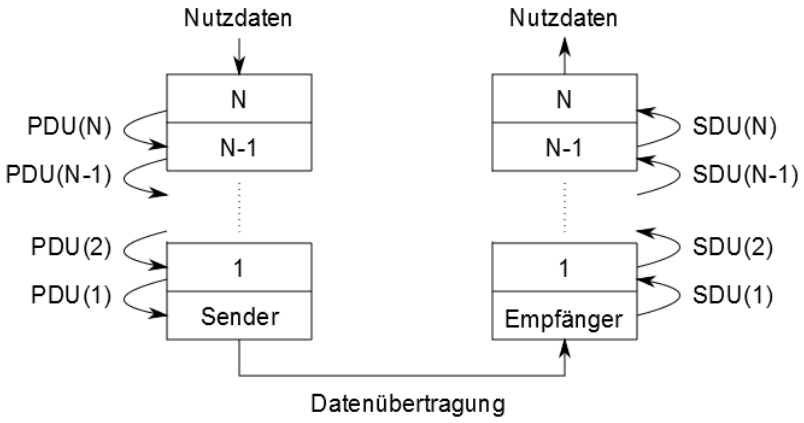

Abb. *Datenübertragung (PDU, SDU)*

Aufgabe 3.4 *Ordnen Sie folgende Begriffe den entsprechenden Schichten (bzw. Teilschichten) des OSI-Modells zu (Beschreibung der Begriffe ist nicht erforderlich): „Datenframe“, „Datenpaket“, „Datensegment“, „Bitstrom“, „DSAP“, „Flusskontrolle“, „Signal“, „SSAP“, „MTU“, „Multiplexing“, „Routing“, „Zugriffsverfahren“.*

Schicht	Begriffe
Transportschicht	„Datensegment“, „Flusskontrolle“, „MTU“
Vermittlungsschicht	„Datenpaket“, „Routing“
Sicherungsschicht (LLC-Schicht)	„DSAP“, „SSAP“
Sicherungsschicht (MAC-Schicht)	„Datenframe“, „Zugriffsverfahren“
Bitübertragungsschicht	„Bitstrom“, „Multiplexing“
Übertragungsmedium	„Signal“

Aufgabe 3.5 *Welche Arten der Adressierung treten in den transportorientierten Schichten des OSI-Modells auf und wofür werden sie benötigt?*

- **Bitübertragungsschicht**

  Das OSI-Modell sieht zwar keine spezielle Adressierung vor, wird allerdings in dieser Schicht z.B. asynchrones Zeitmultiplexing angewandt, so sind IDs der einzelnen Teilnehmer auf dem gemeinsam genutzten Übertragungsmedium erforderlich.

- **Sicherungsschicht**

  Physikalische Adressierung durch MAC-Adressen (OSI Schicht-2 Adressen): Umsetzung von Zugriffsverfahren. Zudem Adressierung von SAPs mittels DSAP und SSAP: Ermöglichung vertikaler Kommunikation

- **Vermittlungsschicht**

  Logische Adressierung (OSI Schicht-3 Adressen) der Start- und Endknoten im Internet, damit Zwischenstationen (Router) die Daten entsprechend weiterleiten können

- **Transportschicht**

  Adressierung der Kommunikationsendpunkte (OSI Schicht-4 Adressen) bzw. Prozesse, um auf einem Host mehrere gleichzeitige Verbindungen zu ermöglichen

Aufgaben zu Kapitel 4

Aufgabe 4.1 *Worin besteht die Verbindung zwischen Netzwerkadresse, Hostadresse und IP-Adresse und wofür werden diese benötigt?*

Die IP-Adresse setzt sich aus der Netzwerkadresse und der Hostadresse zusammen. Die Netzwerkadresse identifiziert ein Netzwerk eindeutig im Internet. Innerhalb dieses Netzwerkes wird der Host mittels der Hostadresse eindeutig identifiziert. Wenn die IP-Adresse N-Bit lang und die Netzwerkadresse n-Bit lang ist, so ist die Hostadresse $(N - n)$-Bit lang

Aufgabe 4.2 *Beschreiben Sie die historische Entwicklung der Netzwerkadressierung (Netzklassen ⇒ Subnet-Mask & Subnetting ⇒ CIDR)*

Anfänglich wurden Netzklassen verwendet um den IP-Adressraum in Netzwerke unterschiedlicher Größen einzuteilen. Dabei war es möglich die Größe des Netzwerkes und damit die Netzwerkadresse direkt aus der IP-Adresse auszulesen.

Diese grobe Einteilung in drei verschiedene Größen führte jedoch zu einer ineffizienten Nutzung der zur Verfügung stehenden Adressen. Daher ging man dazu über mittels einer Subnet-Mask die Bits, welche in der IP-Adresse die Netzwerkadresse repräsentieren anzugeben. Dies ermöglichte bestehende Netzwerke einer bestimmten Klasse durch Subnetting in kleinere aufzuteilen und somit den Adressbereich besser auszunutzen.

Durch häufiges Subnetting wurden allerdings die Routing-Tabellen immer länger. Daher schaffte man beim Classless Interdomain Routing (CIDR) die Netzklassen (und damit auch Teilnetze von Netzklassen) schließlich ab und führte sog. Präfixlängen ein. Diese bezeichnet die Länge der Netzwerkadresse

Aufgabe 4.3 *Welche Netzwerkadressen und Broadcast-Adressen gehören zu folgenden IPv4-Adressen:* (1) `192.168.0.1/24` (2) `130.94.122.195/16` (3) `172.16.4.0/20`

IPv4-Adresse	Netzwerkadresse	Broadcastadresse
`192.168.0.1/24`	`192.168.0.0/24`	`192.168.0.255/24`
`130.94.122.195/16`	`130.94.0.0/16`	`130.94.255.255/16`
`172.16.4.0/20`	`172.16.0.0/20`	`172.16.15.255/20`

Erklärung von (3): Die Netzwerkadresse ist 20-Bit lang, damit die die Hostadresse 12-Bit.
Wenn man die „hinteren“ 12 Bits durch 0 ersetzt erhält man: `172.16.0.0/20`
Wenn man die „hinteren“ 12 Bits durch 1 ersetzt erhält man: `172.16.15.255/20`

Aufgabe 4.4 *Nennen Sie drei Vorteile von IPv6 gegenüber IPv4*

- Wesentlich größerer Adressbereich
- Effizienteres Routing durch die Verwendung eines Routing-Prefix
- Netzwerk und Host können mittels Network-ID und Interface-ID unabhängig voneinander identifiziert werden
- Feinere Aufteilung der Gültigkeitsbereiche von IP-Adressen
- Unterstützung automatischer Adresszuweisung und Konfiguration

Aufgabe 4.5 *Welche Netzwerkadresse, Interfaceadresse und welcher Adress-Typ gehört zu folgender IPv6-Adresse:* `2001:470:1f06:11f::2/16`

Netzwerkadresse	`2001:470:1f06:11f::`
Interfaceadresse	`::2`
Adress-Typ	Global-scope Unicast Adresse die durch RIR/LIR verwaltet wird

Aufgabe 4.6 *Welche Gültigkeitsbereiche kennt IPv4, welche IPv6? (inkl. kurze Beschreibung)*

Gültigkeitsbereich	IPv4	IPv6	Eindeutigkeit innerhalb …
Node-local		X	Einzelner Host oder Anycastgruppe
Link-local		X	Link-Layer Broadcastdomäne (OSI-Schicht 2)
Site-local	X	X	Lokales Netzwerk
Organisation-local		X	LIR bzw. ISP
Global	X	X	Global

Aufgabe 4.7 *In welchen Gültigkeitsbereichen ist in IPv4 und IPv6 Multicasting & Broadcasting möglich?*

Site-local Broadcasting ist in IPv4 möglich mittels "Directed Broadcasts" und "Limited Broadcasts". Für Multicasting existieren hier feste Adressbereiche mit Site-local Multicasting Adressen und Global Multicasting Adressen.

In IPv6 werden keine Broadcasts in diesem Sinne unterstützt. Hierfür existieren spezielle Multicast Adressen, welche Node-local „Broadcasting" und Link-local „Broadcasting" ermöglichen. Des Weiteren können alle Router im lokalen Netzwerk adressiert werden. Für Multicasting werden in IPv6 mittels einer Scope-ID alle Gültigkeitsbereiche von Node-local bis Global ermöglicht.

Aufgabe 4.8 *Welche Vorteile besitzen UDP und TCP und wofür eignen sie sich daher besonders?*

TCP arbeitet verlässlich (fehlerfrei), verbindungsorientiert und verfügt über Mechanismen der Flusskontrolle. Zudem bleibt die Reihenfolge der Pakete erhalten. Dadurch eignet es sich um Daten vollständig und fehlerfrei zu übertragen.

UDP hingegen verzichtet auf die Fehlerkorrektur, auf die Einhaltung der Reihenfolge etc. Dadurch treten keine zeitlichen Verzögerungen (z.B. durch Rückwärtsfehlerkorrektur) auf und die Datenlast im Netzwerk ist minimal. Daher eignet sich UDP z.B. zur Übertragung von Audio- / Video-Signalen.

Aufgabe 4.9 *Beschreiben Sie eine DHCP Anfrage und deren Antwort wenn mehrere DHCP-Server zur Verfügung stehen*

Ein DHCP-Client versendet einen Site-local Broadcast mit einer DHCPDISCOVER-Nachricht. Empfängt ein DHCP-Server diese Anfrage und kann der MAC-Adresse eine Konfiguration zuordnen, so schickt dieser ein DHCPOFFER an den DHCP-Client. Nach Vorgabe der Aufgabenstellung erhält der Client nun mehrere DHCPOFFER-Nachrichten und wählt eine aus. Anschließend schickt er die Antwort, den DHCPREQUEST an den entsprechenden DHCP-Server, jedoch als Site-local Broadcast. Dadurch erkennen alle anderen DHCP-Server, dass sie „nicht erwünscht“ sind. Daraufhin schickt der DHCP-Server die Konfiguration an den Client als DHCPACK-Nachricht. Abschließend Überprüft der Client mittels einer ARP-Anfrage ob die IP-Adresse bereits belegt ist.

Aufgabe 4.10 *Warum sind die Rootserver theoretisch (im Sinne des DNS-Protokolls) interessante Ziele eines terroristischen Anschlags? Warum sind sie es nicht in der Praxis?*

Bei DNS-Anfragen außerhalb der eigenen Domäne erfolgt zunächst eine Anfrage der TLDs bei den Rootservern. Könnten diese nicht antworten würden viele DNS-Anfragen mit einem Timeout abbrechen. Dadurch würden die Kommunikationsinfrastrukturen vieler Einrichtungen und Behörden den Dienst versagen. Darüber hinaus hätte dies vermutlich weltweit volkswirtschaftliche Schäden zur Folge.

Da mind. 13 Rootserver zur Verfügung stehen, welche selbst wiederum aus weltweit verteilten Anycast-Gruppen bestehen wäre der Aufwand, alle gleichzeitig stillzulegen enorm. Daher würden höchstwahrscheinlich einzelne Rootserver bestehen bleiben. Diese Wiederum verfügen über vorgeschrieben Leistungsreserven die es ermöglichen, dass bereits fünf von Ihnen alle DNS-Anfragen übernehmen könnten.

LITERATURVERZEICHNIS

Becker, Drechsler, Molitor: Technische Informatik, Eine Einführung. München: Pearson Studium, 2005

Frisch, Hölzel, Lintermann, Schaefer: Vernetzte IT-Systeme. Bildungsverlag EINS, 2006

Freyer: Nachrichtenübertragungstechnik. 6., neu bearbeitete Auflage. München: Hanser, 2009

Lindner: Informationsübertragung. Berlin, Heidelberg: Springer, 2005

Tanenbaum: Computer Networks. 2nd Edition. Prentice Hall, 1988

STICHWORTVERZEICHNIS

E

F

G

H

I

R

S

T

U

V

W

Z

Korrekturaufgaben

Fernlehrgang:	Lehrbrief: XX/X/XX	Teilnehmer-Nr.:
Name:	Vorname:	Note:
Straße:		Datum der Bearbeitung:
PLZ, Wohnort:		Unterschrift des Korrektors:
Einsendedatum:		

Bei Teilnahme am Fernlehrgang senden Sie bitte Ihre Korrekturaufgaben mit dieser Vorlage an:
Fernlehrinstitut Dr. Robert Eckert GmbH, Dr.-Robert-Eckert-Str. 3, 93128 Regenstauf