

HANSER

Anatol Badach, Erwin Hoffmann

Technik der IP-Netze

TCP/IP incl. IPv6 - Funktionsweise, Protokolle und Dienste

ISBN-10: 3-446-21935-8

ISBN-13: 978-3-446-21935-9

Leseprobe

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-21935-9>
sowie im Buchhandel

1 Grundlagen der IP-Netze

Die heutige Gesellschaft kann man sich ohne Internet kaum noch vorstellen. Das Internet ist ein weltweites Rechnernetz, in dem die Daten mit dem sog. *Internet Protocol* (IP) übermittelt werden. Das Internet und alle anderen Netze mit dem Protokoll IP nennt man *IP-Netze*. Die Kommunikation zwischen zwei Rechnern über ein IP-Netz bedeutet aber nicht nur zwei Rechner und IP dazwischen, sondern dahinter verbergen sich sehr komplexe Kommunikationsregeln, die in Form von sog. *Kommunikationsprotokollen* spezifiziert werden.

*Internet als
IP-Netz*

In IP-Netzen bilden alle Kommunikationsprotokolle eine Protokollfamilie, die sog. *Protokollfamilie TCP/IP*. Diese Familie, die sich seit mehr als 30 Jahren entwickelt hat, enthält außer IP und TCP (*Transmission Control Protocol*) eine Vielzahl weiterer Protokolle. Um diese Protokolle systematisch erläutern zu können, ist ein anschauliches Modell sehr hilfreich. Es basiert auf dem sog. *OSI-Referenzmodell* (*Open System Interconnection*), das bereits Ende 70er-Jahre eingeführt wurde.

*Protokoll-
familie
TCP/IP*

Dieses Kapitel schildert in Abschnitt 1.1 kurz die bisherige und zukünftige Entwicklung des Internet und beschreibt in komprimierter Form die Hauptkomponenten des WWW (*World Wide Web*). Abschnitt 1.2 erläutert die grundlegenden Funktionen der Kommunikationsprotokolle. Dem Schichtenmodell für die Darstellung von Prinzipien der Rechnerkommunikation widmet sich Abschnitt 1.3. Allgemeine Prinzipien der Kommunikation in IP-Netzen erläutert Abschnitt 1.4. Die wichtigsten Komponenten der Protokollfamilie TCP/IP präsentiert kurz Abschnitt 1.5. Auf die Internet-Standards und die Struktur der Organisation IETF geht Abschnitt 1.6 ein. Die Schlussbemerkungen in Abschnitt 1.7 runden dieses Kapitel ab.

*Überblick
über das
Kapitel*

In diesem Kapitel werden u.a. folgende Fragen beantwortet:

*Ziel dieses
Kapitels*

- Wie hat sich das Internet bisher entwickelt und welche Trends gibt es bei der Weiterentwicklung?
- Welche Funktionen liegen den Kommunikationsprotokollen zugrunde?
- Wie kann die Kommunikation in IP-Netzen mithilfe eines Schichtenmodells anschaulich dargestellt werden?
- Wie können die verbindungslose und die verbindungsorientierte Kommunikation in IP-Netzen interpretiert werden?
- Welche Bedeutung hat die sog. *Transportschicht* in IP-Netzen mit den Protokollen TCP und UDP?
- Wie koordiniert die IETF die technologische Internet-Weiterentwicklung?

1.1 Entwicklung des Internet

*Es begann
in den 60er-
Jahren*

Die ersten Spuren, die in indirekter Form zur Entstehung des Internet beigetragen haben, führen zurück in die 60er-Jahre. In dieser Zeit wurde zum ersten Mal für die amerikanische Regierung eine Kommunikationsform für den Fall eines nuklearen Krieges erforscht. Die damaligen Überlegungen beinhalten bereits die noch heute geltenden Grundprinzipien der paketvermittelnden Kommunikation. Die Entwicklung des Internet lässt sich grob in folgende Phasen einteilen:

- *Das Internet vor der Nutzung des WWW (World Wide Web):* Aufbau- und Experimentierphase als ARPANET und Verbreitung des Internet vor allem als Forschungs- und Wissenschaftsnetz,
- *Die Schaffung des WWW,*
- *Das Internet nach der Etablierung des WWW* als weltweite Kommunikationsstruktur für wissenschaftliche, private und kommerzielle Nutzung.

1.1.1 Internet vor der Nutzung des WWW

*ARPANET
als Vorläufer
des Internet*

Die Geschichte des Internet ist eng mit der Entstehung des ersten Rechnernetzes im Jahr 1969 auf der Welt eng verbunden. Die Entwicklung dieses Rechnernetzes wurde vom *US Defense Advanced Research Project Agency (DARPA)*, einer Organisation des *Department of Defense (DoD)*, initiiert und es trug den Namen ARPANET (*Advanced Research Project Agency Network*). Abbildung 1.1-1 illustriert den Aufbau von ARPANET.

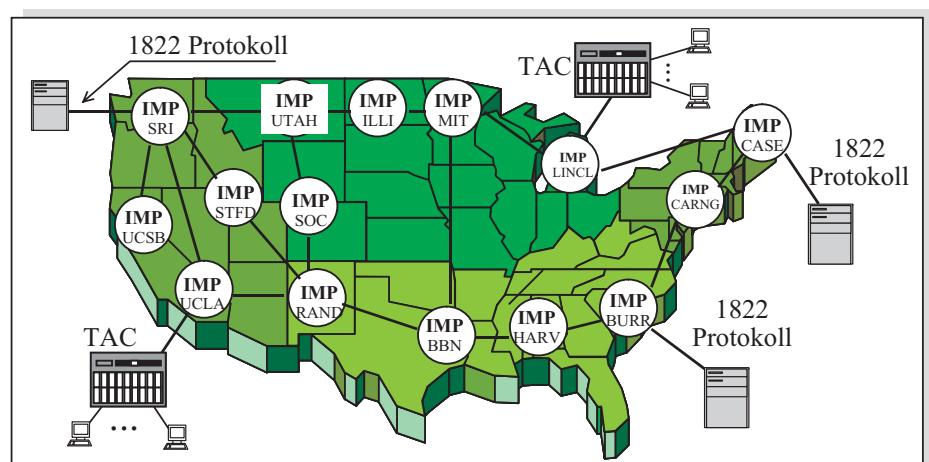


Abb. 1.1-1: Allgemeiner Aufbau von ARPANET
TAC: Terminal Access Controller, IMP: Internet Message Processor

DARPA wollte zunächst digitale Telekommunikation auf Basis einer „packet switching“-Methode über unterschiedliche Netze bereitstellen. Als erster Schritt hierzu wurde am 2. September 1969 an der *University College of Los Angeles (UCLA)* ein Computer an einen sog. *Internet Message Processor (IMP)* angeschlossen. Der IMP war auf der Basis eines Honeywell 516 Rechners der Firma Bolt, Beranek & Newman (BBN) gebaut worden.

Geburt von ARPANET

Anfang der 70er-Jahre wurden die mittlerweile 15 zusammengeschalteten IMPs unter dem Namen *ARPANET* gehandelt. Das Kommunikationsprotokoll der IMPs trug die Bezeichnung *BBN 1822* und kann als Vorläufer von IP gelten. Um *ARPANET* mit anderen Paketnetzen koppeln zu können, wurden 1974 ein Internetwork-Protokoll sowie Gateways entwickelt.

70er-Jahre

Die weitere technische Entwicklung der zunächst *NCP (Network Control Program)* genannten Protokolle wurde vom DARPA entkoppelt und in die Obhut des *Internet Configuration Control Board (ICCB)* gegeben. Mit der 1983 von der *Defense Communication Agency (DCA)* vorgenommenen Trennung des militärisch genutzten Teils des Netzes *MILNET* vom *ARPANET* war ein weiterer wichtiger Schritt für die breite öffentliche Entwicklung des Internet gemacht.

ICCB und NCP

Diese Trennung hatte auch entscheidenden Einfluss auf das Betriebssystem UNIX, das von der Firma AT&T 1969/1970 entwickelt wurde. Wiederum am UCLA wurde in dieses Betriebssystem (genauer: unter UNIX System III) eine Netzwerk-Programmierschnittstelle *Sockets* implementiert, die es erlaubte, eine direkte Rechnerkommunikation mit dem *ARPANET* aufzunehmen. Dieses UNIX wurde als *Berkeley Software Distribution (BSD)* gegen eine geringe Gebühr abgegeben und fand daher schnellen Einzug in Lehre und Forschung. Die weitere Verbreitung von UNIX und Internet sowie ihre technische Fortführung waren die Folge. Nach der ersten Version BSD 4.0 folgte 4.2 und anschließend 4.3, wobei die spätere kommerzielle Weiterentwicklung durch die Firma SUN Microsystems als Betriebssystem SUN OS und später Solaris erfolgte.

BSD und Sockets

Eine 1983 stattfindende Reorganisation des ICCB führte nicht nur zur Konstituierung des *Internet Activity Board (IAB)* anstelle des ICCB, sondern auch zur Festlegung der als Standard geltenden, nun *TCP/IP* genannten Protokollfamilie. Mit der weiteren Entwicklung wurde auch dieser organisatorische Rahmen zu eng. Das IAB wurde zum *Internet Architecture Board* umfirmiert und u.a. um folgende Gremien ergänzt:

IAB und TCP/IP

- IETF (*Internet Engineering Task Force*) als offenes Gremium von Netzwerk-Architekten und -Designern, vor allem aus interessierten Firmen und Einzelpersonen gebildet, um die Entwicklung des Internet zu koordinieren [<http://www.ietf.org>]. Auf die Organisation der IETF geht Abschnitt 1.6 näher ein.
- IESG (*Internet Engineering Steering Group*) mit der Aufgabe, die Tagesaufgaben der IETF zu managen und eine erste technische Stellungnahme zu neuen Internet-Standards zu beziehen [<http://www.ietf.org/iesg.html>].
- IRTF (*Internet Research Task Force*) als Gremium zur Grundlagendiskussion langfristiger Internet-Strategien und -Aufgaben [<http://www.irtf.org>].
- IEPG (*Internet Engineering and Planning Group*), eine offene Arbeitsgruppe von Internet-Systemadministratoren, die dem Ziel verpflichtet sind, einen koordinierten Internet-Betrieb zu gewährleisten [<http://www.iepg.org>].
- ICANN (*Internet Corporation for Assigned Names and Numbers*) mit der Aufgabe, die Verwendung und die Konsistenz der im Internet benutzten Namen, Optionen, Codes und Typen zu regeln und zu koordinieren [<http://www.icann.org>].

Nach der Trennung des militärischen vom zivilen Teil des *ARPANET* wurde dieses zunächst zum Austausch wissenschaftlicher Informationen genutzt und von der *National Science Foundation (NSF)* betreut. Diese baute 1986 den zivilen Teil als nationales Backbone-Netz aus, das als *NSFNet* bekannt geworden ist. Drei Jahre später (1989) waren ca. 100 000 Rechner, die sich an Universitäten und Forschungslabors, in der US-Regierung und in Unternehmen befanden, am

NSFNet

	NSFNet angeschlossen. In nur einem Jahr (1990) hat sich die Anzahl der angeschlossenen Rechner verdoppelt, wobei das NSFNet ca. 3 000 lokale Netze umfasste. Dies war auch der Zeitpunkt, an dem das <i>Domain Name System (DNS)</i> eingeführt wurde.
<i>EARN</i>	Auch in Europa wurden die ersten Ansätze zur Vernetzung der Forschungsinstitute durch <i>EARN (European Academic Research Network)</i> durchgeführt, um die bislang nationalen Netze, wie z.B. BitNet in England und das vom DFN-Verein (<i>Deutsches Forschungsnetz</i>) getragene WiN (<i>Wissenschafts-Netz</i>), miteinander zu koppeln.
<i>USENET</i>	Neben dem direkten, d.h. festgeschalteten und teuren Anschluss ans Internet, wie er bei Universitäten und Forschungseinrichtungen sowie auch bei Firmen üblich ist, wurde bald ein loser Verbund von Systemen – vor allem auf UNIX-Rechnern basierend – aufgebaut, die über Telefonleitungen und Modems gekoppelt waren: das <i>USENET</i> . Hier wurden die Rechner über das Protokoll <i>UUCP (UNIX to UNIX Copy)</i> miteinander verbunden und Nachrichten ausgetauscht. Hauptzweck des USENET war die Verbreitung von E-Mail sowie vor allem von sog. NetNews, die in sog. <i>Newsgroups</i> themenstrukturierte, virtuelle Nachrichtentretter darstellen, in denen zunächst technische Fragen zu Rechnern, Programmiersprachen und dem Internet behandelt wurden. USENET war zeitweise so populär, dass es mit dem Internet selbst identifiziert wurde.
<i>Cyberspace</i>	Die „kopernikanische Wende“ des Internet vollzog sich mit der Einführung des WWW [Abschnitt 1.1.2]. Damit wurde die Möglichkeit geschaffen, „on line“ auf Internet-Ressourcen, sprich Web-Server, zugreifen zu können. Das Internet explodierte, was die Anzahl der Teilnehmer bzw. der Anwender, die Server und die Datenmenge betraf. Das Internet mutierte vom Wissenschaftsnetz zum multimedialen <i>Cyberspace</i> und zum kommerziellen, immer geöffneten Einkaufsparadies.

1.1.2 Die Schaffung des WWW

Der Aufschwung und die umfassende Verbreitung des Internet ist einer Errungenschaft des europäischen Labors für Elementarteilchenforschung *CERN (Conseil Européen pour la Recherche Nucléaire)* in Genf zu verdanken. Mit dem raschen Wachsen und der Internationalisierung der Forschergruppen stellte sich heraus, dass die bisherige Infrastruktur des Internet, das maßgeblich zum Austausch der Forschungsergebnisse genutzt wurde, nicht mehr adäquat war. So wurde nach einem Verfahren gesucht, mit dem die Informationsquellen mittels sog. *Hyperlinks* untereinander direkt verknüpft werden konnten. Der CERN-Mitarbeiter Tim Berners-Lee hatte 1990 die Idee,

- die Dokumente in einer speziellen Seitenbeschreibungssprache *HTML (Hypertext Markup Language)* aufzubereiten und diese untereinander durch Hyperlinks zu verbinden, wobei
- die Dokumenten-Referenzen über einheitliche Adressen *URL (Uniform Resource Locator)* erfolgen sollten und
- die Verknüpfung über ein neues, einfaches Protokoll *HTTP (HyperText Transport Protocol)* abgewickelt werden sollte.

Diese Idee brachte den Vorteil, dass nun nicht mehr der Systemadministrator des Servers, sondern der Dokumenten-Eigentümer für die Verknüpfung der Informationen verantwortlich war (Abbildung 1.1-2). Das nach dieser Idee weltweit verteilte System stellt heute unter dem Namen *World Wide Web (WWW)* –

auch kurz *Web* genannt – die wichtigste Informationsquelle dar. WWW bildet ein weltweites Geflecht (Web) von Knoten, die sog. *Web-Server* repräsentieren und verschiedene Informationen enthalten.

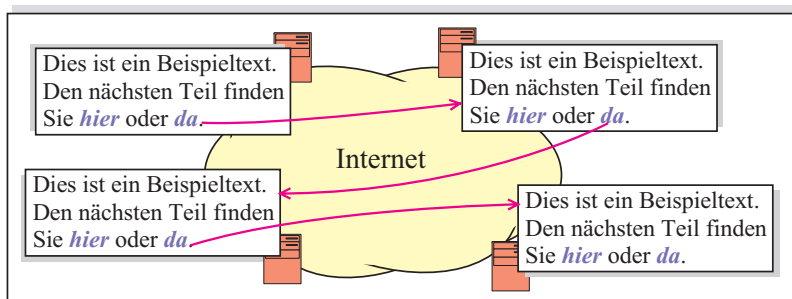


Abb. 1.1-2 Verknüpfung von Dokumenten auf unterschiedlichen Servern mittels Hyperlinks

Zusammen mit seinem Kollegen Robert Cailliau schrieb Tim Berners-Lee den ersten graphischen *Web-Browser* (als Software zur Darstellung der Web-Inhalte) sowie den ersten *Web-Server*. Neben der graphischen Version wurde auch bald eine zeichenorientierte Browser-Version entwickelt, die weitgehend plattformunabhängig war. Mit der Verbreitung von *Web-Browsern* war der Siegeszug des WWW nicht mehr aufzuhalten. Heute spricht man in Bezug auf den Transport der verschiedenen Informationen im WWW vom *Web-Dienst*.

*WWW als
Web-Dienst*

Hauptkomponenten des Web-Dienstes

Der *Web-Dienst* stellt einen *Internet-Dienst* auf grafischer Basis dar, der hauptsächlich zur Informationsabfrage verwendet wird. Die für die Realisierung des *Web-Dienstes* erforderlichen Komponenten zeigt Abbildung 1.1-3.

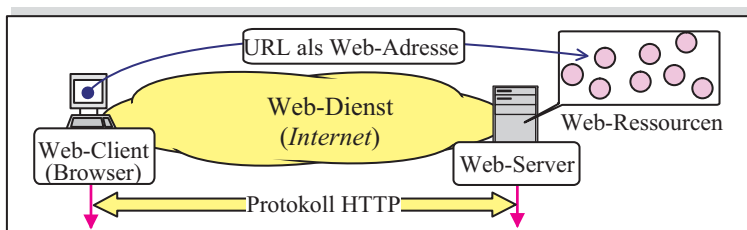


Abb. 1.1-3: Hauptkomponenten des Web-Dienstes

Die Grundkomponenten des *Web-Dienstes* sind:

- Eine Software für die Darstellung von Web-Inhalten in Form von sog. *Web-Seiten* (*Web-Pages*) auf dem Bildschirm des Rechners. Diese Software stellt einen *Web-Client* dar und man bezeichnet sie als *(Web-)Browser*. Ein *Browser* zeigt die angeforderte *Web-Seite* an und bietet zahlreiche Funktionen für die Navigation im *Web-Dienst*.

*(Web-)
Browser*

- URLs als Web-Adressen** ■ Einheitliche *Web-Adressen* zur Angabe der Lokation von Web-Inhalten, die man auch *Web-Ressourcen* nennt. Eine Web-Ressource stellt oft eine Datei in beliebigem Format (wie z.B. HTML, JPEG oder GIF) dar. Als einheitliche Web-Adressen werden sog. *URLs (Uniform Resource Locator)* verwendet. `http://www.hs-fulda.de/fb/ai` ist ein Beispiel hierfür. Die Adressierung von Web-Ressourcen wird noch näher erläutert.
- Web-Server** ■ Web-Server mit *Web-Inhalten (Web-Ressourcen)*, auf die über das Internet zugegriffen werden kann. Die Web-Inhalte werden auch *Web-Content* genannt. Auf einem Web-Server können auch herkömmliche Programme abgespeichert und an den Web-Dienst über eine Software-Schnittstelle, beispielsweise CGI (*Common Gateway Interface*), angebunden werden. Diese Programme können über das Internet aufgerufen werden.
- HTML** ■ Eine abstrakte Sprache für die Beschreibung von *Web-Seiten*. Eine Web-Seite besteht in der Regel aus mehreren Web-Objekten und wird als Hypertext dargestellt. Für die Darstellung von Web-Seiten verwendet man die Seitenbeschreibungssprache *HTML (HyperText Markup Language)*, die in den Jahren 1989/1990 entwickelt wurde. HTML wurde weiterentwickelt und modifiziert, sodass es bereits mehrere HTML-Varianten gibt.
- Protokoll HTTP** ■ Ein Protokoll für den Transport von Web-Inhalten zwischen Browsern und Web-Servern. Hierfür dient das HTTP (*HyperText Transport Protocol*). Hat ein Benutzer eine Web-Seite angefordert (z.B. indem er einen Hyperlink auf dem Bildschirm angeklickt hat), sendet sein Browser die Anforderung (d.h. einen HTTP-Request) an den durch die URL angegebenen Web-Server. Dieser empfängt diese Anforderung und sendet eine Antwort (d.h. einen HTTP-Response), in der sich der angeforderte Web-Inhalt befindet, an den Browser zurück
- HTTP nutzt TCP** Für die Übertragung der Web-Inhalte zwischen Web-Server und -Browser nutzt HTTP das verbindungsorientierte Transportprotokoll TCP (*Transmission Control Protocol*). Dies bedeutet, dass eine TCP-Verbindung für die Übermittlung von Web-Inhalten zwischen Web-Client und -Server aufgebaut werden muss. Das Protokoll TCP wird in Abschnitt 3.3 detailliert beschrieben.

Adressierung von Web-Ressourcen

Um die Lokation einer gewünschten Web-Ressource im Internet anzugeben, braucht man die *Web-Adresse*. Was muss aber eine Web-Adresse angeben und wie sieht sie aus? Abbildung 1.1-4 zeigt, was man beim Web-Dienst zu tun hat.

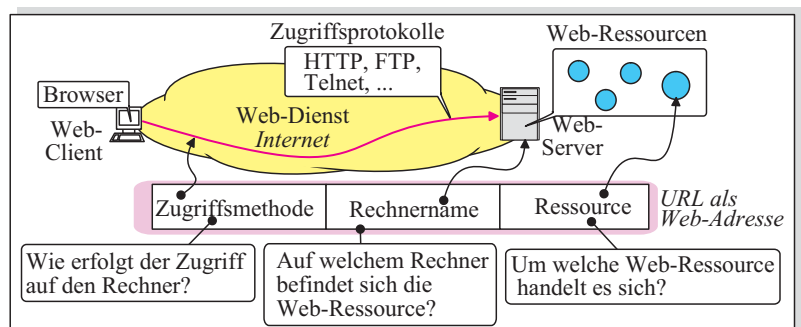


Abb. 1.1-4: Prinzip der Adressierung beim Web-Dienst

Beim Zugriff auf eine Ressource muss Folgendes angegeben werden [BaRS 03]:

- die Art und Weise, wie der Zugriff auf den Web-Server erfolgt, also die Zugriffsmethode, d.h. welches Protokoll (HTTP, FTP, ...) als Zugriffsprotokoll verwendet wird.
- der Rechner, auf dem sich die gewünschte Ressource befindet.
Man muss auf den Rechner verweisen, um ihn eindeutig zu lokalisieren.
- die Ressource, um die es sich handelt.

Was muss eine Web-Adresse enthalten?

1.1.3 Internet nach der Etablierung des WWW

Das Internet ist nach der Geburt des WWW ein so komplexes weltweites Rechnernetz geworden, dass es nicht möglich ist, hier die Struktur seiner physikalischen Vernetzung zu zeigen. Sie ist unbekannt und wächst ständig. Das Internet ist aber nach einem hierarchischen Prinzip aufgebaut. Wie Abbildung 1.1-5 illustriert, stellt das Internet eine Vernetzung von Rechnern dar, in der man mehrere Schichten unterscheiden kann.

Internet-Strukturierung

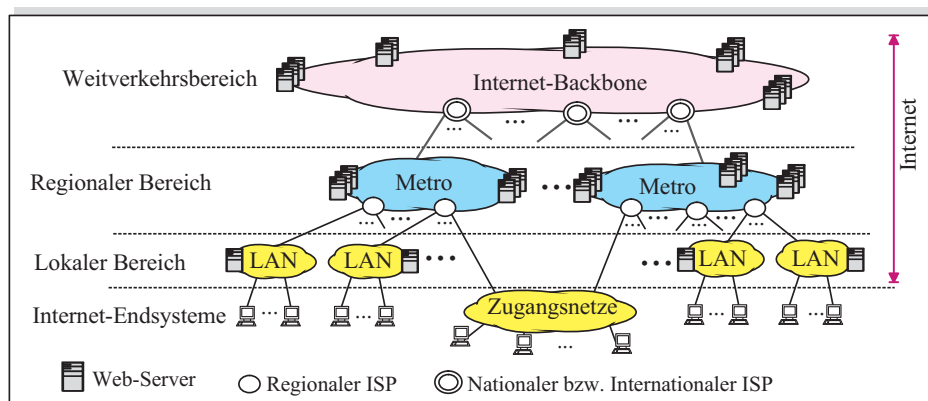


Abb. 1.1-5: Allgemeine Internet-Strukturierung

ISP: Internet Service Provider; LAN: Local Area Network; Metro: Metro(politan)-Netz

Die untere Schicht bilden lokale Netzwerke (LANs) mit den Web-Servern, die den privaten Firmen, öffentlichen Institutionen, Hochschulen und anderen Organisationen gehören; sie können als *lokaler Internet-Bereich* angesehen werden. Die mittlere Schicht bilden regionale Netze mit regionalen Internet-Diensteanbietern, sog. ISPs (*Internet Service Provider*). Diese Schicht stellt den *regionalen Internet-Bereich* dar. Bei den regionalen Netzen handelt es sich in der Regel um Hochgeschwindigkeitsnetze innerhalb von Großstädten, weshalb man sie als *Metro-Netze* bzw. *City-Netze* bezeichnet. Die obere Schicht, die den Internet-Weitverkehrsbereich darstellt, bilden nationale und internationale

Hochgeschwindigkeitsnetze mit nationalen bzw. internationalen ISPs. Die nationalen und internationalen Hochgeschwindigkeitsnetze werden miteinander gekoppelt und bilden das sog. *Internet-Backbone*.

Jeder ISP stellt einen Internet-Zugangspunkt dar, der auch als *Einwahlknoten* bzw. als *POP (Point of Presence)* bezeichnet wird.

1.1.4 Die Zukunft des Internet

Das Internet wächst und entwickelt sich weiter. Es ist hier nicht möglich, alle Entwicklungen darzustellen. In kurzer Form werden in Abbildung 1.1-6 nun die wichtigsten Trends aufgezeichnet.

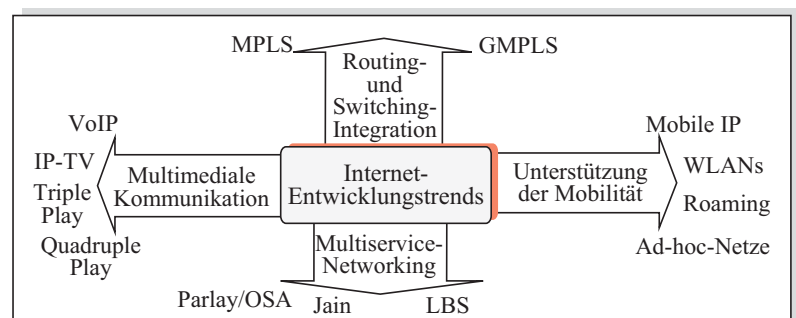


Abb. 1.1-6: Allgemeine Entwicklungstrends in Internet und IP-Netzen

Routing und Switching Integration

In lokalen Netzwerken werden seit einigen Jahren sowohl Router als auch Switches eingesetzt, sodass man von Routing und Switching Integration spricht. Diese Integration findet auch im Backbone des Internet und in großen privaten IP-Netzen statt. Hierfür wurde das Konzept MPLS (*Multi-Protocol Label Switching*) entwickelt [Abb. 1.4-4]. MPLS wird in Abschnitt 11.2 beschrieben.

MPLS, GMPLS

Bei MPLS werden die IP-Pakete über ein Netz als Gänsemarsch übermittelt [Abb. 11.1-1]. Dadurch entsteht die Möglichkeit, die Dienstgüte (*Quality of Service*) auf einem geforderten Level zu garantieren. Dies ist für die multimediale Kommunikation von enormer Bedeutung. Um das MPLS-Konzept in optischen Netzen, in denen man WDM (*Wavelength Division Multiplexing*) verwendet, einsetzen zu können, wurde GMPLS (*Generalized MPLS*) entwickelt [Abschnitt 11.3]. Bei IP-Netzen mit MPLS spricht man von *Next Generation IP Networks*.

Multiservice-Networking

Als wichtiger Trend bei IP-Netzen kann das Multiservice-Networking angesehen werden. Ihm liegt die *Integration (Konvergenz) der Netze* zugrunde. Da verschiedene TK-Netze (wie PSTN, ISDN, GSM, UMTS) konvergieren, ist es wünschenswert, die Möglichkeit zu haben, alle TK-Netze seitens des Internet

als ein heterogenes TK-Netz zu nutzen, intelligente Netzdienste auf Basis des Protokolls IP zu entwickeln und diese den Teilnehmern an allen TK-Netzen über das Internet zugänglich zu machen.

Um intelligente Netzdienste auf Basis eines TK-Netzes zu entwickeln, muss man auf bestimmte Software-Schnittstellen, sog. APIs (*Application Programming Interface*), im Netzkern zugreifen. Da diese nur für den Netzbetreiber zugänglich sind, war es bisher nicht möglich, dass die Netzdienste durch Dritte, also durch die Nicht-Netzbetreiber, konzipiert und entwickelt werden konnten. Um dies zu ändern, wurde ein vom Netz unabhängiges API entwickelt. Es handelt sich um Parlay/OSA (*Open Service Architecture*).

Idee von Parlay/OSA: Die grundlegende Idee von Parlay/OSA besteht darin, dass man verschiedene TK-Netze als Kernnetz betrachtet. Die Dienste dieses Kernnetzes sind über Parlay/OSA API für die Nicht-Netzanbieter zugänglich. Somit können sie Netzanwendungen entwickeln und auf speziellen Application Servern installieren. Ein Application Server am Internet bzw. am privaten IP-Netz (Intranet) stellt bestimmte Netzanwendungen zur Verfügung, auf die man über die verschiedenen TK-Netze zugreifen kann. Solche Netzanwendungen werden auch als *New Generation Network Services* bezeichnet

Parlay/OSA

Ein ähnliches Konzept wie bei Parlay/OSA wird auch bei JAIN (*Java API for Integrated Networks*) von der Firma Sun Microsystems verfolgt.

JAIN

Bei der Nutzung von Parlay/OSA kann jeder seine Kreativität entfalten und beliebige Netzdienste entwickeln und sie über verschiedene Netze und das Internet zugänglich machen. Insbesondere sind hier neue Dienste auf Basis von *Location Based Services* (LBS) zu nennen.

LBS

Mit Multiservice-Networking hängt die Realisierung der multimedialen Kommunikation zusammen. Die Entwickler träumen seit geraumer Zeit davon, über ein Netz zu verfügen, über das alle Informationsarten (Audio, Video und Daten) übermittelt werden können. Die Konzepte und Protokolle für VoIP (*Voice over IP*) sind ein wichtiger Schritt in diese Richtung. Durch die Einführung der geeigneten Protokolle für die Realisierung von Multicasting [Abb. 9.6-1] sind solche Dienste wie IP-Radio und IP-Fernsehen im Kommen. Bereits seit 2005 spricht man von *Triple Play*. Darunter versteht man das gebündelte Angebot der drei Dienste *Internet*, *IP-Telefonie* (VoIP) und *Fernsehen* für private Haushalte. Beim Dienst Fernsehen handelt es sich zurzeit noch um das klassische Fernsehen. Der nächste Schritt kann die Erweiterung von Triple Play um die Mobilfunkdienste sein, genauer gesagt um die UMTS-Dienste. Dieses Angebot, bestehend aus vier Diensten, wird *Quadruple Play* genannt.

Multimediale Kommunikation

Die Unterstützung der Mobilität in IP-Netzen ist seit vielen Jahren ein großes Thema. Aber erst durch die breite Einführung von WLANs (*Wireless LANs*) und UMTS-Diensten hat dieses Thema an Bedeutung zugenommen. Es werden Konzepte entwickelt, um sog. *Ad-hoc-Netze*, in denen sowohl die Knoten als auch die Endsysteme mobil sind, zu realisieren. Kapitel 13 widmet sich der Unterstützung der Mobilität in IP-Netzen.

Unterstützung der Mobilität

1.2 Funktionen der Kommunikationsprotokolle

*Fehler-
ursachen*

In einem Netz können die zu übertragenden Daten verfälscht werden. Die Ursachen dafür sind meist auf die schlechte Qualität des Übertragungsmediums zurückzuführen. Eine Verfälschung der Daten kann auch durch äußere Einflüsse wie etwa starke elektromagnetische Felder in der Umgebung oder durch das sog. Nebensprechen entstehen. Übertragungsstörungen führen nicht nur zu einer Datenverfälschung, sondern sogar zu einem Datenverlust. Um dies zu vermeiden, müssen entsprechende Funktionen in den Kommunikationsprotokollen enthalten sein. Diese Funktionen lassen sich in drei Gruppen aufteilen:

- Fehlerkontrolle (Fault Control),
- Flusskontrolle (Flow Control) und
- Überlastkontrolle (Congestion Control).

*Datenver-
fälschungen
und -verluste*

Die *Fehlerkontrolle* umfasst alle Maßnahmen in einem Kommunikationsprotokoll, mit denen Datenverfälschungen und -verluste während der Übertragung entdeckt und beseitigt werden können. Die *Flusskontrolle* bedeutet eine gegenseitige Anpassung der Send- und der Empfangsseite in Bezug auf die übertragene Datenmenge. Die *Überlastkontrolle* betrifft alle Vorkehrungen, die dazu dienen, ein Netz nicht zu überlasten. Bei der Überlastung eines Netzes müssen die übertragenen Datenblöcke oft verworfen werden und die Verweilzeit von Datenblöcken im Netz durch „Staus“ in Knoten nimmt stark zu. Im Folgenden werden diese Funktionen näher erläutert.

1.2.1 Prinzipien der Fehlerkontrolle

Die Fehlerkontrolle hat die Aufgabe, jede fehlerhafte Situation während der Datenübertragung zu entdecken und zu beseitigen. Sie ist Bestandteil jedes Kommunikationsprotokolls und wird beim Empfänger mithilfe von festgelegten *Quittungen (Bestätigungen)* und beim Sender durch die Zeitüberwachung realisiert. Im Weiteren werden alle möglichen Fehlersituationen dargestellt und notwendige Maßnahmen zu ihrer Beseitigung aufgezeigt.

Allen Kommunikationsprotokollen liegen zwei „eiserne Regeln“ zugrunde – hier die erste:

*Erste
„eiserne
Regel“*

Datenblöcke können während der Übertragung verfälscht werden. Deshalb muss nach dem Absenden jedes Datenblockes eine Kopie davon auf der Quellstation gehalten werden, für den Fall, falls eine wiederholte Übertragung notwendig werden sollte.

Datenblöcke haben in verschiedenen Kommunikationsprotokollen unterschiedliche Namen; oft werden sie als *Datenpakete*, kurz *Pakete*, oder *Frames* bezeichnet. Hier wird einheitlich das Wort *Datenblock* verwendet.

Negative Auswirkungen infolge der Verfälschung von übertragenen Datenblöcken können durch die Umsetzung dieser Regel und durch eine wiederholte Übertragung ausgeglichen werden. Abbildung 1.2-1a zeigt die fehlerlose Übertragung eines Datenblocks. Diese wird von der Empfangsseite positiv quittiert (bestätigt) und eine Kopie des Datenblocks in der Quellstation gelöscht. Auch eine Quittung stellt einen kurzen, vom Protokoll festgelegten Datenblock dar.

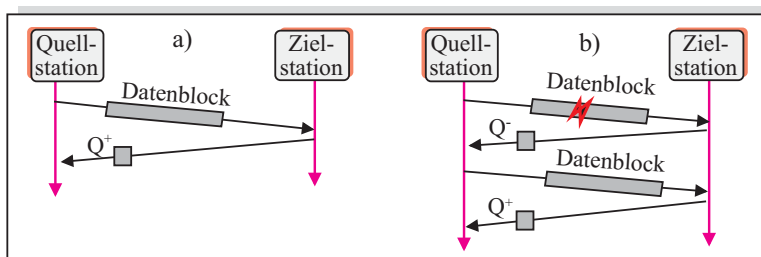


Abb. 1.2-1: Übertragung eines Datenblocks: a) fehlerlos, b) fehlerhaft
Q⁺: positive Quittung, Q⁻: negative Quittung

In Abbildung 1.2-1b tritt bei der Übertragung des Datenblocks eine Störung auf, was eine negative Quittierung zur Folge hat. Der gestörte Datenblock wird durch die Zielstation einfach verworfen. Da in der Quellstation eine Kopie des betreffenden Datenblocks gehalten wird, sendet die Quellstation den gleichen Datenblock noch einmal – diesmal fehlerfrei – zu der Zielstation, die ihn positiv quittiert. Die Kopie des übertragenen Datenblocks kann nun in der Quellstation gelöscht werden.

*Negative
Quittung bei
Störungen*

Eine besondere Situation entsteht dadurch, dass nicht nur die Datenblöcke während der Übertragung verfälscht werden können, sondern auch die Quittungen. Wird eine positive Quittung so verfälscht, dass die Quellstation sie als negative Quittung interpretiert, führt dies zu einer unnötigen wiederholten Übertragung des betreffenden Datenblocks und zur Verdoppelung von Daten am Ziel.

*Verfälschung
von
Quittungen*

Der schlimmste Fall (*worst case*) bei der Übertragung eines Datenblocks entsteht dann, wenn sowohl der übertragene Datenblock als auch dessen negative Quittung verfälscht werden. Wie Abbildung 1.2-2 zeigt, empfängt die Quellstation in diesem Fall eine positive Quittung und könnte deshalb die Kopie des Datenblocks löschen. Dies würde aber zum Verlust des Datenblocks führen. Um einen solchen Fall zu bewältigen, müssen die Kommunikationsprotokolle zwei Stufen der Fehlerkontrolle realisieren. Die hier angesprochene Fehlerkontrolle bezieht sich nur auf die Übertragung einzelner Datenblöcke, die oft aufgrund der Segmentierung von zu übertragenden Dateien entstehen.

Die Fehlerkontrolle muss auch auf Dateiniveau realisiert werden. Die einzelnen Datenblöcke, die zu einer Datei gehören, werden in der Zielstation zu einer Datei zusammengesetzt. Ist ein Datenblock der Datei in der Zielstation nicht vor-

handen, sendet sie eine negative Quittung, die sich auf diese Datei bezieht. Die Quellstation muss dann entweder den verloren gegangenen Datenblock oder sogar die ganze Datei nachsenden.

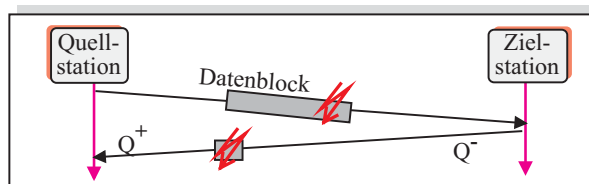


Abb. 1.2-2: Worst case einer Datenübertragung: Daten und Quittung werden verfälscht

Die zweite „eiserne Regel“, die bei allen Kommunikationsprotokollen realisiert werden muss, um Datenverluste während der Übertragung zu erkennen, lautet:

Zweite
„eiserne
Regel“

Datenblöcke können bei der Übertragung verloren gehen, sodass man nur eine begrenzte Zeit auf eine positive oder negative Quittung für einen Datenblock warten soll.

Dies muss mithilfe einer Zeitüberwachung realisiert werden, um die Verluste von Datenblöcken während der Übertragung zu erkennen. Dazu ist im Protokoll eine maximale Wartezeit auf eine Quittung festzulegen. Eine solche Wartezeit wird auch als *time out* bezeichnet und kann als „Geduldzeit“ interpretiert werden. Nach dem Absenden eines Datenblocks muss die Überwachung der maximalen Wartezeit auf die Quittung aktiviert werden. Es stellt sich die Frage, wann die Datenblöcke während der Übertragung eigentlich verloren gehen. Dass ein übertragener Datenblock bei einem plötzlichen Bruch der Leitung verloren geht, ist selbstverständlich, doch das ist selten der Fall. Die häufigste Ursache für den Verlust eines Datenblocks ist eine Verfälschung in seinem Header oder Trailer, sodass er auf der Leitung nicht vollständig erkannt und damit in der Zielstation nicht aufgenommen werden kann.

Geduldzeit

Abbildung 1.2-3a illustriert die fehlerhafte Situation, in der ein Datenblock verloren gegangen ist. Nach dem Absenden des Datenblocks wird die „Geduldzeit“ überwacht. Kommt innerhalb dieser Zeit keine Quittung an, interpretiert dies die Quellstation als verloren gegangenen Datenblock und wiederholt die Übertragung. Nach dem wiederholten Absenden kommt eine positive Quittung noch während der „Geduldzeit“ an und die Kopie des Datenblocks kann dann gelöscht werden.

Nummerierung von Datenblöcken

Auch eine Quittung kann verloren gehen. Wie Abbildung 1.2-3b zeigt, wird dies ebenfalls mithilfe der Zeitüberwachung erkannt. In einem solchen Fall kann ein Datenblock in der Zielstation doppelt vorhanden sein. Deswegen muss für die Zielstation klar werden, dass es sich nicht um einen neuen Datenblock handelt, sondern um eine wiederholte Übertragung. Werden die übertragenen

Datenblöcke nicht nummeriert, kann das zur Verdopplung von Daten am Ziel führen. Derartige Datenverdopplungen lassen sich mit der Nummerierung von Datenblöcken ausschließen. Aus diesem Grund werden bei allen Kommunikationsprotokollen die übertragenen Datenblöcke nummeriert.

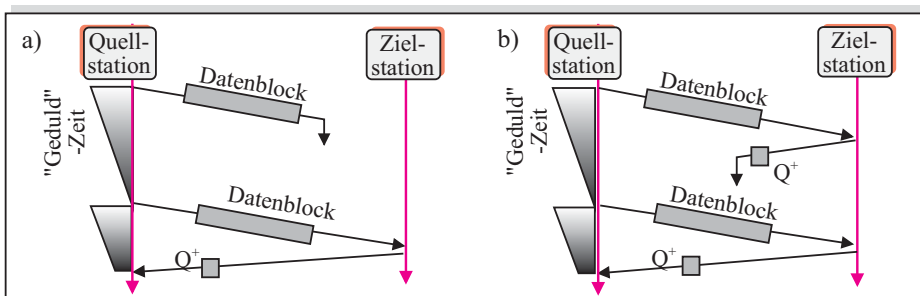


Abb. 1.2-3: Fehlerhafte Übertragung: a) Datenblockverlust, b) Quittungsverlust

Bei einer fortlaufenden Nummerierung der übertragenen Datenblöcke kann die Zielstation erkennen, ob es sich um eine wiederholte Übertragung handelt und somit einen doppelt vorhandenen Datenblock entdecken. Eine Nummerierung der übertragenen Datenblöcke besteht darin, dass jedem Datenblock eine bestimmte Sequenznummer zugeteilt wird. Diese Nummerierung kann aber nicht beliebig fortgesetzt werden. Ursache hierfür ist die begrenzte Anzahl von Bits für die Nummernabspeicherung im Header des Datenblocks und deshalb werden die Datenblöcke nach dem Modulo-Verfahren nummeriert. In den meisten Fällen wird die Nummerierung nach dem Modulo 8 oder 128 realisiert. Bei Modulo 8 werden die einzelnen Datenblöcke von 0 bis 7 gekennzeichnet und verschickt. Ist die 7 als letzte Nummer vergeben worden, wird der Zähler zurückgesetzt und die Nummerierung startet bei 0. Äquivalent dazu funktioniert die Nummerierung nach dem Modulo-128-Verfahren, bei dem die Nummern bis 127 vergeben werden.

Modulo-Verfahren

Bei der Nummerierung von Datenblöcken kann eine Gruppe von empfangenen Blöcken gleichzeitig durch die Zielstation quittiert werden, um damit die Verkehrslast im Netz durch eine geringere Anzahl von Quittungen zu reduzieren.

Beim Aufbau einer Verbindung muss sichergestellt sein, dass die Quellstation den Datenblöcken jene Sequenznummern zuteilt, die auch von der Zielstation erwartet werden. Aus diesem Grund ist zu vereinbaren, welchen Zahlenwert das Nummerierungsfenster hat und mit welcher Sequenznummer bei der Übertragung der Datenblöcke begonnen wird.

Nummerierungsfenster (Window)

1.2.2 Realisierung der Flusskontrolle

Bei der Datenkommunikation tritt häufig der Fall ein, dass die Daten beim Sender rascher „produziert“ werden, als der Empfänger sie „konsumieren“ kann. So ist eine Situation vorstellbar, in der ein Großrechner im Netz eine große Menge von Daten an einen entfernten kleinen Drucker übermittelt. Der Großrechner muss, um ein Überfließen des Druckerspeichers zu verhindern,

Bedeutung der Flusskontrolle

die Menge der zu übertragenden Daten der Aufnahmefähigkeit des Druckers anpassen. Die Anpassung muss durch entsprechende Kommandos vom Drucker gesteuert werden. Dieses einfache Beispiel weist auf die Bedeutung der gegenseitigen Abstimmung zwischen Quell- und Zielstation in Bezug auf die Menge der zu übertragenden Daten hin.

Ziel der Flusskontrolle

Unter *Flusskontrolle* versteht man alle Maßnahmen, die der Anpassung der gesendeten Datenmenge der Quellstation an die Aufnahmekapazität der Zielstation dienen. Die Flusskontrolle kann realisiert werden

- mithilfe der Meldungen `Halt` und `Weitersenden`,
- mithilfe von Krediten und
- über einen Fenstermechanismus (Window).

Meldungen: Halt, Weitersenden

Die einfache Flusskontrolle mithilfe der Meldungen `Halt` und `Weitersenden` verläuft wie folgt: Stellt der Empfänger fest, dass er nicht mehr in der Lage ist, die empfangenen Daten aufzunehmen, schickt er dem Sender die Meldung `Halt`. Der Sender ist nach dem Empfang von `Halt` verpflichtet, das Senden von Daten einzustellen, bis der Empfänger die Meldung `Weitersenden` übermittelt und damit den `Halt`-Zustand aufhebt. Ein Nachteil dieses einfachen Verfahrens besteht darin, dass eine Verfälschung der Meldungen `Halt` oder `Weitersenden` besondere Konsequenzen hat: Wird `Halt` während der Übertragung verfälscht und vom Sender als `Weitersenden` empfangen, so sendet er die Daten weiter. Kommt `Weitersenden` beim Sender als `Halt` an, wird der Sendeprozess auf Dauer gestoppt.

Flusskontrolle mittels Krediten

Bei einer Flusskontrolle mithilfe von Krediten erteilt der Empfänger dem Sender einige Kredite für die Übermittlung von Datenblöcken. Sind diese Kredite aufgebraucht, muss der Sender die Übertragung einstellen. Ein Kredit definiert eine Anzahl von Datenblöcken, d.h. deren Sequenznummer, die der Sender abschicken darf, ohne auf eine Quittung vom Empfänger warten zu müssen. Hierbei ist die maximale Länge der Datenblöcke festgelegt. Im Normalfall werden die Kredite laufend erteilt, sodass ein ununterbrochener Datenverkehr aufrechterhalten werden kann.

Die Übertragung von Krediten muss vor Störungen geschützt werden. Bei der Störung einer Kreditmeldung könnte der Sender ohne weitere Kredite bleiben und der Empfänger auf weitere Datenblöcke warten. Damit wäre die Datenübermittlung blockiert. Es muss sichergestellt sein, dass eine Kreditmeldung nicht verdoppelt wird. Wäre dies nicht der Fall, könnte der Sender weitere Datenblöcke senden, die vom Empfänger nicht aufgenommen werden könnten.

Flusskontrolle über Fenstermechanismus

Die Flusskontrolle über einen Fenstermechanismus stützt sich auf die Sequenznummern von übertragenen Datenblöcken. Vor der Datenübermittlung sprechen sich Quell- und Zielstation über ein sog. *Fenster* innerhalb des Wertebereiches der Sequenznummern ab. Die Fenstergröße W bedeutet: *Die Quellstation darf maximal W Datenblöcke absenden, ohne auf eine Quittung warten zu müssen, d.h. W ist bei der Quellstation als Anzahl der Kredite zu interpretieren.* Bei der Zielstation stellt W die Kapazität des Empfangspuffers für die ankommenden Datenblöcke dar.

Beispiel: Abbildung 1.2-4 zeigt den Fall, in dem die Fenstergröße $W = 3$ und die Datenblöcke nach dem Modulo-8-Verfahren nummeriert werden. Bei $W = 3$ darf die Quellstation drei Datenblöcke absenden, ohne auf eine Quittung warten zu müssen. Abbildung 1.2-4 zeigt gleichzeitig die freie Sequenznummer, die der Sender für die Nummerierung verwenden darf. Da $W = 3$, sind maximal drei Nummern zu vergeben. Wie hier ersichtlich, ist während der Übertragung der ersten drei Datenblöcke keine Quittung angekommen, also muss die Quellstation den Sendeprozess unterbrechen. Dies führt zu einer Senderblockade. Nach der ersten positiven Quittung, mit der die Datenblöcke mit den Nummern 0 und 1 positiv quittiert wurden, darf sie zwei weitere Daten-

blöcke senden. Dieses Beispiel zeigt, welche Auswirkungen die Fenstergröße auf die Auslastung des Übertragungsmediums hat. Insbesondere im Fall $w = 1$ muss man nach dem Absenden jedes Datenblocks den Sendeprozess stoppen. Dies führt selbstverständlich zu einer schlechten Ausnutzung des Übertragungsmediums.

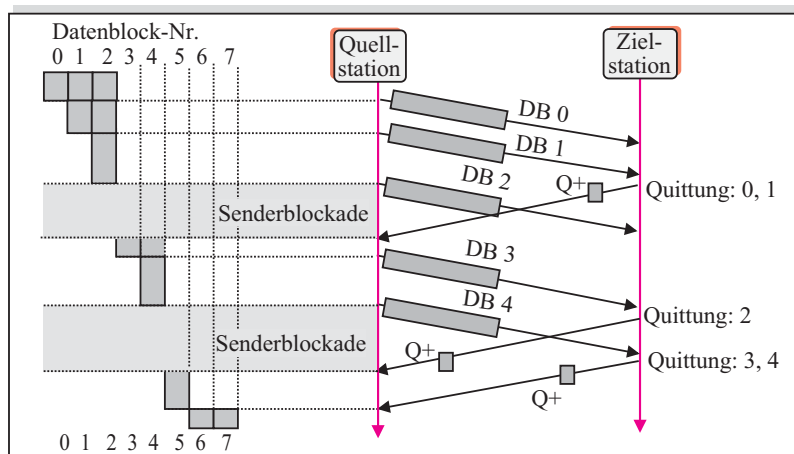


Abb. 1.2-4: Veranschaulichung der Flusskontrolle über den Fenstermechanismus

Die Fenstergröße kann als Kredit für die Vergabe von Nummern für die abzusendenden Datenblöcke interpretiert werden. Die meisten Kommunikationsprotokolle realisieren die Flusskontrolle nach dem Fenstermechanismus.

1.2.3 Überlastkontrolle

Ein Netz hat eine bestimmte Aufnahmekapazität, d.h. zu jedem Zeitpunkt kann sich darin nur eine begrenzte Anzahl von Datenblöcken befinden. Wird diese Anzahl überschritten, hat dies die folgenden negativen Auswirkungen:

- Die Aufnahmepuffer im Netz (in Knoten) sind voll; dies führt dazu, dass die im Netz eintreffenden Datenblöcke verworfen werden müssen.
- Es bilden sich Warteschlangen von Datenblöcken vor den Übertragungsleitungen; durch die so verursachten großen Verweilzeiten der Datenblöcke im Netz entstehen große Verzögerungen der übertragenen Datenblöcke.

Die Maßnahmen, mit denen eine Überlastung des Netzes verhindert wird, bezeichnet man als *Überlastkontrolle (Congestion Control)*.

Congestion Control

Die wichtigsten Kriterien für die Beurteilung der Überlastung von Netzen sind:

- Durchsatz (Throughput) und
- Datenverweilzeit (Latenzzeit, Delay) im Netz.

Durchsatz Unter dem *Durchsatz eines Netzes* versteht man den Anteil des Datenverkehrs, der von dem Netz akzeptiert wird. Den Verlauf des Durchsatzes in Abhängigkeit vom Gesamtdatenverkehr zeigt Abbildung 1.2-5a. Ist der Datenverkehr im Netz klein (kleine Belastung), werden alle ankommenden Daten durch das Netz aufgenommen; dabei müssen normalerweise keine Vorkehrungen gegen die Überlast ergriffen werden. Bei hoher Netzbelastung dagegen müssen bestimmte Maßnahmen getroffen werden, um eine Überlastung zu vermeiden, und sie führen zur Einschränkung der Datenmenge, die ins Netz gesendet wird.

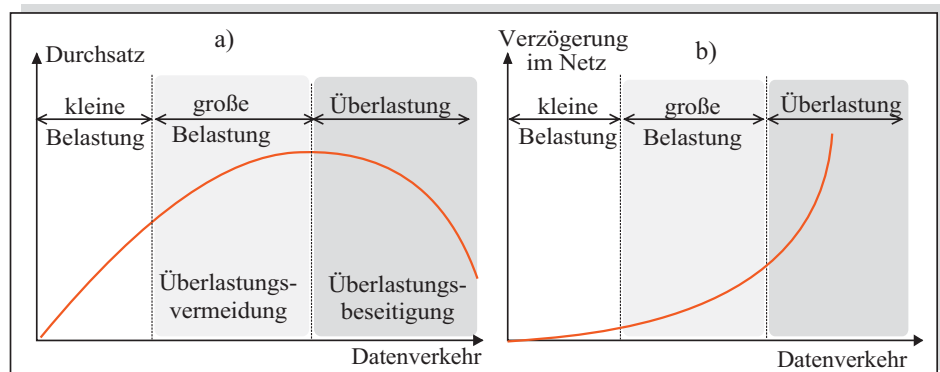


Abb. 1.2-5: Auswirkungen der Netzüberlastung:
a) auf den Durchsatz, b) auf die Datenverweilzeit im Netz

Ist der Datenverkehr im Netz so groß, dass das Netz überlastet ist, müssen andere Aktionen eingeleitet werden, um die bestehende Überlastung zu beseitigen. Wie in Abbildung 1.2-5a ersichtlich, nimmt der Durchsatz in der Überlastsituation mit zunehmendem Datenverkehr sehr stark ab.

Verweilzeit im Netz

Abbildung 1.2-5b veranschaulicht, welche Auswirkungen die Netzbelastung auf das Verhalten der Datenverweilzeit (Latenzzeit) im Netz hat. In einer Überlastsituation muss also mit großen Verzögerungen für die Datenübertragung im Netz gerechnet werden. Die wichtigste Maßnahme für die Vermeidung von Überlasten besteht in der Einschränkung der Datenströme, die ins Netz fließen. Welche Maßnahmen gegen die Überlastung in einzelnen Netzen und Kommunikationsprotokollen ergriffen werden, hängt auch von der Realisierung der Flusskontrolle ab.

1.3 Schichtenmodell der Kommunikation

Was ist OSI? Als man Mitte der 70er-Jahre versuchte, die Rechner unterschiedlicher Hersteller miteinander zu vernetzen, hat sich folgendes Problem ergeben: Es sind dringend Kommunikationsregeln nötig, damit ein Rechner des Herstellers X

mit einem Rechner des Herstellers *Y* kommunizieren kann. Es sollte möglich sein, dass jeder Rechner für die Kommunikation mit allen anderen Rechnern *offen* (bereit) ist. In diesem Zusammenhang wurde bereits damals von der Vernetzung offener Systeme – also von *Open System Interconnection (OSI)* – gesprochen und nach einem Modell für ihre Verwirklichung gesucht.

Daher wurde ein Schichtenmodell eingeführt, das die Prinzipien der Kommunikation zwischen verschiedenen Systemen beschreibt und die OSI-Vorstellung ermöglicht. Es wird deshalb *OSI-Referenzmodell* genannt. Standardisiert wurde es von ISO (*International Organization for Standardization*) und es wird auch als *ISO/OSI-Referenzmodell* bzw. kurz als *ISO/OSI-Modell* bezeichnet.

OSI-Referenzmodell

1.3.1 Konzept des OSI-Referenzmodells

Die Idee von OSI illustriert Abbildung 1.3-1. Gemäß OSI wird ein Rechner als *offenes System* angesehen. Diese Systeme werden durch Übertragungsmedien untereinander verbunden und enthalten entsprechende Kommunikationsprotokolle, nach denen logische Verbindungen zwischen Applikation in den einzelnen Systemen nach Bedarf aufgebaut werden können.

Idee von OSI

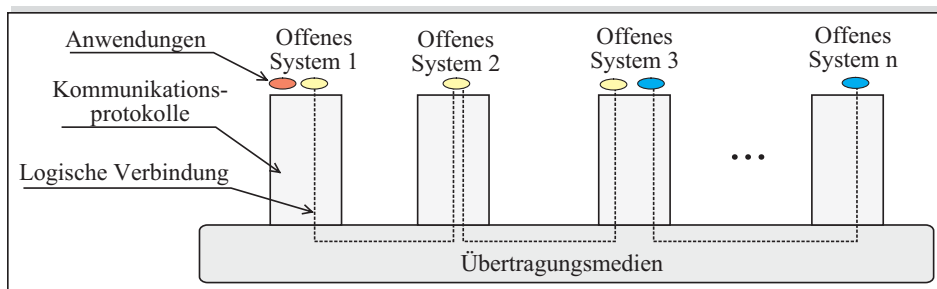


Abb. 1.3-1: Idee von OSI: Jedes System soll mit jedem anderen kommunizieren können

Um die Kommunikationsprotokolle für die Verwirklichung der Zielvorstellung von OSI zu entwickeln, wurde die komplexe Aufgabe der Kommunikation zwischen verschiedenen Systemen so auf sieben Teilaufgaben verteilt, dass diese den einzelnen Schichten, die in einer Hierarchie zueinander stehen, zugeordnet werden. Dadurch ist ein OSI-Referenzmodell mit sieben Schichten entstanden; man spricht hier auch vom *OSI-Schichtenmodell*.

Ein Rechnernetz enthält aber nicht nur die Rechner als Endsysteme, sondern auch die Netzknoten (Router, Switches) als *Zwischensysteme*. Die Aufgabe der Kommunikation in Zwischensystemen kann aber zu drei untereinander liegenden Schichten zusammengefasst werden. Daher enthalten die Zwischensysteme nur die ersten drei Schichten. Abbildung 1.3-2 zeigt die allgemeine Struktur

Allgemeines OSI-Schichtenmodell

des OSI-Referenzmodells. Die unterste Schicht 1 repräsentiert die physikalische Netzanbindung, also die Übertragungstechnik. Die Schichten von 2 bis 6 repräsentieren bestimmte Funktionen der Kommunikation. Schicht 7 enthält die Anwendungen (Applikationen).

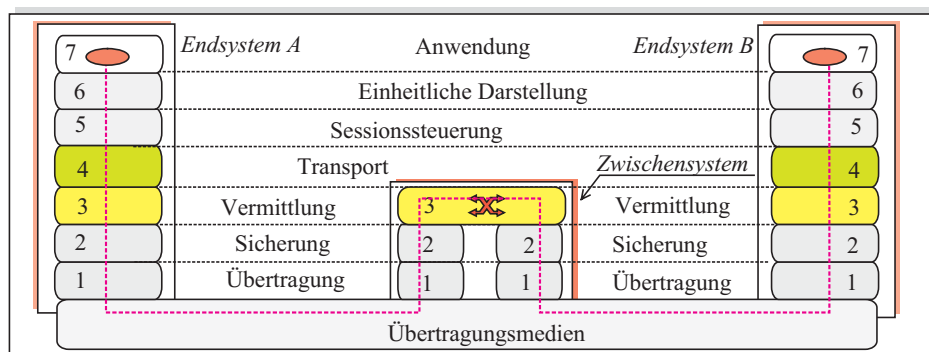


Abb. 1.3-2: OSI-Referenzmodell: Zerlegung der Kommunikationsaufgabe in 7 Schichten

Funktionen der Schichten

Die einzelnen Schichten im OSI-Referenzmodell sind:

1. Physikalische Schicht (Übertragungsschicht, Physical Layer)

Sie definiert die mechanischen und elektrischen Eigenschaften sowie die Funktionen und die Abläufe bei der Bitübertragung.

2. Sicherungsschicht (Data Link Layer)

Diese Schicht garantiert eine sichere Übertragung zwischen zwei direkt benachbarten Stationen. Dazu werden die übertragenen Bits in sog. *Frames (Rahmen)* zusammengefasst und am Ende mit einer Prüfsumme versehen. Dadurch ist eine Fehlererkennung möglich. In LANs wird die Schicht 2 in zwei Teilschichten aufgeteilt: *Schicht 2a* als *MAC-Schicht (Media Access Control)*, die den Zugriff auf das Übertragungsmedium regelt, und *Schicht 2b* als *LLC-Schicht (Logical Link Control)* [Abb. 10.1-1], die eine Sicherungsschicht darstellt.

3. Vermittlungsschicht (Netzwerkschicht, Network Layer)

Diese Schicht hat die Aufgabe, die Daten blockweise zwischen Endsystemen zu übermitteln. Die innerhalb dieser Schicht übertragenen Datenblöcke werden oft *Pakete* genannt. Schicht 3 stellt eine *Paketvermittlungsschicht* dar.

4. Transportschicht (Transport Layer)

Die Transportschicht hat u.a. die Aufgabe, eine gesicherte virtuelle Ende-zu-Ende-Verbindung für den Transport von Daten zwischen den Endsystemen bereitzustellen. Die Aufgaben der Transportschicht bestehen vor allem in der Korrektur der Übermittlungsfehler und sind von den Protokollen der Schicht 2 und 3 sehr stark abhängig.

5. Sitzungsschicht (Sessionsschicht, Session Layer)

Sie ist die unterste anwendungsorientierte Schicht und regelt den Auf- und Abbau von Kommunikationsbeziehungen (Sitzungen, Sessions) sowie deren Wiederherstellung nach Störungen im Transportsystem. Hier findet die Synchronisation und somit der geregelte Dialogablauf zwischen zwei Kommunikationsprozessen statt.

6. Darstellungsschicht (Präsentationsschicht, Presentation Layer)

Die Umsetzung verschiedener Darstellungen der Information (z.B. die Zeichensätze ASCII und EBCDIC) auf ein einheitliches Format auf der Senderseite ist die Aufgabe der Darstellungsschicht. Diese Schicht kann auch Funktionen enthalten, mit denen Daten komprimiert, konvertiert und verschlüsselt werden können. Vor der Web-Ära war das *ASN.1*-Konzept (*Abstract Syntax Notation*) für diese Schicht von großer Bedeutung. Inzwischen wurde die Aufgabe von ASN.1 durch XML (*eXtensible Markup Language*) übernommen.

7. Anwendungsschicht (Applikationsschicht, Application Layer)

In dieser Schicht sind die sog. OSI-Anwendungsprogramme angesiedelt. Einige von ihnen wurden standardisiert. Zu den wichtigsten OSI-Standardanwendungen gehören E-Mail (Standard X.400) und verteilter Verzeichnisdienst (Standard X.500).

Im Allgemeinen kann die Schicht n im OSI-Referenzmodell als Erbringer bestimmter Kommunikationsdienste für die Schicht $n-1$ angesehen werden.

Die mit der Kommunikation verbundenen Aufgaben in den Endsystemen können bestimmten Klassen von Aufgaben zugeordnet werden. Abbildung 1.3-3 bringt dies deutlicher zum Ausdruck.

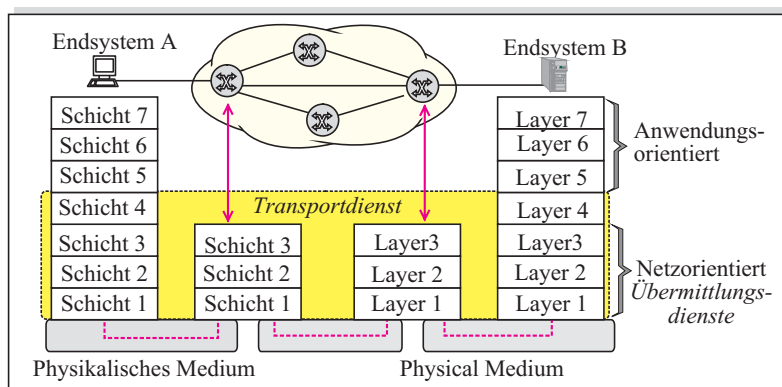


Abb. 1.3-3: Klassen von Aufgaben im OSI-Referenzmodell

Die ersten drei Schichten realisieren die *Übermittlungsdienste*. Schicht 1 realisiert die Übermittlung von Daten bitweise zwischen zwei direkt verbundenen Stationen (d.h. zwischen einem Endsystem und seinem Netzknoten bzw. zwischen zwei benachbarten Netzknoten). Die ersten zwei Schichten realisieren die gesicherte Übermittlung von Daten in Form von Frames zwischen zwei direkt verbundenen Stationen [Abb. 1.3-5]. Die ersten drei Schichten realisieren in der Regel eine ungesicherte Übermittlung von Datenpaketen zwischen zwei Endsystemen, also im Allgemeinen über mehrere Zwischensysteme.

Eine besondere Rolle hat die Schicht 4 (Transportschicht). Sie hat die Aufgabe, die ungesicherte Übermittlung von Datenpaketen zwischen zwei Endsystemen – als den Dienst der ersten drei Schichten – sicher zu machen. Die Aufgabe von

Übermittlungsdienste

Schicht 4 und Schicht 2

Schicht 4 ist somit mit der Aufgabe von Schicht 2 vergleichbar. Diese beiden realisieren die Sicherung der Datenübermittlung. Schicht 2 kümmert sich um die Datenübermittlung über eine „Leitung“ und Schicht 4 kümmert sich um die Übermittlung von Daten zwischen zwei Endsystemen, die in der Regel nicht direkt (physikalisch) verbunden sind.

Transportdienst

Die ersten vier Schichten können daher als *Transportdienst* angesehen werden, der einen gesicherten Datenaustausch zwischen zwei Endsystemen garantiert. Diesen Dienst nutzen die Schichten 5, 6 und 7, die anwendungsorientiert sind.

OSI hat gegen TCP/IP verloren

Das OSI-Referenzmodell stellt ein schönes Konzept für die Strukturierung von Kommunikationsaufgaben in End- und Zwischensystemen dar. In den 80er-Jahren haben die OSI-Verfechter noch geglaubt, dass dieses 7-Schichten-Modell eine ewige Lösung für die offene Kommunikation bleiben wird. Aber bereits in der zweiten Hälfte der 80er-Jahre hat sich herausgestellt, dass die Protokolle für die Realisierung der Schicht 4 (Transportschicht) und der anwendungsorientierten Schichten 5 und 6 sehr komplex sind. Da die Protokollfamilie TCP/IP, die im Vergleich zu den OSI-Protokollen einfacher zu implementieren war, bereits damals breit akzeptiert war, ist der „Zug“ in Richtung TCP/IP abgefahren.

1.3.2 Schichtenmodell der Protokollfamilie TCP/IP

Auch die Protokollfamilie TCP/IP kann in einem Schichtenmodell dargestellt werden. Dieses Modell ist eine vereinfachte Variante des OSI-Referenzmodells, in der die anwendungsorientierten Schichten 5, 6 und 7 aus dem OSI-Referenzmodell [Abb. 1.3-3] zu einer Schicht zusammengefasst sind. Abbildung 1.3-4 zeigt das Schichtenmodell der Protokollfamilie TCP/IP.

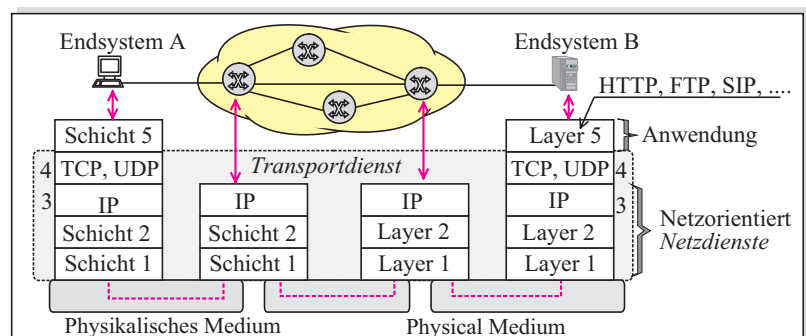


Abb. 1.3-4: Schichtenmodell der Protokollfamilie TCP/IP

Im Allgemeinen entsprechen die Funktionen der Schichten 1, 2, 3 und 4 im Schichtenmodell der Protokollfamilie TCP/IP den Funktionen der gleichen

Schichten im OSI-Referenzmodell. Die Protokolle in den Schichten 1 und 2 in den beiden Schichtenmodellen – d.h. von OSI und von TCP/IP – können auch die gleichen sein. Innerhalb der Schicht 3 im Modell von TCP/IP wird das Protokoll IP (*Internet Protocol*) angesiedelt. Innerhalb der Schicht 4 werden zwei Transportprotokolle TCP (*Transmission Control Protocol*) und UDP (*User Datagram Protocol*) eingesetzt. TCP ist ein verbindungsorientiertes Transportprotokoll. UDP ist dagegen ein verbindungsloses Transportprotokoll. Auf die Unterschiede zwischen TCP und UDP geht Abschnitt 1.4.4 näher ein.

Vergleicht man die Schichtenmodelle von OSI und von TCP/IP, d.h. die Abbildungen 1.3-3 und 1.3-4, stellt man fest, dass die oberen anwendungsorientierten Schichten 5, 6 und 7 aus dem OSI-Referenzmodell beim Schichtenmodell für TCP/IP zu einer Anwendungsschicht zusammengefasst sind. Dies deutet darauf hin, dass die bestimmten Funktionen, die den Schichten 5 und 6 im OSI-Referenzmodell zugeordnet werden, in den TCP/IP-Applikationen entsprechend realisiert werden müssen.

*Schicht 5
als Anwen-
dungsschicht*

Abbildung 1.3-5 zeigt die Strukturen von Daten, die zwischen den kommunizierenden Instanzen innerhalb von einzelnen Schichten übermittelt werden.

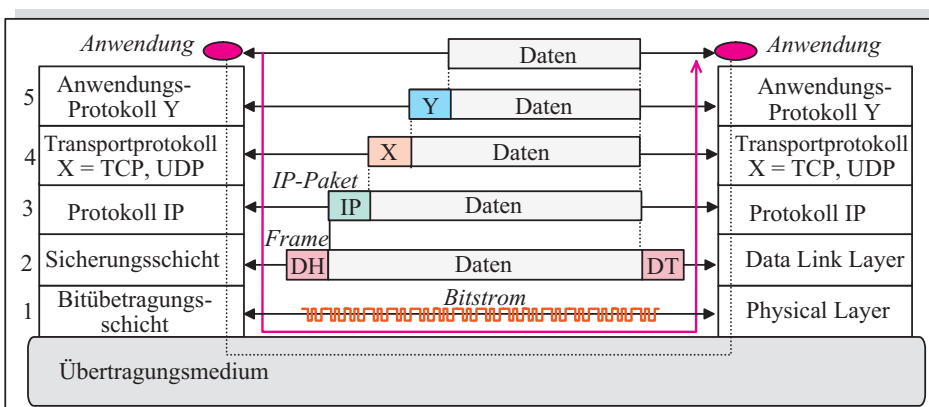


Abb. 1.3-5: Strukturen von zwischen den kommunizierenden Instanzen übermittelten Daten
DH: Data Link Header, DT: Data Link Trailer

Vereinfacht kann man sich die Übermittlung von Daten zwischen zwei Anwendungen folgendermaßen vorstellen: Dem zu sendenden Datenblock *Daten* wird ein Header *Y* mit bestimmten Angaben des Anwendungsprotokolls *Y* (z.B. *Y* = HTTP) im Quellrechner vorangestellt. Das Paar (*Y*, *Daten*) muss nun an die Instanz des gleichen Anwendungsprotokolls *Y* im Zielrechner übermittelt werden und wird hierfür an das Transportprotokoll *x* (*x* = TCP bzw. UDP) übergeben. Hier wird ein Header *x* des Transportprotokolls dem Paar [*Y*, *Daten*] vorangestellt, sodass eine Dateneinheit [*x*[*Y*, *Daten*]] des Transportproto-

*Strukturierung der
übertragenen Daten*

kolls entsteht. Diese Dateneinheit wird nun an die IP-Instanz übergeben, wo ihr ein IP-Header vorangestellt wird. So entsteht ein *IP-Paket*, das zum Aussenden in einen *Data Link Frame (DL-Frame)* eingebettet wird. Dieser DL-Frame wird nun zum Zielrechner übertragen. Dort müssen die empfangenen Daten aus Schicht 1 an die Anwendung (Schicht 5) übergeben werden.

*Übermitt-
lungsvor-
gang*

Dieser in Abbildung 1.3-5 gezeigte Übermittlungsvorgang lässt sich wie folgt zusammenfassen:

1. **Quellrechner: Vorbereitung von Daten zum Senden**

Daten \Rightarrow Anwendungsprotokolleinheit [Y, Daten] \Rightarrow Transportprotokolleinheit [X[Y, Daten]] \Rightarrow IP-Paket [IP[X[Y, Daten]]] \Rightarrow DL-Frame [DH[IP[X[Y, Daten]]]DT]

2. **DL-Frame wird bitweise übertragen**

3. **Zielrechner: Übergabe von Daten an die Anwendung**

DL-Frame [DH[IP[X[Y, Daten]]]DT] \Rightarrow IP-Paket[IP[X[Y, Daten]]] \Rightarrow Transportprotokolleinheit [X[Y, Daten]] \Rightarrow Anwendungsprotokolleinheit [Y, Daten] \Rightarrow Daten

Bemerkung: Abbildung 1.3-5 zeigt eine vereinfachte Situation. Die zu sendenden Datenmengen können so groß sein, dass man sie in einem IP-Paket nicht übermitteln kann. In diesem Fall kommt TCP zum Einsatz und die Daten werden auf mehrere IP-Pakete aufgeteilt. Man spricht hierbei von *Segmentierung von Daten*. Ein IP-Paket enthält damit ein *Datensegment* [Abb. 1.4-2].

1.4 Allgemeine Prinzipien der IP- Kommunikation

Die wichtigen Prinzipien der Kommunikation in IP-Netzen können weitgehend aus dem in Abschnitt 1.3.2 dargestellten Schichtenmodell abgeleitet werden. Hierbei spielen die Schichten *Netzwerkschicht* mit dem Protokoll IP und *Transportschicht* mit den Protokollen TCP und UDP eine dominierende Rolle. Bevor auf diese beiden Schichten eingegangen wird, soll zuerst die Bildung von IP-Paketen kurz dargestellt werden.

1.4.1 Bildung von IP-Paketen

*Nutzung von
UDP*

Bei der Bildung von IP-Paketen ist zu unterscheiden, ob TCP oder UDP als Transportprotokoll eingesetzt wird. Beim Einsatz des verbindungslosen Transportprotokolls UDP werden die Daten bzw. eine Nachricht einer Anwendung – als *Nutzlast* – um den UDP-Header ergänzt, sodass eine *UDP-Dateneinheit* entsteht. Wie Abbildung 1.4-1 zeigt, wird aus jeder UDP-Dateneinheit durch das Voranstellen eines IP-Header ein *IP-Paket* gebildet. Da die IP-Pakete keine

Angaben zur Synchronisation enthalten, um sie auf der Leitung zu „markieren“, müssen sie in *Data Link Frames (DL-Frames)* eingebettet werden.

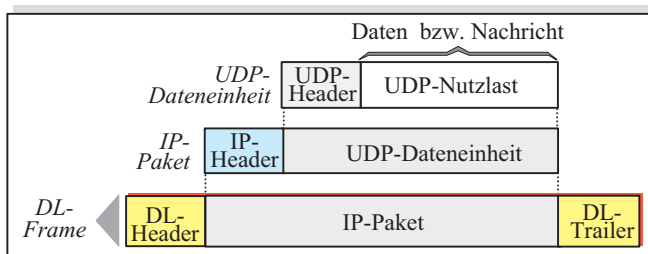


Abb. 1.4-1: Kapselung der Nutzlast beim UDP-Einsatz

In LANs bildet die sog. MAC-Funktion (*Media Access Control*) den Kern der Data-Link-Schicht. Wird ein IP-Paket in einem LAN übermittelt, wird es in einen MAC-Frame eingebettet. Bei der Übermittlung der IP-Pakete über eine Leitung bzw. über eine Punkt-zu-Punkt-Verbindung wird innerhalb der Schicht 2 häufig das Protokoll PPP (*Point-to-Point Protocol*) verwendet. In diesem Fall stellen die DL-Frames *PPP-Frames* dar.

MAC-Frames in LANs

Jedes zu übertragende IP-Paket muss immer in einen DL-Frame eingebettet werden. Dies bedeutet, dass jedem IP-Paket ein DL-Header vorangestellt wird und nach dem Ende des IP-Pakets folgt ein DL-Trailer. Diese beiden enthalten bestimmte *Synchronisationsangaben* (oft sog. Flags 01111110), um den Beginn und das Ende des DL-Frames auf einer Leitung zu erkennen.

Bedeutung von DL-Frames

Abbildung 1.4-2 illustriert, wie die IP-Pakete aus den Daten bzw. aus der langen Nachricht eines Anwendungsprotokolls bei der Nutzung des verbindungsorientierten Transportprotokolls TCP gebildet werden.

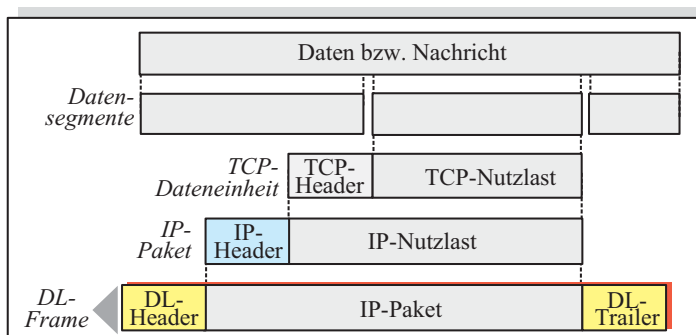


Abb. 1.4-2: Verkapselung der Nutzlast beim TCP-Einsatz

Anders als bei UDP entstehen aus den zu übermittelnden Daten bei TCP mehrere *Datensegmente*. Jedes Datensegment wird dann um einen TCP-Header erweitert, sodass eine *TCP-Dateneinheit* entsteht. Aus jeder TCP-Dateneinheit wird im nächsten Schritt ein IP-Paket gebildet. Zum Senden wird das IP-Paket in einen DL-Frame eingekapselt.

Wie aus den Abbildungen 1.4-1 und 1.4-2 ersichtlich ist, werden die IP-Pakete zum Senden immer in entsprechende DL-Frames der zweiten Schicht eingekapselt, die vom Übermittlungsnetz abhängig sind. Erst in einem DL-Frame kann ein IP-Paket über ein physikalisches Netz gesendet werden.

1.4.2 Netzwerkschicht in IP-Netzen

Arten der
Netzwerk-
schicht

verbindungs-
los

verbindungs-
orientiert

Die Netzwerkschicht in IP-Netzen hat die Aufgabe, die Daten in Form von IP-Paketen zwischen Endsystemen zu übermitteln. Hierbei unterscheidet man zwischen der verbindungslosen und der verbindungsorientierten Netzwerkschicht:

- Wird keine Route über das Netz für einen Strom der von einem Quellrechner zu einem Zielrechner zu übermittelnden IP-Pakete festgelegt, sondern jedes einzelne Paket aus diesem Strom nach einem eigenen Weg über das Netz zum Zielrechner übermittelt, handelt es sich um die *verbindungslose Netzwerkschicht*.
- Wird eine Route über das Netz für einen Strom der von einem Quellrechner zu einem Zielrechner zu übermittelnden IP-Pakete festgelegt und werden alle Pakete aus diesem Strom nach dem gleichen Weg über das Netz, der eine *logische Verbindung* darstellt, zum Zielrechner übermittelt, handelt es sich um die *verbindungsorientierte Netzwerkschicht*.

Verbindungslose Netzwerkschicht

Die verbindungslose Netzwerkschicht bedeutet, dass die Vermittlungsnetzknöten im IP-Netz die Router darstellen und die einzelnen IP-Pakete als sog. *Datagramms* voneinander unabhängig über das Netz übermittelt werden. Diese Übermittlungsart entspricht dem Versand von Briefen bei der Post. Jedes IP-Paket kann daher mit einem Brief verglichen werden. Der Router würde einer Briefverteilungsstelle entsprechen. Abbildung 1.4-3 illustriert die Struktur der verbindungslosen Netzwerkschicht in IP-Netzen.

Interpretation der
IP-Adresse

Die ersten drei unten liegenden Schichten realisieren also beim Einsatz von Routern einen *verbindungslosen Übermittlungsdienst*. Dieser entspricht dem Briefpostdienst und eine IP-Adresse ist mit einer postalischen Adresse vergleichbar. Die IP-Adresse stellt auch einen Zugangspunkt zum Dienst für die Übermittlung der IP-Pakete dar und ist oberhalb der Schicht 3 – also an der Grenze zu Schicht 4 – anzusiedeln.

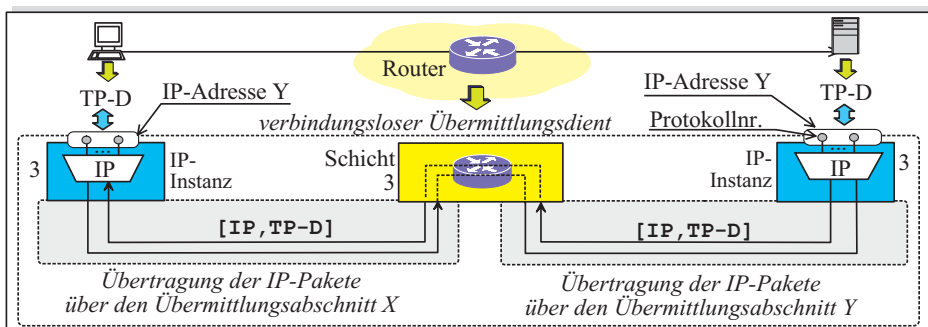


Abb. 1.4-3: Struktur der verbindungslosen Netzwerkschicht in IP-Netzen
TP-D: Transportprotokoll dateneinheit

Die IP-Instanz kann als ein *IP-Multiplexer* angesehen werden. Zwischen den IP-Instanzen werden die IP-Pakete übermittelt. Jedes IP-Paket setzt sich aus einem IP-Header *IP* und einer Transportprotokoll dateneinheit *TP-D* zusammen, d.h., es hat die Struktur $[IP, TP-D]$.

Die Ports im IP-Multiplexer repräsentieren die Nummern der Protokolle von Schicht 4, die auf die Übermittlungsdienste direkt zugreifen können [Abb.1.4-7 und -8]. Die Protokollnummer wird im IP-Header übermittelt [Abb. 2.2-1] und informiert, von welchem Protokoll die Dateneinheit im IP-Paket stammt. Jedem Protokoll der Schicht 4 wird daher von der IANA (*Internet Assigned Numbers Authority*) eine feste und weltweit eindeutige Nummer zugewiesen.

Verbindungsorientierte Netzwerkschicht

Der Einsatz von MPLS (*Multi-Protocol Label Switching*) bzw. von GMPLS (*Generalized MPLS*) führt zur verbindungsorientierten Netzwerkschicht in IP-Netzen [Kapitel 11]. In diesem Fall fungieren die sog. *(G)MPLS-Switches* als Vermittlungsnetzknotten. Bei der verbindungsorientierten Netzwerkschicht wird zuerst eine Route über das Netz für die Übermittlung eines Stroms der IP-Pakete festgelegt und danach werden alle IP-Pakete aus diesem Strom als „Gänsemarsch“ über das Netz vom Quellrechner zum Zielrechner übermittelt. Diese Übermittlungsart wird heute hauptsächlich in IP-Netzen von großen Netzdienst Anbietern realisiert. Die Technik *(G)MPLS* wird im *Next Generation Internet*, auch als *Internet2* bezeichnet, eingesetzt.

Abbildung 1.4-4 zeigt die Struktur der verbindungsorientierten Netzwerkschicht in IP-Netzen beim MPLS-Einsatz.

Die ersten drei unten liegenden Schichten realisieren beim MPLS-Einsatz einen verbindungsorientierten Übermittlungsdienst. Die IP-Adresse stellt einen Zugangspunkt zu diesem Dienst dar. Die IP-Instanz enthält hier – im Vergleich zur IP-Instanz in Abbildung 1.4-3 – zusätzlich einen *MPLS-Multiplexer*.

MPLS-Multiplexer

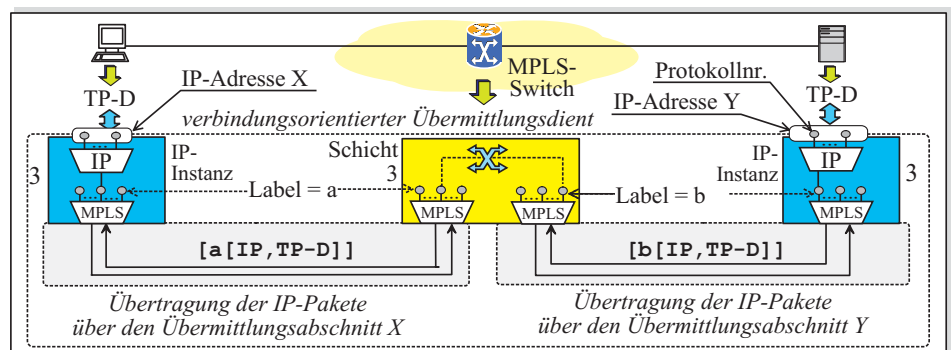


Abb. 1.4-4: Struktur der verbindungsorientierten Netzwerkschicht in IP-Netzen

Bedeutung von Label

Einem Strom von IP-Paketen wird ein Port im MPLS-Multiplexer zugeordnet. Somit können mehrere Datenströme parallel übermittelt werden. Die Ports im MPLS-Multiplexer stellen die sog. *Labels* dar. Ein Label wird immer den zu übermittelnden IP-Paketen eines Stroms vorangestellt. Abbildung 1.4-4 bringt dies zum Ausdruck. Ein Label informiert, von welchem Port im MPLS-Multiplexer ein IP-Paket stammt bzw. welchem Port es übergeben werden muss. Ein MPLS-Switch leitet – im Allgemeinen – ein empfangenes IP-Paket nach einer Switching-Tabelle von einem Port zu einem anderen zum Aussenden weiter. Daher kann ein anderes Label den IP-Paketen eines Stroms auf einem anderen Übermittlungsabschnitt vorangestellt werden.

Zwischen den Ports im MPLS-Multiplexer entsteht entsprechend im Quell- und im Zielrechner eine logische Verknüpfung, die als *virtuelle (logische) Verbindung* interpretiert wird.

1.4.3 Verbindungslose IP-Kommunikation im Internet

Nachbildung des Briefdienstes

Das Internet stellt eine weltweite Kopplung von physikalischen Netzen dar, in denen das Protokoll IP eingesetzt wird. Somit kann das Internet als *heterogenes IP-Netz* angesehen werden. Als IP-Netz setzt sich das Internet aus einer Vielzahl von IP-Subnetzen zusammen, die mithilfe von Routern miteinander vernetzt sind. Daher ist die Netzwerkschicht im heutigen Internet verbindungslos [Abb. 1.4-3]. Ein Router leitet jedes empfangene IP-Paket unabhängig von der aktuellen Lage im Netz und von anderen Paketen weiter. In Analogie zum Briefdienst der Post könnte man ein IP-Subnetz mit einem Postleitzahlengebiet vergleichen und einen Router mit einer Briefverteilungsstelle.

Abbildung 1.4-5 illustriert das Prinzip der Kommunikation im Internet an einem Beispiel, in dem eine Folge von TCP-Dateneinheiten gesendet wird. Jede dieser Dateneinheiten wird als ein IP-Paket gesendet. Im Zielrechner setzt TCP die in den IP-Paketen empfangenen Daten wieder zusammen. Gehen einige

TCP-Dateneinheiten bei der Übertragung verloren bzw. werden sie verfälscht, so fordert TCP im Zielrechner vom Quellrechner eine wiederholte Übertragung [Abschnitt 3.3].

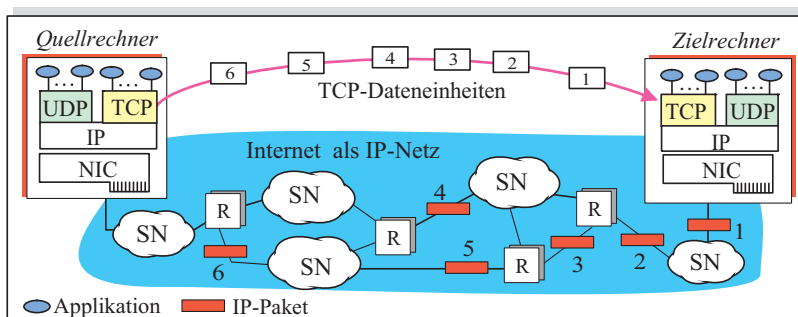


Abb. 1.4-5: Prinzip der Kommunikation im Internet
R: Router, SN: IP-Subnetz, NIC: Network Interface Card (Controller)

Beim Einsatz von Routern werden die IP-Pakete als sog. *Datagrams* (also wie Briefe) unabhängig voneinander zum Zielrechner gesendet. Die wichtigsten Angaben in IP-Paketen sind die IP-Adressen von Quell- und Zielrechner. Da die einzelnen IP-Pakete unabhängig voneinander abgeschickt werden, können sie am Ziel in einer anderen Reihenfolge ankommen, als sie abgeschickt wurden. Für die Wiederherstellung von Daten aus so empfangenen IP-Paketen ist TCP verantwortlich.

*IP-Pakete
wie Briefe*

Da die IP-Pakete im Netz zirkulieren können, ist es nötig, ihre Verweilzeit im Netz zu kontrollieren. Der Quellrechner gibt als TTL-Angabe (*Time To Live*) im IP-Header [Abb. 2.2-1] an, wie lange das IP-Paket im Netz verweilen darf. Weil der TTL-Wert in jedem Router um 1 verringert wird, ist er identisch mit der maximalen Anzahl von Routern, die ein IP-Paket durchlaufen darf. Fällt der TTL-Wert auf 0, wird das IP-Paket im Router verworfen. Der Quellrechner wird dann mit einer Meldung des Protokolls ICMP (*Internet Control Message Protocol*) darüber informiert.

*Bedeutung
von TTL*

1.4.4 Transportschicht in IP-Netzen

Um die Bedeutung der Transportschicht in IP-Netzen näher zu erläutern, zeigt Abbildung 1.4-6 die vereinfachte Struktur von Rechnern am IP-Netz. Die IP-Adresse eines Rechners kann einem Kommunikationspuffer zugeordnet werden, der einen Zugangspfort zum Protokoll IP darstellt. Dieser Kommunikationspuffer befindet sich an der Grenze zwischen der Schicht 3 mit dem Protokoll IP und der Schicht 4 mit den Transportprotokollen TCP und UDP.

*Interpretation
der
IP-Adresse*

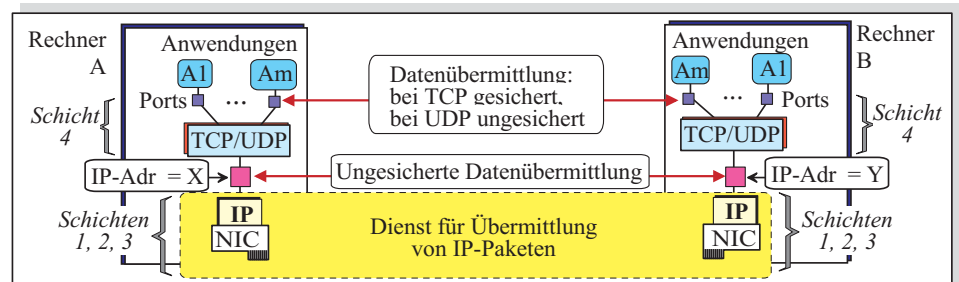


Abb. 1.4-6: Vereinfachte Struktur von Rechnern am IP-Netz
A: Applikation, Adr: Adresse, NIC: Network Interface Controller

Die drei Schichten 1, 2 und 3 stellen einen Dienst für die Übermittlung der IP-Pakete zwischen den Rechnern zur Verfügung. Es handelt sich hier um eine ungesicherte Übermittlung von IP-Paketen zwischen IP-Adressen. Eine IP-Adresse stellt einen Zugangspunkt zu diesem Übermittlungsdienst für die Protokolle TCP und UDP der Transportschicht (Schicht 4) dar.

Arten der Kommunikation

Die Transportschicht regelt den Verlauf der Datenübermittlung zwischen Anwendungen – genauer gesagt zwischen Ports dieser Anwendungen – in verschiedenen Rechnern. Hierbei sind zwei Arten der Kommunikation zu unterscheiden:

- *verbindungslose Kommunikation* beim UDP-Einsatz,
- *verbindungsorientierte Kommunikation* beim TCP-Einsatz

UDP-Multiplexer

Abbildung 1.4-7 zeigt die Transportschicht mit UDP. Eine UDP-Instanz kann als *UDP-Multiplexer* angesehen werden. Die Eingangsporte zu diesem Multiplexer stellen die Kommunikationspuffer einzelner UDP-Anwendungen dar, die kurz als *Ports* bezeichnet werden. Der Ausgangsport des UDP-Multiplexers führt zu einer IP-Adresse. Damit können mehrere UDP-Anwendungen parallel auf den Dienst für die Übermittlung der IP-Pakete zugreifen.

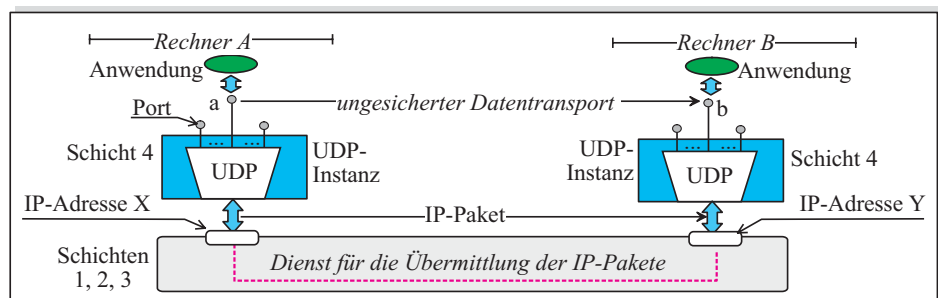


Abb. 1.4-7: Transportschicht mit UDP; ungesicherter Datentransport

Beim UDP-Einsatz ist die Kommunikation zwischen zwei Anwendungen verbindungslos, d.h. es wird keine Vereinbarung über den Verlauf der Kommunikation zwischen ihnen getroffen. Der Quellrechner als Initiator der Kommunikation übermittelt ein UDP-Paket an den Zielrechner, ohne ihn zu „fragen“, ob er in der Lage ist, dieses Paket zu empfangen. Bei derartiger Kommunikation findet daher keine Fehler- und Flusskontrolle statt [Abschnitt 1.2].

verbindungslose Kommunikation

Bei der *verbindungsorientierten Kommunikation* zwischen zwei Anwendungen beim TCP-Einsatz vereinbaren die beiden kommunizierenden Rechner zuerst, wie die Kommunikation zwischen ihnen verlaufen soll, d.h., wie die zu übertragenden Daten zu nummerieren sind und wie die Fehler- und die Flusskontrolle ablaufen sollen. Eine Vereinbarung zwischen zwei Rechnern in Bezug auf den Verlauf der Kommunikation zwischen ihnen wird als *TCP-Verbindung* bezeichnet. Abbildung 1.4-8 illustriert dies näher.

verbindungsorientierte Kommunikation

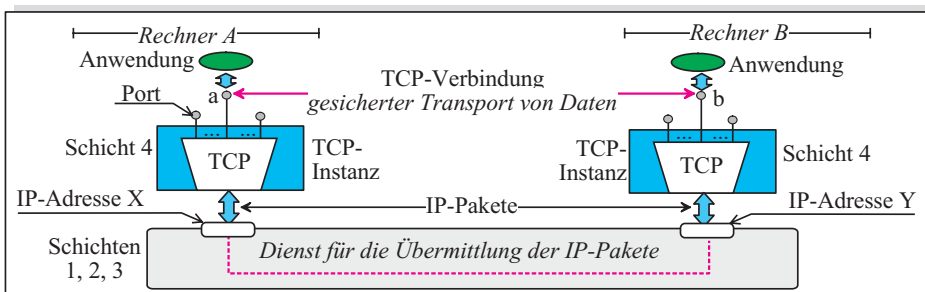


Abb. 1.4-8: Transportschicht mit TCP; gesicherter Datentransport

Eine TCP-Instanz ist auch ein *TCP-Multiplexer*. Die Eingangsports zu diesem Multiplexer stellen die *Ports* einzelner TCP-Anwendungen dar. Der Ausgangsport des TCP-Multiplexers führt, wie bei UDP, zu einer IP-Adresse, sodass mehrere TCP-Anwendungen parallel auf den Dienst für die Übermittlung der IP-Pakete zugreifen können.

TCP-Multiplexer

Die TCP- und UDP-Anwendungen, wie z.B. HTTP, FTP bzw. SIP, sind feste Standardanwendungen, die unter den allgemein bekannten und weltweit eindeutigen Portnummern (in Zielrechnern!) erreichbar sind. Eine derartige Nummer wird in der TCP/IP-Welt als *Well Known Port* bezeichnet. Eine Zusammenstellung von Standardanwendungen und deren Portnummern kann in UNIX-Rechnern in der Datei `/etc/services` eingesehen werden. Unter der Adresse <http://www.iana.org/assignments/port-numbers> befindet sich die Auflistung aller Well Known Ports.

Well Known Ports

Lokation von Anwendungen

Um eine TCP- und eine UDP-Anwendung eindeutig weltweit zu lokalisieren, muss man Folgendes angeben:

- auf welchem Rechner die Anwendung läuft; das bestimmt eindeutig die IP-Adresse des Rechners.
- auf welchen Port im UDP- bzw. TCP-Multiplexer die Anwendung zugreift; das bestimmt die UDP- bzw. TCP-Portnummer.

Bedeutung von Socket

Eine TCP- und UDP-Anwendung lokalisiert man daher durch die Angabe (IP-Adresse, Port). Dieses Paar hat eine fundamentale Bedeutung bei der Rechnerkommunikation und wird als *Socket* bezeichnet. Die Rechnerkommunikation bei TCP/IP kann mithilfe von Sockets sehr anschaulich dargestellt werden. Abbildung 1.4-9 illustriert dies.

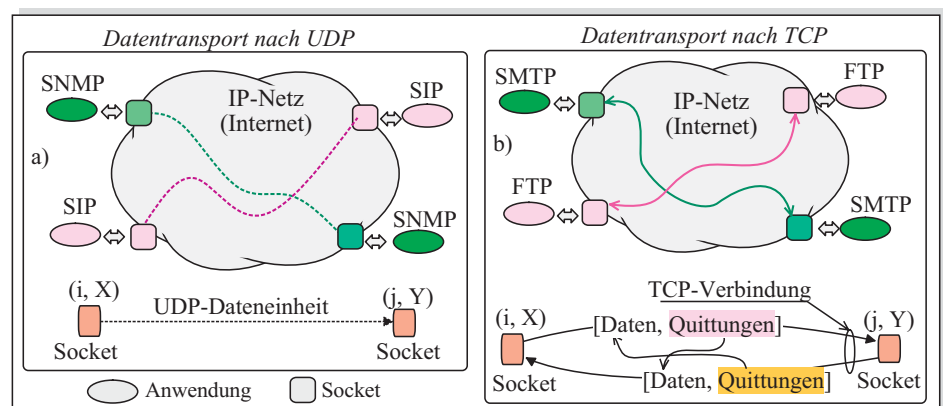


Abb. 1.4-9: Datentransport zwischen Anwendungen:
a) beim UDP-Einsatz, b) beim TCP-Einsatz,

Socket als Software-Steckdose

Sockets dienen somit als *Zugangspunkte* zu einer Wolke, die ein IP-Netz bzw. das ganze Internet repräsentiert. Ein Socket kann auch als „Software-Steckdose“ für den Anschluss einer Anwendung an das IP-Netz angesehen werden. Jedem Socket steht im Rechner ein reservierter Speicherplatz als Kommunikationspuffer zur Verfügung. Die zu übertragenden und zu empfangenden Daten einer Anwendung werden jeweils in dem für das Socket reservierten Kommunikationspuffer abgelegt. Sockets sind somit auf die *Zeitdauer* der Verbindung beschränkt.

Wie Abbildung 1.4-9a zeigt, wird bei UDP keine Verknüpfung von Sockets hergestellt, sondern eine UDP-Dateneinheit direkt an den Zielrechner gesendet und ihr Empfang vom Zielrechner nicht bestätigt.

TCP-Verbindung

Bei TCP hingegen (Abbildung 1.4-9b) vereinbaren die zwei Rechner, wie der Verlauf des Datentransports zwischen zwei Sockets geregelt werden soll. Damit wird zwischen zwei Sockets eine *logische Verknüpfung* hergestellt, die eine

TCP-Verbindung darstellt. Ein Socket bei TCP ist auch ein Endpunkt einer TCP-Verbindung. Eine TCP-Verbindung ist vollduplex und setzt sich aus zwei entgegengerichteten unidirektionalen Verbindungen zusammen. Eine TCP-Verbindung kann auch als „zweispurige virtuelle Straße“ über ein IP-Netz bezeichnet werden. Über diese Straße erfolgt ein gesicherter Datentransport, d.h. die empfangenen Daten werden entsprechend quittiert [Abschnitt 3.3].

1.4.5 Multiplexmodell der Protokollfamilie TCP/IP

Nach der Beschreibung der einzelnen Schichten im Schichtenmodell für TCP/IP soll jetzt die Adressierung in IP-Netzen näher dargestellt werden. Abbildung 1.4-10 zeigt ein Multiplexmodell der Protokollfamilie TCP/IP, falls ein IP-Netz auf LAN-Basis, z.B. auf Ethernet-Basis, aufgebaut wird.

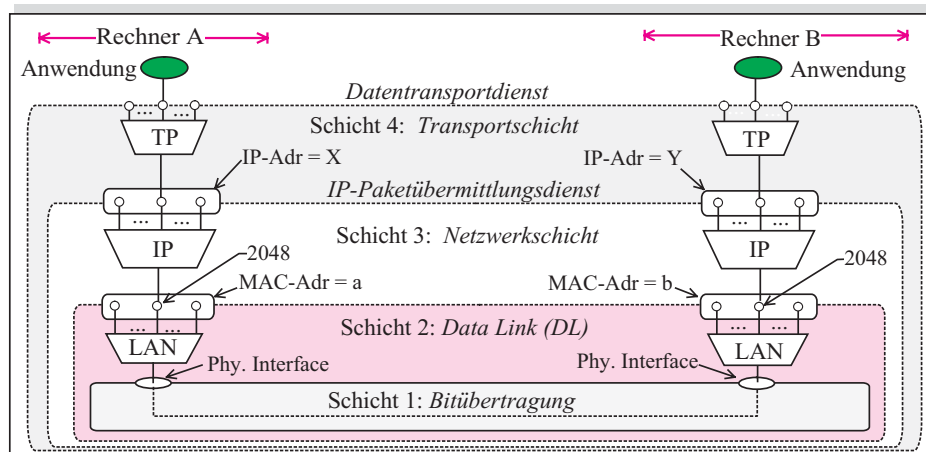


Abb. 1.4-10: Multiplexmodell der Protokollfamilie TCP/IP beim IP-Netz auf LAN-Basis
TP: UDP bzw. TCP

Hier soll u.a. gezeigt werden, dass alle Schichten von 1 bis $n-1$ einen Übermittlungsdienst der Schicht n zur Verfügung stellen.

Die Schicht 1 stellt einen Dienst für die Übermittlung der Bitströme zur Verfügung. Der Zugang zu diesem Dienst erfolgt über physikalische Interfaces.

Ein Rechner am LAN enthält normalerweise eine LAN-Adapterkarte, die zusammen mit einem Treiber u.a. die Funktion eines Multiplexers realisiert [Abb. 10.1-1]. Die Ports in diesem *LAN-Multiplexer* repräsentieren die Nummern der Protokolle von Schicht 3 [Abschnitt 10.1.2]; die Nummer von IP ist beispielsweise 2048. Jeder Rechner am LAN ist unter einer sog. *MAC-Adresse* erreichbar. Sie ist an der Grenze zwischen Schicht 2 und 3 anzusiedeln und kann auch als Zugangspunkt zum Dienst der Schicht 2 interpretiert werden. Über eine

*Interpretation
der MAC-
Adresse*

MAC-Adresse können daher verschiedene Protokolle der Schicht 3 auf diesen Dienst – also auf den LAN-Dienst – zugreifen.

IP-Multiplexer

Logisch gesehen wird die IP-Protokollinstanz aus der Schicht 3, die als *IP-Multiplexer* interpretiert werden kann [Abb. 1.4-3 und -4], an den Port 2048 im LAN-Multiplexer angebunden. Ein Port im IP-Multiplexer repräsentiert die Nummer eines Protokolls der Transportschicht. Schicht 3 stellt einen Dienst für die Übermittlung der IP-Pakete zwischen entfernten Rechnern. Eine IP-Adresse kann als Zugangspunkt zu diesem Dienst betrachtet werden und über sie können mehrere Protokolle der Transportschicht diesen Dienst nutzen.

Die Instanzen der Transportprotokolle TCP bzw. UDP realisieren ebenfalls die Multiplexfunktion [Abb. 1.4-7 und -8]. Daher können mehrere TCP- bzw. UDP-Anwendungen über eine IP-Adresse auf die Dienste für die Übermittlung der IP-Pakete zugreifen.

1.5 Komponenten der Protokollfamilie TCP/IP

Nach der Darstellung der Kommunikationsprinzipien bei TCP/IP anhand des Schichtenmodells soll nun gezeigt werden, welche Protokolle den einzelnen Schichten zuzuordnen sind und wie sie miteinander kooperieren. Abbildung 1.5-1 zeigt die Protokollfamilie TCP/IP beim klassischen Protokoll IP, d.h. IP in Version 4 (IPv4). Eine ähnliche Darstellung bei IP in Version 6 (IPv6) findet sich in Abschnitt 8.1.1.

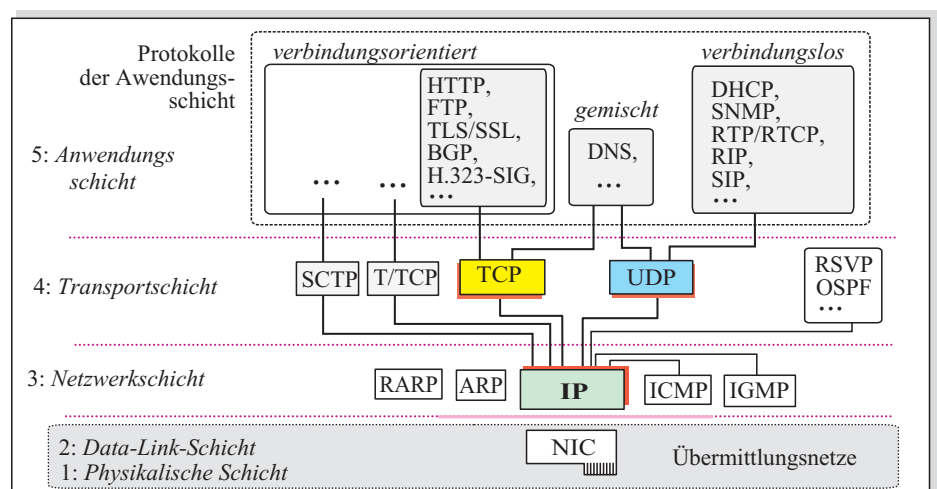


Abb. 1.5-1: Protokolle der Familie TCP/IPv4 im Schichtenmodell
NIC: Network Interface Card (Adapterkarte)

Wie hier gezeigt wird, besteht die Protokollfamilie TCP/IP nicht nur aus den Protokollen TCP und IP, sondern enthält eine Reihe weiterer Protokolle, die den Schichten *Netzwerkschicht*, *Transportschicht* und *Anwendungsschicht* im Schichtenmodell für TCP/IP [Abb. 1.3-4] zugeordnet werden können.

1.5.1 Protokolle der Netzwerkschicht

Die Netzwerkschicht im Schichtenmodell für TCP/IP beschreibt u.a., wie die IP-Netze logisch auf sog. *IP-Subnetze* aufgeteilt werden können und wie die Daten in Form von *IP-Paketen* in einzelnen IP-Subnetzen und zwischen ihnen übermittelt werden. Die Protokolle der Netzwerkschicht sind:

- **IP: *Internet Protocol***
IP liegt sowohl in der alten Version 4 (IPv4) als auch in der neuen Version 6 (IPv6) vor. IPv4 und IPv6 sind unterschiedliche Implementierungen auf der Netzwerkschicht und nutzen getrennte Adressräume bzw. Adressierungsverfahren. IPv4 wird ausführlich in Kapitel 2 dargestellt. Auf IPv6 wird in Kapitel 6 detailliert eingegangen.
- **ARP: *Address Resolution Protocol***
Dieses Protokoll ist ein Broadcast-Dienst zur dynamischen Ermittlung einer MAC-Adresse eines Rechners im LAN, falls seine IP-Adresse bekannt ist [Abschnitt 2.6.1].
- **RARP: *Reverse Address Resolution Protocol***
RARP unterstützt ebenfalls die Adressierung und stellt das Gegenstück zu ARP dar. Es hat die Aufgabe, für eine MAC-Adresse eine IP-Adresse zu bestimmen [Abschnitt 2.6.3].
- **ICMP: *Internet Control Message Protocol***
ICMP wird für die Übermittlung von Fehlermeldungen und anderen Kontrollangaben verwendet [Abschnitt 2.7].
- **IGMP: *Internet Group Management Protocol***
IGMP gilt als Erweiterung von ICMP und dient vornehmlich dazu, das Management von sog. Multicast-Gruppen in IP-Subnetzen zu unterstützen [Abschnitt 2.8.2].

Bemerkung: Die Protokolle ICMP und IGMP werden üblicherweise der Schicht 3 im TCP/IP-Schichtenmodell zugeordnet. Da die Nachrichten dieser Protokolle in IP-Paketen übermittelt werden, könnte man ICMP und IGMP nach den im OSI-Referenzmodell geltenden Prinzipien zwar der Schicht 4 zuordnen, aber ICMP und IGMP sind keine Transportprotokolle.

1.5.2 Protokolle der Transportschicht

In der Transportschicht befinden sich die Protokolle für die Unterstützung der verbindungsorientierten und der verbindungslosen Kommunikation sowie andere spezielle Protokolle. Die wichtigsten sind:

- **TCP: *Transmission Control Protocol***
Mit TCP wird die verbindungsorientierte Kommunikation zwischen Rechnern unterstützt. Hierbei wird zwischen ihnen eine virtuelle Verbindung aufgebaut, die als *TCP-Verbindung*

bezeichnet wird. Eine TCP-Verbindung kann als „Straße“ mit zwei entgegengerichteten Spuren angesehen werden [Abb. 1.4-9b]. Somit ist TCP ein *verbindungsorientiertes Transportprotokoll*. Durch die Realisierung der Fehler- und der Flusskontrolle garantiert TCP einen zuverlässigen Datentransport. Da bei TCP die zu übertragenden Bytes nummeriert werden, ist TCP ein *bytestream-orientiertes Protokoll*. TCP wird in Abschnitt 3.3 beschrieben.

■ **UDP: User Datagram Protocol**

Mit UDP wird die verbindungslose Kommunikation zwischen Rechnern unterstützt, bei der keine virtuelle Verbindung aufgebaut wird. Somit ist UDP ein *verbindungsloses Transportprotokoll*. Bei UDP erfolgt keine Fehler- bzw. Flusskontrolle, sodass UDP im Gegensatz zu TCP keinen zuverlässigen Datentransport garantiert. Auf UDP geht Abschnitt 3.2 ein.

■ **T/TCP: Transaction TCP**

T/TCP ist eine Ergänzung von TCP im Hinblick auf die Unterstützung von sog. *Transaktionen*. Unter einer Transaktion versteht man einen Kommunikationsvorgang, der aus mehreren Phasen besteht, die alle korrekt durchgeführt werden müssen (s. Abschnitt 3.4.4).

■ **SCTP: Stream Control Transmission Protocol**

Ebenso wie TCP ermöglicht SCTP die verbindungsorientierte Kommunikation zwischen Rechnern. Bei SCTP wird eine virtuelle Verbindung, die *SCTP-Assoziation*, aufgebaut. Eine SCTP-Assoziation kann als „Autobahn“ mit einer beliebigen Anzahl von entgegengerichteten Spuren angesehen werden. Daher ist SCTP ein *verbindungsorientiertes Transportprotokoll*. SCTP garantiert einen zuverlässigen Datentransport durch die Realisierung der Fehler- und der Flusskontrolle. Da bei SCTP im Gegensatz zu TCP nicht die zu übertragenden Bytes nummeriert werden, sondern die zu übertragenden Nachrichten (Datenblöcke), ist SCTP ein *nachrichtenorientiertes Protokoll*. Auf SCTP geht Abschnitt 3.5 näher ein.

■ **RSVP: ReSource ReserVation Protocol**

RSVP ist kein Transportprotokoll, sondern ein Protokoll für die Reservierung von bestimmten Netzressourcen, wie z.B. der Bandbreite in Leitungen, um die Anforderungen der Echtzeitkommunikation zu erfüllen. Diese Anforderungen sind unter dem Begriff *Quality of Service (QoS)* bekannt. RSVP wird erweitert und als Signalisierungsprotokoll in (G)MPLS-Netzen verwendet. Dies wird in Abschnitt 11.5 näher dargestellt.

■ **OSPF: Open Shortest Path First**

OSPF ist ein Routing-Protokoll, das vor allem bei Internet-Routern Verwendung findet. OSPF wird in Kapitel 9 ausführlich dargestellt.

Bemerkung: Die sog. *Routing-Protokolle* [Kapitel 9] werden in der Literatur der Netzwerkschicht (Schicht 3) zugeordnet, also der Schicht, in der IP angesiedelt ist. Da die OSPF-Nachrichten direkt in IP-Paketen übermittelt werden, lässt sich OSPF nach den im TCP/IP-Schichtenmodell geltenden Prinzipien nicht der Netzwerkschicht zuordnen, sondern der Transportschicht.

Bei der Übermittlung von Nachrichten des Routing-Protokolls RIP (*Routing Information Protocol*) wird UDP verwendet; somit ist RIP der Anwendungsschicht zuzuordnen. Das Routing-Protokoll BGP (*Border Gateway Protocol*) nutzt dagegen TCP und ist daher ebenfalls der Anwendungsschicht zuzuordnen

1.5.3 Komponenten der Anwendungsschicht

In der Anwendungsschicht werden verschiedene Funktionskomponenten angesiedelt. Diese lassen sich in die folgenden vier Gruppen aufteilen:

- Anwendungsprotokolle

Unter einem *Anwendungsprotokoll* wird im Weiteren ein Protokoll verstanden (z.B. FTP), mit dem sich eine bestimmte Anwendung realisieren lässt.

- Netzdienstprotokolle

Als *Netzdienstprotokoll* wird hier ein Protokoll (z.B. DHCP) verstanden, mit dem ein bestimmter Netzdienst erbracht wird. Beispielsweise können mit DHCP-Hilfe die IP-Adressen dem Rechner nach Bedarf dynamisch zugeteilt werden. Dies stellt einen Netzdienst dar. Auch Routing-Protokolle können als Netzdienstprotokolle betrachtet werden.

- Benutzerdienstprotokolle

Als *Benutzerdienstprotokoll* wird ein Protokoll (z.B. SIP) verstanden, mit dem ein bestimmter Dienst für den Benutzer erbracht werden kann. Beispielsweise kann mit SIP der VoIP-Dienst (*Voice over IP*) realisiert werden.

- Remote-Kommandos

Hierzu gehören einige Kommandos unter UNIX und LINUX bzw. unter anderen Betriebssystemen, die bestimmte Netzdienste in Anspruch nehmen.

Je nachdem, ob ein Protokoll der Anwendungsschicht das verbindungsorientierte Transportprotokoll TCP oder das verbindungslose UDP verwendet, lassen sich die Protokolle der Anwendungsschicht als *verbindungsorientiert*, *verbindungslos* bzw. *gemischt* klassifizieren [Abb. 1.5-1].

Abbildung 1.5-2 listet die wichtigsten Funktionskomponenten der Anwendungsschicht auf.

Verbindungsorientierte Anwendungsprotokolle sind u.a.:

- HTTP: *HyperText Transport Protocol*

Neben SMTP ist HTTP das wichtigste Anwendungsprotokoll im Internet. HTTP sorgt für die Datenübermittlung zwischen Web-Browser und Web-Server. *HTTP over TLS* wird als *HTTPS* bezeichnet.

- SMTP: *Simple Mail Transport Protocol*

Die Übermittlung von E-Mails im Internet erfolgt mithilfe von SMTP. Heute wird in der Regel das *Extended SMTP (ESMTP)* eingesetzt, das eine 8-Bit-transparente Übermittlung der Nachrichten ermöglicht.

- TELNET

TELNET kann als Protokoll – mit dem sich der Anwender in einer interaktiven Sitzung auf einem entfernten Computer einloggen kann – als Urvater der anwendungsbezogenen TCP/IP-Protokolle angesehen werden.

- FTP: *File Transfer Protocol*

FTP dient zur Übermittlung von Dateien zwischen zwei über ein IP-Netz verbundenen Rechnern. Es ist bewusst einfach und robust aufgebaut, sodass die Datenübertragung auch über in der Qualität schlechte Verbindungen (z.B. Satellitenkommunikation) möglich ist. FTP kann auch die TLS-Funktion nutzen. Man spricht dann von *FTPS*.

Verbindungsorientierte Anwendungsprotokolle

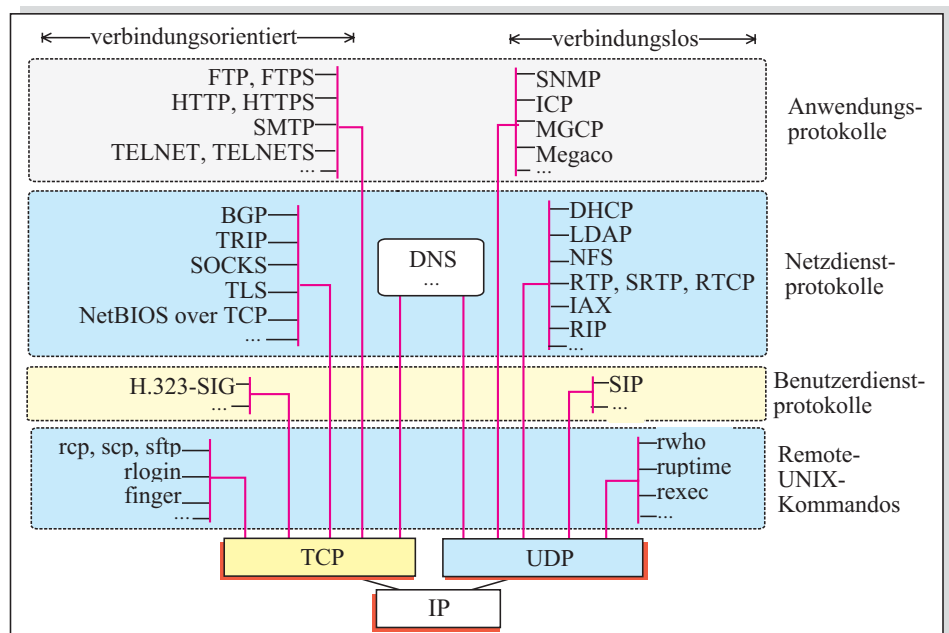


Abb. 1.5-2: Auflistung wichtiger Funktionskomponenten der Anwendungsschicht

Verbindungsorientierte Anwendungsprotokolle

Verbindungsorientierte Anwendungsprotokolle sind u.a.:

- **SNMP: Simple Network Management Protocol**
SNMP ermöglicht die Abfrage der Zustände von Netzwerkkomponenten und liegt dem Netzwerkmanagement zugrunde.
- **ICP: Internet Cache Protocol**
ICP ist ein Protokoll, nach dem Web-Caching-Systeme im Internet kooperieren [BaRS 03].
- **MGCP: Media Gateway Control Protocol**
MGCP ist ein Protokoll zwischen den sog. *VoIP-Gateways* für die Anbindung herkömmlicher Komponenten an VoIP-Systeme [Bada 07]. Das Protokoll *Megaco* entspricht der Funktion nach dem MGCP.

Verbindungsorientierte Dienstprotokolle

Verbindungsorientierte Netzdienstprotokolle sind u.a.:

- **BGP: Border Gateway Protocol**
BGP ist ein Routing-Protokoll und dient der Übermittlung von Routing-Informationen zwischen autonomen Systemen. Auf BGP geht Abschnitt 9.4 näher ein.
- **TRIP: Telephony Routing over IP**
TRIP dient der Übermittlung von Routing-Informationen zwischen autonomen Systemen für die VoIP-Unterstützung. Daher gilt TRIP als Bruder von BGP [Bada 07].
- **SSL/TLS: Secure Socket Layer / Transport Layer Security**
SSL wurde von der Firma Netscape entwickelt, um die Web-Transaktionen zu sichern. Mit Hilfe von SSL werden normalerweise die Daten verschlüsselt übertragen und die beiden Kommunikationspartner können sich gegenseitig authentifizieren. TLS ist der offizielle Nachfolger der SSLv3-Implementierung [Abschnitt 5.3].

- **SOCKS: *SOCKets***

SOCKS dient als Protokoll zwischen einem Client und einem Proxy-Server [Abschnitt 5.2].

- **NetBIOS over TCP/IP (NBoT)**

NetBIOS (Network Basic Input Output System) ist eine Programmschnittstelle (API), die häufig von Windows- und OS/2-Endsystemen genutzt wird. Das Protokoll NBoT legt fest, wie NetBIOS über IP-Netze zu übertragen ist.

Verbindungslose Netzdienstprotokolle sind u.a.:

- **DHCP: *Dynamic Host Configuration Protocol***

Mithilfe von DHCP kann die dynamische Vergabe von IP-Adressen realisiert werden. DHCP wird im Abschnitt 4.2 beschrieben.

- **LDAP: *Lightweight Directory Access Protocol***

LDAP verwendet man bei der Realisierung verteilter Verzeichnisdienste. Es ermöglicht die Abfrage und die Modifikation einer Datenbank im Verzeichnisdienst [Abschnitt 5.4].

- **RTP: *Real-time Transport Protocol***

RTP hat die Aufgabe, zeitkritische Anwendungen wie Audio- und Videokommunikation über ein IP-Netz zu unterstützen. Ihm steht RTCP (*RTP Control Protocol*) zur Seite. RTP ist die Grundlage von VoIP. Eine erweiterte RTP-Version zur sicheren Audio- und Videokommunikation trägt die Bezeichnung SRTP (*Secure RTP*) [Bada 07].

- **IAX: *Inter-Asterisk eXchange***

IAX ist ein kombiniertes Protokoll für die Signalisierung (z.B. bei VoIP) und für den Transport von Echtzeitdaten (Audio, Video) über IP-Netze. Die Version 2 von IAX beschreibt das IETF-Dokument *draft-guy-iax-03*. IAX2 nutzt UDP für den Transport seiner Nachrichten. Bei IAX2 unterscheidet man zwischen *zuverlässigen* und *unzuverlässigen* Nachrichten. Die zuverlässigen Nachrichten transportieren die Signalisierungsangaben und werden von der Empfangsseite bestätigt. Die unzuverlässigen Nachrichten transportieren Echtzeitdaten und werden nicht bestätigt. IAX2 hat viel gemeinsam mit dem Protokoll SCTP.

- **RIP: *Routing Information Protocol***

RIP dient als internes Routing-Protokoll vornehmlich in kleineren IP-Netzen.

Das wohl wichtigste Protokoll im Internet ist *DNS (Domain Name System)*, das sowohl TCP als auch UDP nutzt [Abschnitt 4.1.6]. DNS ist ein gemischtes Netzdienstprotokoll.

Als *Benutzerdienstprotokolle* gelten z.B. die Protokolle für den Aufbau von Verbindungen zwischen IP-Telefonen bei VoIP. Diese Protokolle werden als *Signalisierungsprotokolle* bezeichnet. Hierzu gehören die Signalisierung nach dem ITU-T-Standard H.323 (kurz *H.323-SIG*), die über TCP abgewickelt wird, und das Protokoll SIP (*Session Initiation Protocol*) über UDP. Für weitere Informationen sei auf [Bada 07] verwiesen.

Im Rahmen der Entwicklung von UNIX BSD haben einige spezifische UNIX-Kommandos eine Netzwerk-Erweiterung *r (remote)* erfahren, zu denen z.B. *rlogin*, *rcp*, *rexec* zählen. Hierbei ist zwischen verbindungsorientierten und verbindungslosen Kommandos zu unterscheiden.

*Verbindungslose
Dienstprotokolle*

*Benutzerorientierte
Protokolle*

*Remote-UNIX-
Kommandos*

1.6 IETF und Internet-Standards

<i>IETF</i>	Um die Weiterentwicklung des Internet und seine Anwendungen voranzutreiben, wurde die Organisation <i>Internet Engineering Task Force</i> (IETF) gegründet [http://www.ietf.org]. Zu ihren Aufgaben gehört die Koordination sämtlicher Aktivitäten, die mit der technologischen Weiterentwicklung und der Standardisierung der Internet-Dienste und -Protokolle zusammenhängen. Die IETF-Dokumente werden als sog. RFCs (<i>Request for Comments</i>) im Internet veröffentlicht.
<i>RFCs als Internet-Standards</i>	Ein Schlüssel zur raschen Entwicklung des Internet und der IP-Netze ist vor allem der offene Zugang zu den als RFCs im Internet veröffentlichten IETF-Dokumenten, die als <i>Internet-Standards</i> dienen. Außerdem kann jeder einen neuen RFC vorschlagen, wobei die Vorgehensweise RFC 3700 festlegt.
<i>Datenbank mit RFCs</i>	RFCs reichen bis ins Jahr 1969 zum Vorläufer des Internet zurück. Derzeit (Juni 2007) sind fast 5000 RFCs veröffentlicht. Alle RFCs sind auf mehreren Rechnern im Internet abgespeichert und kostenlos für jeden Nutzer verfügbar. Eine Datenbank mit RFCs, die vom sog. <i>RFC Editor</i> verwaltet wird, ist unter der Adresse http://www.rfc-editor.org/rfcsearch.html zu finden. Die Suche in dieser Datenbank kann durch die Angabe der Nummer des gesuchten RFC oder durch die Angabe eines Suchkriteriums (z.B. Name eines Protokolls wie IP, TCP, OSPF, ...) erfolgen. Eine aktuelle Auflistung von allen RFCs findet man z.B. unter der Adresse http://www.rfc-editor.org/rfc-index2.html
<i>Organisation der IETF</i>	Der Erfolg des Internet ist teilweise der gut durchdachten Organisation der Zusammenarbeit zwischen der IETF und den anderen Institutionen zu verdanken. Welche Institutionen an der Entstehung von Internet-Standards beteiligt sind und wie sie zueinander stehen, zeigt Abbildung 1.6-1.

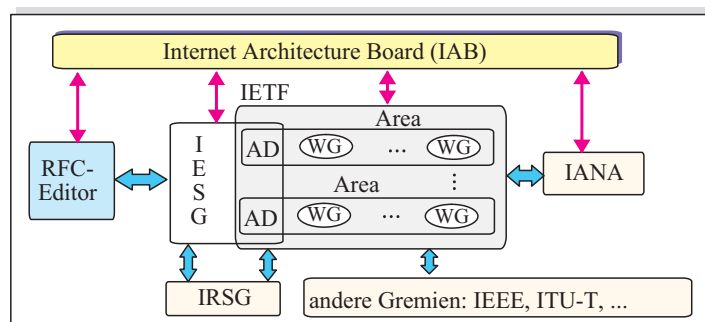


Abb. 1.6-1: Organisation der IETF und die Zusammenarbeit mit anderen Internet-Gremien
 AD: Area Director, IANA: Internet Assigned Numbers Authority, IESG: Internet Engineering Steering Group, IRSG: Internet Research Steering Group, WG: Working Group

Die Entwicklung des Internet wird vom *Internet Architecture Board* (IAB) koordiniert [<http://www.iab.org>]. Dem Vorsitzenden des IAB wurde der Titel *Internet Architect* verliehen. Außerdem wurde im IAB der Posten des *RFC Editor* [<http://www.rfc-editor.org>] eingerichtet, der jeden RFC überprüfen und zur Veröffentlichung vorbereiten soll.

*IAB und
RFC Editor*

Da die Palette von Entwicklungen um das Internet und deren Anwendungsspekte herum sehr breit ist, werden bei der IETF bestimmte Themenbereiche definiert. Ein Themenbereich wird als *Area* bezeichnet. In jeder Area wird ein *Area Director* (AD) benannt, der die Aktivitäten innerhalb der Area koordiniert. Es existieren u.a. folgende Areas: *Applications Area*, *Internet Area*, *Routing Area*, *Security Area*, *Transport Area*.

*Area als
Themen-
bereich*

Für die Entwicklung von Standards zu den einzelnen Themen in jeder Area werden mehrere *Working Groups* (WGs) gebildet. Eine WG übernimmt die Verantwortung für die Entwicklung von Standards, die in der Regel ein Thema (z.B. ein Protokoll oder eine Applikation) betreffen. Eine Auflistung von WGs findet man unter <http://www.ietf.org/html.charters/wg-dir.html>

WG

Hervorzuheben sind u.a. folgende aktive WGs in:

- Internet Area: *dhc* (*Dynamic Host Configuration*), *ipv6* (*IP Version 6*), *mip4* (*Mobility for IPv4*), *mip6* (*Mobility for IPv6*), *l2vpn* (*Layer 2 VPN*), *l3vpn* (*Layer 2 VPN*), *pw3* (*Pseudowire Emulation Edge to Edge*)
- Routing Area: *ccamp* (*Common Control and Measurement Plane*), *mpls* (*Multiprotocol Label Switching*), *ospf* (*Open Shortest Path First*), *pim* (*Protocol Independent Multicast*), *vrp* (*Virtual Router Redundancy Protocol*)
- Transport Area: *dccp* (*Datagram Congestion Control Protocol*), *tsvwg* (*Transport Area Working Group*), *tcpm* (*TCP Maintenance and Minor Extensions*)
- Real-time Applications and Infrastructure Area: *avt* (*Audio/Video Transport*), *enum* (*Telephone Number Mapping*), *sip* (*Session Initiation Protocol*)

Für die technische Verwaltung von IETF-Aktivitäten ist die *Internet Engineering Steering Group* (IESG) verantwortlich. Zur IESG gehören die Direktoren der einzelnen Areas, die ADs. Der Entwurf jedes Internet-Standards, den man als *Internet Draft* bezeichnet, wird vor seiner Spezifikation als RFC innerhalb der IESG diskutiert. Ein Internet Draft wird nur mit der Zustimmung der IESG als Internet-Standard veröffentlicht. Die IESG arbeitet mit dem RFC-Editor zusammen, der für die Veröffentlichung von RFCs zuständig ist.

IESG

Eine besondere Rolle unter den Internet-Gremien spielt die *Internet Assigned Numbers Authority* (IANA). Sie dient als zentrale Stelle für die Registrierung von Internet-Adressen, -Namen, Protokollnummern und anderen Parametern, die weltweit eindeutig sein müssen. Die Auflistung dieser Informationen findet man unter <http://www.iana.org/numbers.html>

IANA

1.7 Schlussbemerkungen

In diesem Kapitel wurden in komprimierter Form vor allem die notwendigen Grundlagen dargestellt, die für die Beschreibung von Ideen, Kommunikationsprotokollen und Systemlösungen für IP-Netze in den weiteren Kapiteln hilfreich sind. Abschließend ist noch auf Folgendes hingewiesen:

- IPv4, IPv6* ■ Es gibt bereits zwei Versionen des Protokolls IP: Die herkömmliche Version, d.h. die Version 4, wird kurz als *IPv4* bezeichnet; die neue Version, die sog. Version 6, bezeichnet man als *IPv6*. IPv4 wird ausführlich in Kapitel 2 beschrieben, IPv6 und seine Dienstprotokolle in Kapitel 6 und 7. Die Konzepte für die Koexistenz von IPv4 und IPv6 präsentiert Kapitel 8. Die in diesem Kapitel dargestellten Grundlagen betreffen sowohl die IPv4- als auch die IPv6-Netze.
- Web-Technologien* ■ Das Internet verdankt die heutige Popularität hauptsächlich dem Web-Dienst mit dem Protokoll HTTP. Aus Platzgründen wurde der Web-Dienst hier aber nur sehr kurz angesprochen. Der Web-Dienst bedeutet heute nicht nur TCP/IP und HTTP, sondern für seine effiziente Realisierung werden verschiedene Technologien eingesetzt, sodass man von *Web-Technologien* spricht. Zu ihnen gehören u.a. die Konzepte und Protokolle für *Web-Switching*, *Web-Caching*, *Content Delivery Networks*. Diese Aspekte werden in diesem Buch außer Acht gelassen; für Näheres sei auf [BaRS 03] verwiesen.
- Multimediale Kommunikation, VoIP* ■ Ein wichtiger Trend bei der Weiterentwicklung des Internet ist die Unterstützung der multimedialen Kommunikation. Hierzu gehören u.a. die Konzepte und Protokolle für VoIP. Bei VoIP handelt es sich um die Echtzeitkommunikation und es ist eigentlich eine Anwendung der IP-Netze. Aus diesem Grund wurde hier auf die Darstellung der Protokolle und Lösungen für VoIP verzichtet. Eine ausführliche Beschreibung der Technik und der Lösungen von VoIP findet man in [Bada 07].
- Sicherheit der IP-Netze* ■ Die IP-Netze werden immer komplexer, ebenso die auf sie einwirkenden Faktoren und die Anforderungen, die an sie gestellt werden. Zudem ist hier ständig mit unterschiedlichen böswilligen Angriffen und anderen Gefährdungen zu rechnen. Die *Sicherheit der IP-Netze* und somit die Maßnahmen zur Vermeidung von Unsicherheiten und hohen Risiken erfordern ausführliche Betrachtungen, ebenso die Darstellung von Angriffen, mit denen man in IP-Netzen rechnen muss. Aus Platzgründen war es hier nicht möglich, die Konzepte und die Protokolle für die Sicherheit von IP-Netzen zu beschreiben.

8 Migration zum IPv6-Einsatz

Die Umstellung aller Rechner, in denen das herkömmliche Internet-Protokoll IPv4 verwendet wird, auf das neue Protokoll IPv6 kann nicht auf einen Schlag geschehen. Dazu sind weltweit viel zu viele Rechner mit IPv4 installiert. Der Schlüssel zur Einführung von IPv6 liegt in der langfristigen und kostengünstigen Migration. Es muss mit einer Übergangszeit gerechnet werden, während der IPv4 und IPv6 parallel eingesetzt werden. Daher benötigt man bestimmte Ansätze und Systemlösungen, um die Integration von IPv4- und IPv6-Netzen zu ermöglichen. Unter dem Begriff *IPv4-Netz* wird jedes beliebige Netz verstanden, in dem alle Systeme IPv4 unterstützen. Ähnlich bezeichnet *IPv6-Netz* ein Netz, in dem sämtliche Systeme IPv6 unterstützen.

*Koexistenz
von IPv4 und
IPv6*

Dieses Kapitel gibt einen Überblick über verschiedene Ansätze und Systemlösungen, um die Koexistenz von IPv4 und IPv6 in verschiedenen Netzstrukturen zu ermöglichen. Nach der Darstellung der Struktur von Rechnern mit IPv4 und IPv6 in Abschnitt 8.1 zeigt Abschnitt 8.2 technische Möglichkeiten für die Integration der IPv4- und IPv6-Netze. Den Einsatz von *IPv6-in-IPv4-Tunneling* präsentiert Abschnitt 8.3. Auf das Konzept von *6to4* geht Abschnitt 8.4 ein. Abschnitt 8.5 erläutert die IPv6-Kommunikation über IPv4-Netze mit ISATAP. Den Einsatz von *Teredo* erläutert Abschnitt 8.6. Auf die Prinzipien der IPv4-Kommunikation über IPv6-Netze mit *DSTM* geht Abschnitt 8.7 ein. Die Integration der IPv4- und IPv6-Netze mithilfe der *Translation IPv4 ⇔ IPv6* wird in Abschnitt 8.8 dargestellt. Abschließende Bemerkungen in Abschnitt 8.9 runden dieses Kapitel ab.

*Überblick
über das
Kapitel*

In diesem Kapitel werden u.a. folgende Fragen beantwortet:

*Ziel dieses
Kapitels*

- Wie kann man sich die beiden Protokolle IPv4 und IPv6 in einem Rechner bzw. in einem Router vorstellen?
- Welche Möglichkeiten der Koexistenz von IPv4 und IPv6 gibt es?
- Wie kann die Kommunikation nach IPv6 über IPv4-Netze realisiert werden?
- Wie kann der Zugang zum IPv6-Internet bereits heute erfolgen?
- Wie lassen sich die sog. *IPv6-Sites* über IPv4-Netze vernetzen?
- Wie können die Rechner in IPv6-Netzen auf das herkömmliche IPv4-Internet zugreifen?
- Welche Möglichkeiten bringt der Einsatz von IPv6 in Netzwerken mit privaten IPv4-Adressen und mit NAT?
- Wie erfolgt die Translation IPv4 ⇔ IPv6 und was ermöglicht sie?

8.1 Integration von IPv4 und IPv6 in Rechnern

Dual-Stack-Rechner und -Router

Es lassen sich einige Situationen in der Praxis vorstellen, in denen die beiden Protokolle IPv4 und IPv6 koexistieren können. In der ersten Phase der Migration zum IPv6-Einsatz werden oft nur einige Rechner und Router um IPv6 erweitert. Derartige Systemkomponenten mit den beiden Protokollstacks IPv4 und IPv6 bezeichnet man als *Dual-Stack-Rechner (Dual-Stack-Host)* und *Dual-Stack-Router*. Das Ziel dieses Abschnittes ist es, die Prinzipien der Kommunikation nach IPv6 zwischen Dual-Stack-Rechnern in IPv4-Netzen zu zeigen.

8.1.1 IPv4- und IPv6-Protokollfamilien im Schichtenmodell

Protokollarchitektur im Dual-Stack-Rechner

Die IPv4-Protokollfamilie im Schichtenmodell wurde bereits Abbildung 1.5-1 gezeigt. Um eine Vorstellung über die Protokollarchitektur eines Dual-Stack-Rechners und eines Dual-Stack-Routers zu vermitteln, zeigt Abbildung 8.1-1 die beiden Protokolle IPv4 und IPv6 im Schichtenmodell. Die hier dargestellte Struktur kann als allgemeine Protokollarchitektur eines Dual-Stack-Rechners angesehen werden. Man findet sie z.B. in einem Server unter Windows 2003 vor, der als Dual-Stack-Rechner bzw. als Dual-Stack-Router dienen kann.

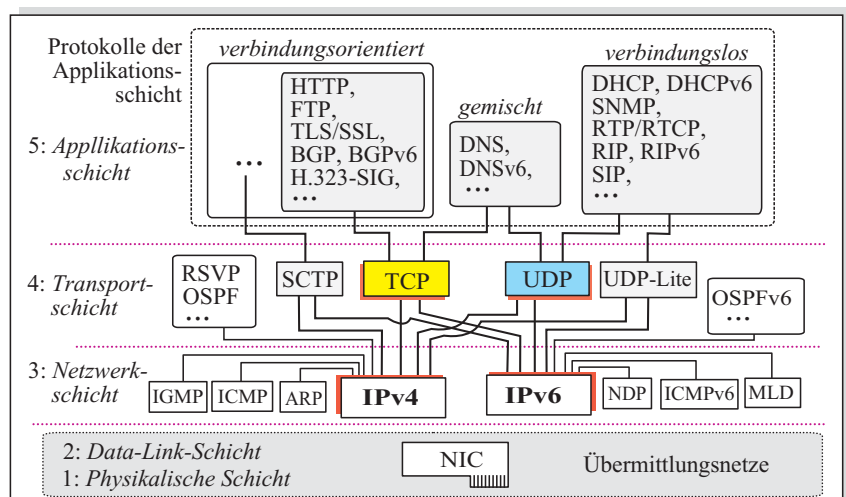


Abb. 8.1-1: Die Protokollfamilien IPv4 und IPv6 im Schichtenmodell
 NIC: Network Interface Controller (Adapterkarte)
 Für Erläuterung von Abkürzungen sei auf das Abkürzungsverzeichnis verwiesen.

Es wurde hier angenommen, dass die beiden untersten Schichten (d.h. die physikalische Schicht und die Data-Link-Schicht) mithilfe einer Adapterkarte rea-

lisiert werden, z.B. in einem Rechner am Ethernet also mit einer Ethernet-Adapterkarte.

Innerhalb der Netzwerkschicht werden die beiden Internet-Protokolle IPv4 und IPv6 mit ihren Hilfsprotokollen angesiedelt. Wie hier ersichtlich ist, wird ARP (*Address Resolution Protocol*), das von IPv4 verwendet wird, bei IPv6 durch NDP (*Neighbor Discovery Protocol*) ersetzt. IGMP (*Internet Group Management Protocol*) wird bei IPv6 durch MLD-Protokoll (*Multicast Listener Discovery*) ersetzt. ICMP (*Internet Control Message Protocol*) wird bei IPv6 zu ICMPv6 erweitert.

*Netzwerk-
schicht*

Die Transportschicht in Abbildung 8.1-1 enthält die gleichen Transportprotokolle wie die Transportschicht in Abbildung 1.5-1. Das Routing-Protokoll OSPF (*Open Shortest Path First*) ist der Transportschicht zuzuordnen und es wird bei IPv6 zu OSPFv6 erweitert.

*Transport-
schicht*

Die Applikationsschicht in Abbildung 8.1-1 enthält im Vergleich zu der Darstellung in Abbildung 1.5-1 auch einige Netzdienstprotokolle von IPv6, wie z.B. DHCPv6 (*Dynamic Host Configuration Protocol*), DNSv6 (*Domain Name System*) und IPv6-Routing-Protokolle wie BGPv6 (*Border Gateway Protocol*) sowie RIPv6 (*Routing Information Protocol*).

*Applika-
tionsschicht*

Unterstützt ein System (z.B. Rechner, Router) IPv4 und IPv6, muss ihm sowohl eine IPv4- als auch eine IPv6-Adresse zugeteilt werden.

8.1.2 Dual-Stack-Rechner in einem LAN-Segment

Den Einsatz von IPv4 und IPv6 in einem physikalischen LAN-Segment illustriert Abbildung 8.1-2. Zwischen IPv4-Rechnern findet die Kommunikation nach IPv4 und zwischen IPv6-Rechnern nach IPv6 statt. Hier wird das ganze Netzwerk in zwei logische „Netzwerkeile“ aufteilt, sodass IPv4-Rechner einen IPv4-Netzteil und entsprechend IPv6-Rechner einen IPv6-Netzteil bilden.

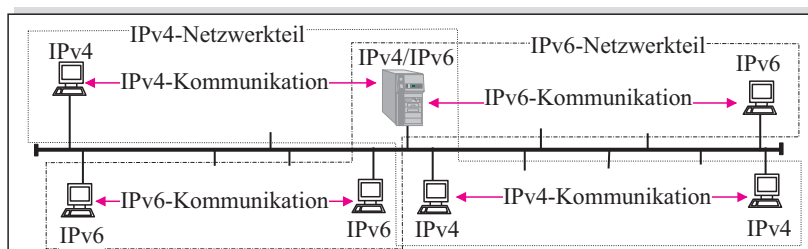


Abb. 8.1-2: Paralleler Einsatz von IPv4 und IPv6 in einem LAN-Segment

Ein Dual-Stack-Rechner mit IPv4 und IPv6 kann sowohl nach IPv4 mit IPv4-Rechnern als auch nach IPv6 mit IPv6-Rechnern kommunizieren. Jeder Dual-Stack-Rechner gehört daher gleichzeitig zu beiden IPv4- und IPv6- Netzteilen.

In der in Abbildung 8.1-2 gezeigten Situation können den IPv6-Rechnern beliebige Unicast-IPv6-Adressen zugeteilt werden, d.h. es müssen nicht unbedingt IPv4-kompatible IPv6-Adressen sein.

8.1.3 Betrieb von Dual-Stack-Rechnern in IPv4-Netzen

Dual-Stack-Rechner = IPv4/IPv6-Rechner

Sollen neue IPv6-Applikationen in einem IPv4-Netz eingesetzt werden, müssen einige IPv4-Rechner um IPv6 erweitert werden. Sie werden damit zu Dual-Stack-Rechnern, die man auch als *IPv4/IPv6-Rechner* bezeichnet, umgerüstet. Ein IPv4-Netz kann somit als *Transitnetz* für die IPv6-Kommunikation zwischen den derart erweiterten Rechnern eingesetzt werden.

Abbildung 8.1-3a zeigt ein Beispiel, in dem zwei Ethernet-Segmente über einen Router miteinander verbunden sind. Da der Router nur IPv4 unterstützt, handelt es sich hierbei um ein „reines“ IPv4-Netz. Werden an diesem Netz auch die Dual-Stack-Rechner angeschlossen, stellt sich die Frage: Wie erfolgt die IPv6-Kommunikation zwischen ihnen? Die Antwort gibt Abbildung 8.1-3b.

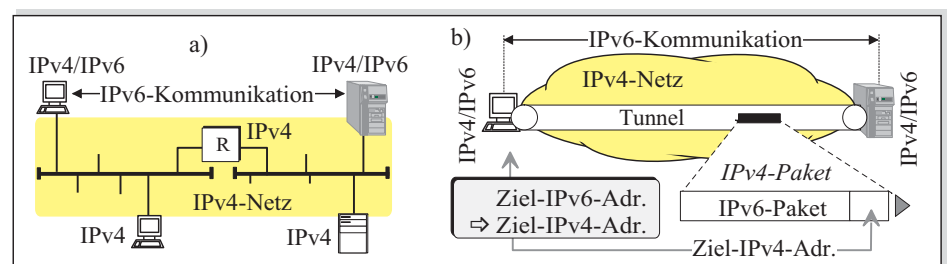


Abb. 8.1-3: IPv6-Kommunikation zwischen Dual-Stack-Rechnern am IPv4-Netz:
 a) physikalische Konfiguration, b) Prinzip der IPv6-Kommunikation
 R: Router

Logischer IPv6-in-IPv4-Tunnel

Bei der IPv6-Kommunikation zwischen IPv4/IPv6-Rechnern über ein IPv4-Netz werden die IPv6-Pakete in IPv4-Pakete eingebettet und als Nutzlast transportiert. Auf diese Art und Weise entsteht ein *logischer IPv6-in-IPv4-Tunnel* über das IPv4-Netz als Transitnetz zwischen den beteiligten IPv4/IPv6-Rechnern. Beginn und Ende des Tunnels über ein IPv4-Netz bestimmen die IPv4-Adressen. Da Datenquelle und -senke bei der IPv6-Kommunikation durch die IPv6-Adressen festgelegt werden, muss der Quell-Rechner eine Adressermittlungstabelle mit der Zuordnung enthalten:

Ziel-IPv6-Adresse \Rightarrow *Ziel-IPv4-Adresse*.

Besteht kein Zusammenhang zwischen IPv4- und IPv6-Adressen, so müssten diese Zuordnungen manuell bei der Rechnerkonfiguration eingegeben werden. Um dies zu vermeiden, können sog. *IPv4-kompatible IPv6-Adressen* verwendet werden. In Abschnitt 6.9.5 wurde bereits gezeigt [Abb. 6.9-8a], dass eine IPv4-Adresse in einer IPv4-kompatiblen IPv6-Adresse enthalten ist. Daher kann jede IPv4-Adresse mit dem Präfix `::/96` zu einer IPv4-kompatiblen IPv6-Adresse erweitert werden. Dadurch können die IPv6-Adressen aus den IPv4-Adressen gebildet werden und dies kann man für die Unterstützung der IPv6-Kommunikation zwischen IPv4/IPv6-Rechnern über IPv4-Netze verwenden [Abb. 8.3-3].

Einsatz von IPv4-kompatiblen IPv6-Adressen

8.2 Arten der Koexistenz von IPv6 und IPv4

Es stehen bereits mehrere Ansätze zur Verfügung, um die Koexistenz von IPv4 und IPv6 in verschiedenen Netzstrukturen zu ermöglichen. Abbildung 8.2-1 illustriert die möglichen Alternativen.

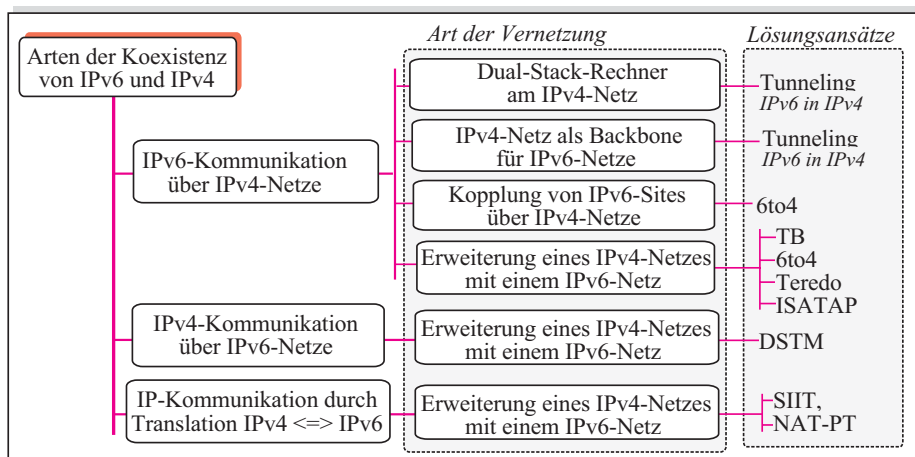


Abb. 8.2-1: Die Zusammenstellung der Ansätze für die Koexistenz von IPv6 und IPv4
 6to4: 6to4 Transition Mechanism, DSTM: Dual-Stack Transitdom Mechanism,
 ISATAP: Intra-Site Automatic Tunnel Addressing Protocol, NAT-PT: Network Address Translation – Protocol Translation, SIIT: Stateless IP/ICMP Translation Algorithm,
 TB: Tunnel Broker, Teredo: Tunneling IPv6 over UDP through NATs

Es kommen folgende Arten der Koexistenz von IPv4 und IPv6 infrage:

■ IPv6-Kommunikation über IPv4-Netze

Arten der Koexistenz

Es handelt sich hier um die Kopplung von Rechnern mit IPv6 über IPv4-Netze bzw. um die Erweiterung der IPv4-Netze mit IPv6-Netzen. In diesem Fall unterscheidet man zwischen den folgenden Vernetzungsarten:

- *Einsatz von Dual-Stack-Rechnern an einem IPv4-Netz*: Dies ist durch das IPv6-in-IPv4-Tunneling möglich [Abb. 8.1-3b].
- *IPv4-Netz als Backbone bzw. als Transitnetz für IPv6-Netze*: Dies ist ebenfalls durch das IPv6-in-IPv4-Tunneling möglich [Abb. 8.3-4].
- *Kopplung von IPv6-Sites über IPv4-Netze*: In einem IPv4-Netz können einige „Inseln“ mit nur IPv6-Systemkomponenten eingerichtet werden. Solche IPv6-Inseln werden als *IPv6-Sites* bezeichnet. Die Kopplung von IPv6-Sites über IPv4-Netze ermöglicht das Konzept 6to4 [Abschnitt 8.4].
- *Erweiterung eines IPv4-Netzes mit einem IPv6-Netz*: Ein IPv4-Netz kann „räumlich“ mit einem IPv6-Netz erweitert werden. Um dies zu erreichen, stehen folgende Konzepte zur Verfügung: Tunnel Broker [Abschnitt 8.3.3], 6to4, ISATAP [Abschnitt 8.5] und Teredo [Abschnitt 8.6].

Auf IPv6-Kommunikation über IPv4-Netze geht Abschnitt 8.2.1 näher ein.

■ IPv4-Kommunikation über IPv6-Netze

Es handelt sich hier um den Einsatz von Dual-Stack-Rechnern in einem IPv4-Netz bzw. um eine räumliche Erweiterung eines IPv4-Netzes mit einem IPv6-Netz [Abschnitt 8.2.2]. Diese Art der Kommunikation ist mithilfe von DSTM [Abschnitt 8.7] möglich.

■ IP-Kommunikation durch Translation IPv4 ↔ IPv6

Zwischen einem IPv4-Netz und einem IPv6-Netz kann ein Router eingesetzt werden, in dem der IPv4-Header auf den IPv6-Header und umgekehrt umgesetzt werden kann. Es handelt sich daher um eine *Translation IPv4 ↔ IPv6* [Abschnitt 8.2.3]. Man kann in diesem Fall von *IP-Kommunikation* zwischen IPv4-Rechner und IPv6-Rechner sprechen. Für die Unterstützung dieser Art der Kommunikation stehen SIIT [Abschnitt 8.8.1] und NAT-PT [Abschnitt 8.8.2] zur Verfügung.

8.2.1 IPv6-Kommunikation über IPv4-Netze

Eine IPv4-Netzinfrastruktur wird nicht innerhalb einer Nacht auf IPv6 umgestellt. Stattdessen werden zunächst einige Rechner um IPv6 erweitert bzw. kleine IPv6-Inseln eingerichtet. Eine bestehende IPv4-Netzinfrastruktur kann daher für die Unterstützung der IPv6-Kommunikation verwendet werden. In der ersten Phase der Migration zum Einsatz von IPv6 können die bestehenden IPv4-Netze als Transitnetze fungieren. Abbildung 8.2-2 zeigt eine Zusammenstellung von Lösungen, bei denen IPv4-Netze als Transitnetze für die Unterstützung der IPv6-Kommunikation dienen.

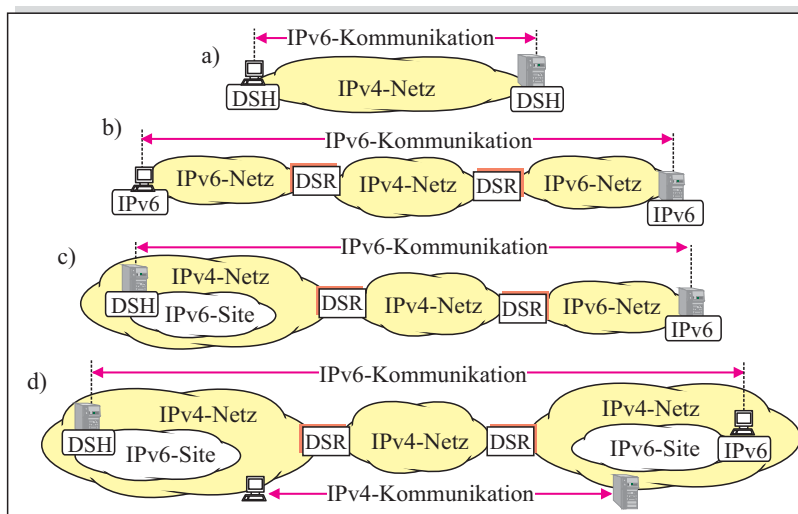


Abb. 8.2-2: IPv4-Netze als Transitnetze für die Unterstützung der IPv6-Kommunikation:
a) Dual-Stack-Rechner am IPv4-Netz, b) IPv4-Netz als Transitnetz für IPv6-Netze,
c) IPv4-Netz als Zubringer zum IPv6-Netz, d) Vernetzung von IPv6-Sites
DSH: Dual-Stack-Host, DSR: Dual-Stack-Router

Dient ein IPv4-Netz als Transitnetz bei der IPv6-Kommunikation, so kann es sich um folgende Vernetzungsarten handeln:

*IPv4-Netz
als
Transitnetz*

- a. Einsatz von Dual-Stack-Rechnern (-Hosts) am IPv4-Netz [Abb. 8.2-2a]
Für die IPv6-Kommunikation zwischen zwei Dual-Stack-Rechnern am IPv4-Netz wird ein IPv6-in-IPv4-Tunnel aufgebaut [Abb. 8.1.3b]. Darauf geht Abschnitt 8.3 näher ein.
- b. Das IPv4-Netz fungiert als Transitnetz für IPv6-Netze [Abb. 8.2-2b]
Um die IPv6-Kommunikation bei dieser Vernetzungsart zu ermöglichen, wird ebenfalls das IPv6-in-IPv4-Tunneling eingesetzt. Abschnitt 8.3 präsentiert dies näher.
- c. Das IPv4-Netz dient als Zubringer zum IPv6-Netz [Abb. 8.2-2c]
In einem IPv4-Netz kann eine „IPv6-Insel“ eingerichtet werden. Sie wird auch *IPv6-Site* genannt. Ein IPv4-Netz kann dann für Rechner aus der IPv6-Site als Zubringer zu einem IPv6-Netz dienen. Um dies zu ermöglichen, steht das in Abschnitt 8.4 dargestellte Konzept *6to4* zur Verfügung.
- d. Vernetzung von IPv6-Sites über IPv4-Netze [Abb. 8.2-2d]
Diese Vernetzungsart ist auch mithilfe von *6to4* möglich. Dies wird in Abschnitt 8.4 detailliert dargestellt.

*IPv6-Insel
als IPv6-Site*

Bei der Migration zu IPv6 kann ein bestehendes IPv4-Netz um ein IPv6-Netz bzw. um eine IPv6-Site erweitert werden. Abbildung 8.2-3 illustriert dies.

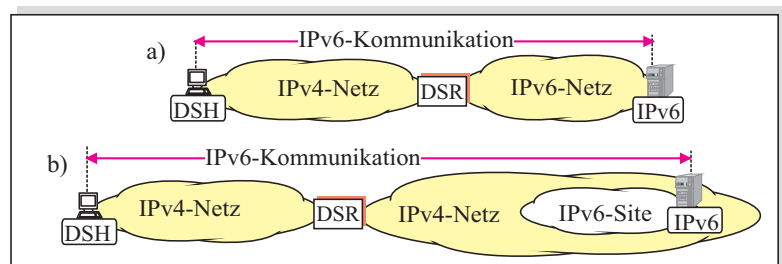


Abb. 8.2-3: IPv6-Kommunikation zwischen Dual-Stack-Rechner am IPv4-Netz und:
 a) IPv6-Rechner in einem IPv6-Netz, b) IPv6-Rechner in einer IPv6-Site
 Abkürzungen wie in Abbildung 8.2-2

Wird ein IPv4-Netz um ein IPv6-Netz bzw. um eine IPv6-Site erweitert, so handelt es sich um die IPv6-Kommunikation zwischen einem Dual-Stack-Rechner am IPv4-Netz auf der einen Seite und auf der anderen Seite

- a. einem IPv6-Rechner in einem IPv6-Netz [Abb. 8.2-3a]
 Diese IPv6-Kommunikation wird mithilfe von IPv6-in-IPv4-Tunneling realisiert [Abschnitt 8.3].
- b. einem IPv6-Rechner in einer IPv6-Site [Abb. 8.2-3b]
 Diese IPv6-Kommunikation ermöglicht das Konzept 6to4 [Abschnitt 8.4].

8.2.2 IPv4-Kommunikation über IPv6-Netze

*Bedeutung
 von IPv4
 über IPv6*

Es sollte möglich sein, dass Rechner in IPv6-Netzen auf die Ressourcen im IPv4-Internet zugreifen können. Dafür muss in Rechnern im IPv6-Netz zusätzlich IPv4 installiert werden. Daher ist der Betrieb von Dual-Stack-Rechnern im IPv6-Netz von großer Bedeutung, genauso wie die Möglichkeit, dass sie die IPv4-Kommunikation zu Rechnern in IPv4-Netzen initiieren können. Wie Abbildung 8.2-4 zeigt, handelt es sich hier um die IPv4-Kommunikation über ein IPv6-Netz, also um eine Art von *IPv4 über IPv6*.

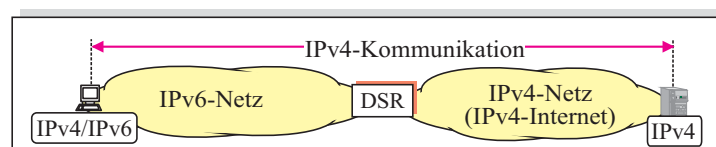


Abb. 8.2-4: IPv4-Kommunikation zwischen Dual-Stack-Rechnern im IPv6-Netz und IPv4-Rechnern am IPv4-Netz
 DSR: Dual-Stack-Router,

Um IPv4 über IPv6 zu unterstützen, wurde das Konzept DSTM (*Dual Stack Transition Mechanism*) entwickelt. DSTM wird in Abschnitt 8.7 präsentiert.

12 Virtual Private Networks und Remote Access Services

In der Vergangenheit wurden standortübergreifende private Netze so aufgebaut, dass mehrere Standorte über gemietete physikalische Standleitungen verbunden waren. Da die Standleitungen in der Regel teuer waren, haben derartige Lösungen mit einer Vielzahl von weit voneinander entfernten Standorten zu sehr hohen Kosten geführt. Mithilfe von sog. *Tunneling-Techniken* lassen sich virtuelle Standleitungen für den Transport von vertraulichen Daten über öffentliche IP-Netze aufbauen. Setzt man solche virtuellen Standleitungen ein, um mehrere Netzwerke an verschiedenen Standorten miteinander zu verbinden, entsteht ein VPN (*Virtual Private Network*). Für den Aufbau von VPN verwendet man sog. *Tunneling-Protokolle*.

Was ist ein VPN?

Oft werden die *Remote Access Services* (RAS) mit VPNs so integriert, dass die Verbindung zwischen einem Remote-Benutzer und dem Intranet über ein öffentliches IP-Netz in einem „Tunnel“ verläuft. Beim RAS in VPNs müssen die Remote-Benutzer entsprechend authentifiziert werden. Dafür steht das Protokoll RADIUS (*Remote Access Dial-In User Service*) zur Verfügung.

VPNs mit RAS

Dieses Kapitel gibt einen fundierten Überblick über die technischen Konzepte von VPNs und den RAS. Abschnitt 12.1 erläutert kurz verschiedene VPN-Arten. Danach werden in Abschnitt 12.2 Provider Provisioned VPNs (PPVPN) präsentiert. Auf Layer-2-Tunneling über klassische IP-Netze geht Abschnitt 12.3 ein. Abschnitt 12.4 erläutert die Funktionsweise des IPsec und von Layer-3-Tunneling. Das Konzept von RAS und RADIUS präsentiert Abschnitt 12.5. Abschließende Bemerkungen in Abschnitt 12.6 runden das Kapitel ab.

Überblick über das Kapitel

In diesem Kapitel werden u.a. folgende Fragen beantwortet:

Ziel dieses Kapitels

- Welche Konzepte liegen den VPNs über klassische IP-Netze und MPLS-Netze zugrunde?
- Wie kann man sich ein VPN vorstellen und welche Ansätze dafür gibt es?
- Welche Arten von PPVPNs gibt es und wie werden sie eingerichtet?
- Wie können die Layer-2-Übermittlungsdienste (z.B. Ethernet) über IP-Netze bereitgestellt werden?
- Wie kann ein privates Ethernet auf Basis eines MPLS-Netzes aufgebaut werden und welche Bedeutung hat hierbei VLAN-Stacking?
- Wie funktioniert das IPsec und wie werden VPNs mit dem IPsec aufgebaut?

12.1 Grundlagen und Arten von VPNs

*Site als
Netzwerk
an einem
Standort*

Ein VPN stellt eine Vernetzung mehrerer Netzwerke eines Unternehmens bzw. einer anderen Institution (z.B. einer Hochschule), die sich an verschiedenen Standorten befinden, mithilfe von virtuellen Standleitungen über IP-Netze dar. In der englischen Literatur und in allen VPNs betreffenden Standards wird ein Netzwerk an einem Standort eines Unternehmens als *Site* bezeichnet, daher wird dieser Begriff auch hier verwendet. Da es sich hierbei natürlich um IP-Netzwerke handelt, können Sites als *Intranets* – also als private Internets – angesehen werden.

Tunneling

Die Eigenschaften einer virtuellen Standleitung über ein IP-Netz u.a. im Hinblick auf die Datensicherheit können mit einem Tunnel verglichen werden. Aus diesem Grund spricht man von *Tunneling über IP-Netze*. Das Tunneling liegt den VPNs zugrunde. Darauf wird in Abschnitt 12.1.1 näher eingegangen.

Bei IP-Netzen ist zwischen klassischen IP-Netzen und MPLS-Netzen [Kapitel 11] zu unterscheiden. Das führt dazu, dass man auch unterscheidet zwischen

- VPNs auf Basis klassischer IP-Netze und
- VPNs auf Basis der MPLS-Netze.

Einen Überblick über grundlegende Arten von VPNs gibt Abschnitt 12.1.2.

12.1.1 Tunneling als Basis von VPNs

*Was ist
Tunneling?*

Das Tunneling ist ein Konzept, nach dem man beliebige Datenpakete aus einem Netzwerk an einem Standort (*Site*), in der Regel aus einem Intranet, über ein Weitverkehrsnetz als reines Transitnetz transportiert. Dabei spielen die Adressierung und das verwendete Protokoll keine Rolle. Daher ist es mithilfe des Tunneling möglich, mehrere Sites über ein IP-Weitverkehrsnetz transparent zu koppeln. Das Tunneling wird schon seit Langem eingesetzt, um IP-Pakete über andere Netze zu transportieren, in denen ein anderes Protokoll verwendet wird.

Tunneling über klassische IP-Netze

*IP-Pakete
als
Container
für L3-
Pakete*

Abbildung 12.1-1 illustriert das Tunneling über ein klassisches verbindungsloses IP-Netz, das nach dem Datagram-Prinzip funktioniert [Abb. 11.1-1a]. Hier wird das ganze über das IP-Netz zu übertragende Original-L3-Paket (d.h. ein Paket nach einem Layer-3-Protokoll wie z.B. IP, IPX) aus Site A innerhalb der Randkomponente beim *Network Service Provider* (NSP), die man oft als PE (*Provider Edge*) bezeichnet, als Nutzlast (*Payload*) in ein neues IP-Paket verpackt. Diese Verpackung wird auch als *Encapsulation* bezeichnet und besteht darin, dass dem Original-L3-Paket zuerst ein Header nach einem *Tunneling-*

Protokoll und dann noch ein IP-Header vorangestellt werden. Im Ziel-PE werden diese beiden vorangestellten Header entfernt, sodass das Original-L3-Paket wieder in der ursprünglichen Form vorliegt und an den Zielrechner in Site B übermittelt werden kann. Man nennt dies *Decapsulation*. Diese sog. *Transit-Pakete* dienen als *Container* für die Übermittlung verschiedener L3-Pakete.

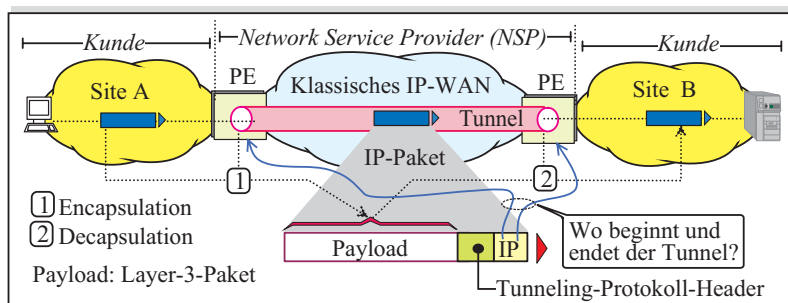


Abb. 12.1-1: Das Tunneling-Prinzip bei der Datenübermittlung über ein klassisches IP-WAN
PE: Provider Edge (Router, Layer-2/3-Switch)

Der IP-Header, der dem Original-L3-Paket im Quell-PE vorangestellt wird, bestimmt Beginn und Ende des Tunnels sowie den Übertragungsweg über das IP-Netz. Der zweite vorangestellte Header, der nach dem IP-Header folgt, wird von einem Tunneling-Protokoll, wie z.B. L2TP (*Layer 2 Tunneling Protocol*) bzw. IPsec (*IP Security*), festgelegt. Im Header des Tunneling-Protokolls können weitere Angaben zu den PEs, die Beginn und Ende des Tunnels realisieren, übermittelt werden. Insbesondere werden oft Angaben gemacht, um das Original-L3-Paket über das Transit-IP-Netz verschlüsselt zu transportieren.

Wozu ein Tunneling-Protokoll?

Bemerkung: Bei der Bildung von VPNs über klassische IP-Netze wird der PE oft als *VPN-Gateway (VG)* bezeichnet.

Da der Transport des Original-L3-Pakets über das IP-Netz auf Basis des Transit-IP-Pakets verläuft, können eventuelle Lauscher unterwegs die Quell- und die Zieladresse des verschlüsselt übertragenen L3-Pakets nicht ablesen, sondern lediglich nachvollziehen, dass die Daten zwischen Tunnel-Beginn und -Ende transportiert werden. Ein Tunnel über ein IP-Netz verhält sich also wie eine bidirektionale Direktverbindung zwischen den Systemkomponenten, die Tunnel-Beginn und -Ende bilden. Somit ist jeder Tunnel als eine virtuelle Standleitung zu interpretieren.

Virtuelle Standleitung

Tunneling über MPLS-Netze

Um mehrere Standorte eines Unternehmens bzw. einer anderen Institution zu vernetzen, eignen sich MPLS-Netze besonders gut [Abschnitt 11.2]. Über ein MPLS-Netz kann eine bidirektionale Punkt-zu-Punkt-Verbindung zwischen

Pseudo-Drahtverbindung

zwei Standorten mithilfe zweier entgegengerichteter LSPs (*Label Switched Path*) aufgebaut werden [Abb. 11.2-1 und -2], was quasi als Drahtverbindung betrachtet werden kann. In diesem Zusammenhang spricht man von einem *Pseudo-Draht* (*Pseudo Wire, PW*) bzw. von einer *Pseudo-Drahtverbindung* über ein MPLS-Netz. Eine derartige Verbindung wird zwischen den zwei Randkomponenten PE (*Provider Edge*) eines Network Service Provider (NSP) eingerichtet, an die verschiedene Sites angeschlossen sein können. Es handelt sich hier um eine Nachbildung (Emulation) einer Edge-zu-Edge-Drahtverbindung.

Um eine Pseudo-Drahtverbindung über MPLS-Netze einzurichten, wird das Tunneling verwendet. Abbildung 12.1-2 illustriert dies.

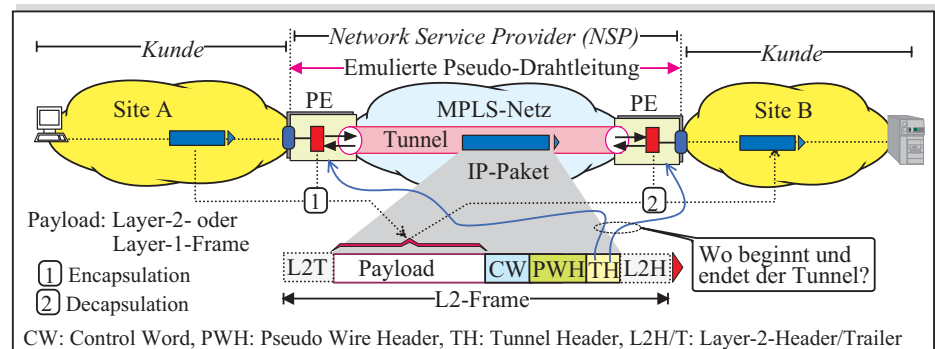


Abb. 12.1-2: Das Tunneling-Prinzip bei der Datenübermittlung über ein MPLS-Netz

*L2-Frames
als
Container
für L1/2-
Daten*

Der Quell-PE am Standort A verpackt die zu übertragenden Original-Daten als Payload (Nutzlast) in einen L2-Frame (*Encapsulation*). Beim L2-Frame handelt es sich um einen Ethernet-Frame, FR-Frame bzw. AAL5-Frame [Abb. 10.4-3], falls entsprechend Ethernet, FR- oder ATM-Netz als Übermittlungsnetz beim MPLS dient. Im Ziel-PE werden die eingekapselten Daten aus dem L2-Frame herausgenommen (*Decapsulation*). Da die Original-Daten als L1- bzw. L2-Frames in einem L2-Frame übermittelt werden, sind sie im MPLS-Netz als Transitnetz nicht zugänglich. Dies könnte man sich als Übermittlung von Daten in einem Tunnel vorstellen.

*Payload-
Arten*

Beim Tunneling über MPLS-Netze handelt es sich in der Regel um einen bidirektionalen Tunnel, der durch zwei entgegengerichtete LSPs gebildet wird. Auf Basis eines solchen Tunnels wird dann eine Pseudo-Drahtverbindung realisiert, über die L2-Frames als quasi Container übermittelt werden. Die im L2-Frame als Payload übermittelten Daten können sein:

- die Layer-1-Daten, z.B. ein TDM-Frame (*Time Division Multiplexing*) bzw.
- die Layer-2-Daten, wie z.B. Ethernet- bzw. PPP-Frame.

Somit kann eine emulierte Leitung als virtuelle Layer-1- bzw. Layer-2-Leitung angesehen werden.

Wie aus Abbildung 12.1-2 ersichtlich, werden der im L2-Frame zu übertragenden Payload folgende drei Header vorangestellt:

Was wird der Payload vorangestellt?

- **Tunnel-Header (TH)** mit einem äußeren Label (sog. *Outer Label*), nach dem der L2-Frame im MPLS-Netz übermittelt wird. Dieses Label bestimmt Beginn und Ende des Tunnels.
- **Pseudo-Wire-Header (PWH)** mit einem inneren Label (sog. *Inner Label*), das als Identifikation der emulierten Leitung dient. Dieses Label kann auch als Identifikation des Kunden angesehen werden.
- **Control Word (CW)**, um verschiedene L1/2-Frames über eine Pseudo-Drahtleitung auf gleiche Art und Weise übermitteln zu können.

12.1.2 Arten von VPNs

VPNs können nach verschiedenen Merkmalen klassifiziert werden. Ein VPN ist u.a. davon abhängig, welche Systemkomponenten die virtuellen Standleitungen als Tunnel verbinden. Die Führung des Tunnels bestimmt daher die Art des VPN. Abbildung 12.1-3 bringt dies zum Ausdruck.

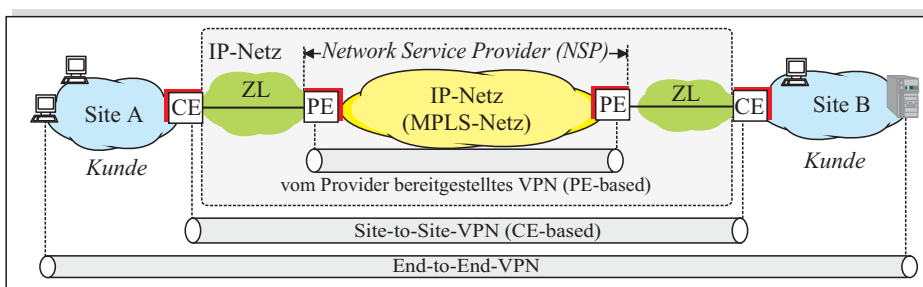


Abb. 12.1-3: Führung des Tunnels bestimmt die VPN-Art

CE: Customer Edge, PE: Provider Edge, ZL: Zubringerleitung bzw. -netz

Wird ein Tunnel zwischen zwei Randkomponenten PE bei einem NSP (*Network Service Provider*) eingerichtet, handelt es sich um ein *vom Provider bereitgestelltes VPN*. Man bezeichnet derartige VLANs kurz als *PPVPNs* (*Provider Provisioned VPN*). Sie werden in der Regel auf Basis der MPLS-Netze gebildet und stellen ein breites Spektrum der Kommunikationsdienste zur Verfügung. Auf PPVPNs geht Abschnitt 12.2 näher ein.

Vom Provider bereitgestellte VPNs

Wird ein Tunnel zwischen zwei Randkomponenten CE bei einem Kunden eines NSP – also zwischen mehreren Standorten eines Unternehmens bzw. einer anderen Institution – eingerichtet, spricht man vom *Site-to-Site-VPN*. Ein solches

Site-zu-Site-VPN

VPN stellt eine Lösung dar, um mehrere Sites über ein IP-Netz standortübergreifend zu verbinden. Für den Aufbau von Site-to-Site-VPNs verwendet man oft das Protokoll IPsec [Abschnitt 12.4.9].

End-to-End-VPN

Werden die kommunizierenden Rechner mit einem Tunnel über ein IP-Netz verbunden, bezeichnet man das als *End-to-End-VPN*. Ein solches VPN ermöglicht es, mehrere Remote-Rechner über virtuelle Standleitungen mit einem zentralen Rechner zu verbinden. Hierbei ist zu beachten, dass ein entsprechendes Tunneling-Protokoll auf jedem an das VPN „angeschlossenen“ Rechner installiert werden muss. End-to-End-VPNs können mithilfe des IPsec eingerichtet werden. Auf Konzept und Einsatz des IPsec geht Abschnitt 12.4 detaillierter ein.

Remote-Access-VPN

Soll Remote Access auf Intranet-Dienste in einem VPN ermöglicht werden, so kann ein Remote-Benutzer die Verbindungen über ein Zugbringernetz zu einem Zugangskonzentrator aufbauen, wo der Tunnel über ein IP-Netz zu einem bestimmten Intranet beginnt. Eine solche Lösung stellt ein *Remote-Access-VPN* (*RA-VPN*) dar. Abbildung 12.1-4 illustriert das Konzept des RA-VPN.

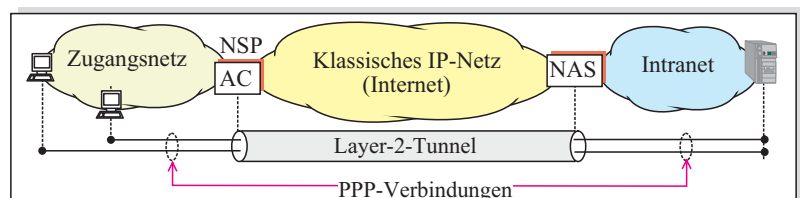


Abb. 12.1-4: Grundlegendes Konzept von Remote-Access-VPNs
AC: Access Concentrator, NAS: Network Access Server

Bei RA-VPNs handelt sich um VPNs, denen das Layer-2-Protokoll PPP (*Point-to-Point Protocol*) zugrunde liegt [Abschnitt 10.2.2]. Daher spricht man auch von *Layer-2-Tunnel* bzw. von *Layer-2-Tunneling*. RA-VPNs sind somit als *PPP-basierte VPNs* zu bezeichnen. Zwischen den kommunizierenden Rechnern wird eine PPP-Verbindung aufgebaut. Über einen Tunnel können mehrere solche Verbindungen verlaufen.

Klassifizierung von VPNs

Bei der Klassifizierung von VPNs ist zuerst zu unterscheiden zwischen

- VPNs, die auf Basis der klassischen verbindungslosen IP-Netze (d.h. Datagram-Netze) aufgebaut wurden, und
- VPNs, die auf Basis der verbindungsorientierten MPLS-Netze eingerichtet worden sind.

Wie Abbildung 12.1-5 zeigt, können weitere Arten von VPNs innerhalb dieser beiden Klassen unterschieden werden.

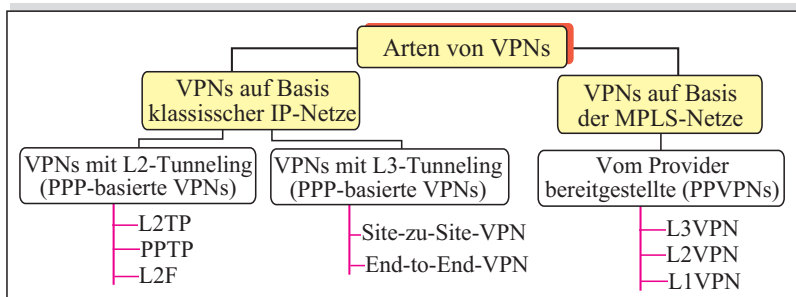


Abb. 12.1-5: Grundlegende VPN-Arten auf Basis von IP-Netzen
 IPsec: IP Security, L2TP: Layer 2 Tunneling Protocol, Ln: Layer n,
 L2F: Layer 2 Forwarding, PPTP: Point-to-Point Tunneling Protocol

Den vom Provider bereitgestellten VPNs auf Basis der MPLS-Netze wird Abschnitt 12.2 gewidmet. Auf die VPNs mit Layer-2- und Layer-3-Tunneling gehen die Abschnitte 12.3 und 12.4 ein.

12.2 Vom Provider bereitgestellte VPNs

Vom Provider bereitgestellte VPNs, die man kurz als PPVPNs (*Provider Provisioned VPN*) bezeichnet, werden auf Basis der MPLS-Netze aufgebaut. Zum Aufbau eines PPVPN werden emulierte Leitungen über ein MPLS-Netz eines NSP eingerichtet. Es ist zwischen verschiedenen Typen von PPVPNs zu unterscheiden. Abbildung 12.2-1 bringt dies näher zum Ausdruck.

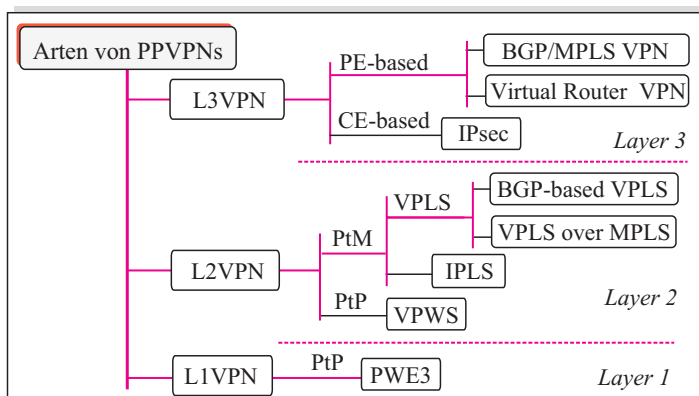


Abb. 12.2-1: Klassifizierung verschiedener Arten von PPVPNs
 BGP: Border Gateway Protocol, CE: Customer Edge, IPLS: IP-Only LAN Service,
 Ln: Layer n, PtP: Point-to-Point, PtM: Point-to-Multipoint, PE: Provider Edge,
 VPLS: Virtual Private LAN Services

- In Abhängigkeit davon, welche Datenformate – also Layer-1-Frame, Layer-2-Frame bzw. Layer-3-Pakete – übermittelt werden, kann ein PPVPN dem Layer 1, 2 bzw. 3 zugeordnet werden. Man bezeichnet sie dementsprechend als *Layer-1-VPN* (L1VPN), *Layer-2-VPN* (L2VPN) oder *Layer-3-VPN* (L3VPN).
- CE-based* bzw. *PE-based* Wird eine virtuelle Leitung in einem VPN zwischen zwei Randkomponenten CE eines Kunden eingerichtet [Abb. 12.1-3], nennt man dies *CE-based VPN*. Verläuft eine virtuelle Leitung nur zwischen zwei Randkomponenten PE eines Providers, bezeichnet man dies als *PE-based VPN*. L1VPNs und L2VPNs werden in der Regel als PE-based VPNs realisiert. Bei L3VPNs ist zwischen CE-based und PE-based VPNs zu unterscheiden.
- L1VPNs* Ein L1VPN stellt eine Punkt-zu-Punkt-Verbindung von zwei Sites über eine emulierte physikalische Leitung (virtuelle L1-Leitung, Pseudo Wire) dar. Die IETF-Standards für L1VPN werden als *Pseudo Wire Emulation Edge-to-Edge* (PWE3) bezeichnet. Ein Layer-1-VPN kann eine Kopplung von zwei TDM-Systemen (*Time Division Multiplexing*) darstellen.
- L2VPNs* Wird eine virtuelle L2-Leitung zum Aufbau eines VPN eingesetzt, handelt es sich um ein L2VPN. Die wichtigste Besonderheit von L2VPN ist, dass die Layer-2-Frames über L2VPN übermittelt werden. Ein L2VPN kann entweder
- ein PtP-L2VPN (*Punkt-zu-Punkt L2VPN*) oder
 - ein PtM-L2VPN (*Punkt-zu-Mehrpunkt L2VPN*) sein.
- PtP-L2VPN* Ein PtP-L2VPN stellt eine direkte Verbindung von zwei Standorten über eine virtuelle Leitung dar. Daher bezeichnet man eine derartige Lösung als *Virtual Private Wire Service* (VPWS). Sie eignet sich insbesondere für die Kopplung von ATM- bzw. Frame-Relay-Systemen und für die Unterstützung von PPP-Verbindungen über MPLS-Netze.
- PtM-L2VPN* Das PtM-L2VPN dagegen stellt eine Vernetzung von mehreren Standorten mithilfe von virtuellen Leitungen dar, die über ein MPLS-Netz eines Providers verlaufen. Ein derartiges L2VPN eignet sich insbesondere für die Vernetzung von LANs, wie z.B. Ethernets. Bei der Unterstützung der LAN-Kommunikation über MPLS-Netze unterscheidet man zwischen den folgenden zwei Ansätzen:
- VPLS (*Virtual Private LAN Service*) und
 - IPLS (*IP-Only LAN Service*).
- Beim VPLS handelt es sich um ein Konzept, nach dem das ganze MPLS-Netz sich als Layer-2-Switch verhält. Dies bedeutet eine *LAN-Emulation* innerhalb eines MPLS-Netzes. Sie ermöglicht sogar, ein weltweites Ethernet einzurichten, in dem ein MPLS-Netz als Backbone fungiert. Der IPLS stellt ein Konzept für die Vernetzung einzelner LAN-Komponenten über ein MPLS-Netz dar.

13 Unterstützung der Mobilität in IP-Netzen

Mobile Rechner mit Internet-Anschluss, insbesondere Laptops mit WLAN-Adaptoren, werden immer beliebter. Das Protokoll, das die Mobilität in IP-Netzen unterstützt, heißt *Mobile IP (MIP)*. Ein mobiler Rechner mit dem MIP kann z.B. ein Subnetz während einer bestehenden und aktiven TCP-Verbindung bzw. einer anderen Session wechseln, ohne diese abbrechen zu müssen.

Mobilität mit MIP

Um die Mobilität in IP-Netzen mit dem IPv6 zu ermöglichen, wurde das *Mobile IPv6 (MIPv6)* entwickelt. Das MIPv6 kann als Erweiterung des IPv6 im Hinblick auf die Unterstützung der Mobilität angesehen werden. Falls ein mobiler Rechner während einer bestehenden Session ein Subnetz wechselt, darf die bestehende Session nicht abgebrochen werden. Die Übernahme eines mobilen Rechners in einem neuen Subnetz während einer Session wird *Handover* genannt. Weil eine bestehende Session beim Handover „eingefroren“ wird, muss die Dauer des Handover so weit wie möglich reduziert werden, um die Qualität von Echtzeitapplikationen, wie z.B. Voice over IP, nicht negativ zu beeinflussen. Hierfür wurde HMIPv6 (*Hierarchical Mobile IPv6*) entwickelt.

MIPv6 und HMIPv6

Dieses Kapitel stellt die Ansätze und die Protokolle für die Unterstützung der Mobilität in IP-Netzen dar. Abschnitt 13.1 geht kurz auf die verschiedenen Ansätze ein. Die Prinzipien von Roaming zwischen öffentlichen WLANs (Hotspots) erläutert Abschnitt 13.2. Die Funktionsweise des MIPv4 wird in Abschnitt 13.3 dargestellt. Abschnitt 13.4 erläutert das Konzept des MIPv6. Auf das HMIPv6 geht Abschnitt 13.5 ein. Ergänzende Bemerkungen in Abschnitt 13.6 schließen dieses Kapitel ab.

Überblick über das Kapitel

In diesem Kapitel werden u.a. folgende Fragen beantwortet:

Ziel dieses Kapitels

- Welche Ansätze und Protokolle für die Unterstützung der Mobilität in IP-Netzen gibt es?
- Wie kann Roaming zwischen Hotspots realisiert werden?
- Welche Idee liegt den Protokollen MIP und MIPv6 zugrunde?
- Welche Probleme müssen gelöst werden, um die Mobilität in IP-Netzen zu ermöglichen?
- Wie verläuft die Kommunikation beim Einsatz von MIP bzw. von MIPv6?
- Wie unterstützt HMIPv6 die Mobilität in WLANs mit mehreren Zellen?

13.1 Ansätze für die Unterstützung der Mobilität

Hotspot als PWLAN

WLANs (*Wireless LANs*) werden in Unternehmen immer häufiger als Erweiterung von kabelgebundenen Netzwerken eingesetzt. Sie werden auch in öffentlichen Bereichen, wie z.B. in Hotels, auf Flughäfen, in Sitzungssälen, installiert. Ein WLAN, das in einem öffentlichen Bereich eingerichtet wird und als Internet-Zubringer für mobile Rechner dient, bezeichnet man als *Hotspot* bzw. als *PWLAN (Public WLAN)*.

Bedeutung von Hotspot-Roaming

Ein Benutzer, der mit seinem tragbaren Rechner unterwegs ist, sollte die Möglichkeit haben, jeden Hotspot so zu nutzen, wie er es in seiner Firma bzw. zu Hause gewohnt ist, ohne immer ein Entgelt für die Hotspot-Nutzung zahlen zu müssen. Das sog. *Hotspot-Roaming* ermöglicht die „Wanderung“ von Benutzern zwischen Hotspots verschiedener Betreiber. Man spricht hierbei auch von *PWLAN-Roaming*.

MIPv4, MIPv6 und HMIPv6

Alle aktuellen Notebook-Modelle verfügen heute über einen WLAN-Adapter. Das Protokoll *Mobile IP (MIP)* ermöglicht die uneingeschränkte Nutzung dieser Adapter und damit die Mobilität in IP-Netzen. Man unterscheidet zwischen dem *MIP für IPv4 (MIPv4)* und dem *MIP für IPv6 (MIPv6)*. Eine Erweiterung des *MIPv6* stellt *Hierarchical Mobile IPv6 (HMIPv6)* dar.

13.1.1 Bedeutung von WLAN- und Hotspot-Roaming

Struktur von Hotspots

Um ein öffentliches Gebiet mit den WLAN-Diensten zu versorgen, können in einem Hotspot mehrere *Access Points* installiert werden. Abbildung 13.1-1 veranschaulicht die allgemeine Struktur von Hotspots.

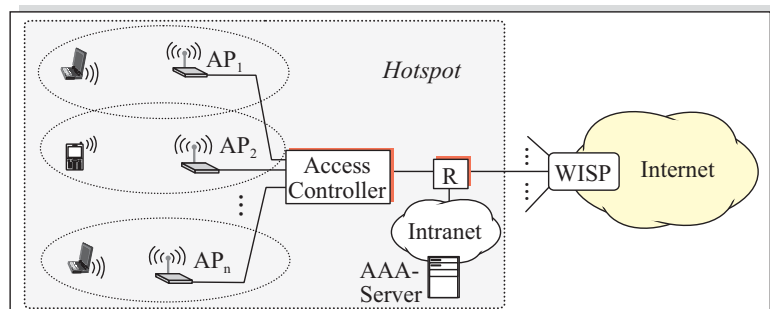


Abb.13.1-1: Allgemeine Struktur von Hotspots
WISP: Wireless Internet Service Provider

WLAN-Roaming

Die Rechner im Übertragungsbereich eines Access Point (AP) bilden keine Funkzelle. Falls ein mobiler Rechner „wandert“, kann er die Funkzelle eines AP

verlassen und in die Funkzelle eines anderen AP hineingehen. Dass ein mobiler Rechner in einem WLAN, in dem mehrere Funkzellen vorhanden sind, von einer Funkzelle zu einer anderen wandern kann, ohne die bestehende Verbindung zu verlieren, wird möglich durch das sog. *WLAN-Roaming*. Hierbei ist zu unterscheiden, ob die beiden APs zu demselben IP-Subnetz oder zu verschiedenen IP-Subnetzen gehören:

- Falls die beiden APs zu dem gleichen IP-Subnetz gehören, handelt es sich um ein *Layer-2-Roaming*.
- Gehören sie zu verschiedenen IP-Subnetzen, handelt es sich um ein *Layer-3-Roaming*.

Um Layer-2-Roaming zu realisieren, müssen sich die Funkzellen benachbarter APs etwas überlagern. Hält sich ein mobiler Rechner noch im diesem Bereich auf, in dem sich die Funkzellen von zwei benachbarten APs überlagern, kann er sich bereits beim neuen AP anmelden, ohne sich vorher beim alten AP abmelden zu müssen. In diesem Fall würde eine aktive Verbindung des Rechners nicht abrechnen. Für die Unterstützung von Layer-2-Roaming wurde das Protokoll IAPP (*Inter Access Point Protocol*) als Standard IEEE 802.11f spezifiziert.

Layer-2-Roaming mit IAPP

Hat der mobile Rechner den AP gewechselt, d.h. es hat auch ein Layer-2-Roaming stattgefunden, und ist er zusätzlich in ein neues IP-Subnetz hineingegangen, muss nun das MIP zum Einsatz kommen. Mithilfe des MIP kann der mobile Rechner während einer bestehenden Session in ein anderes IP-Subnetz aufgenommen werden, ohne die Session abrechnen zu müssen. Das MIP ist somit die Voraussetzung für das Layer-3-Roaming in WLANs.

Layer-3-Roaming mit MIP

Mit der Einführung von Hotspots ist eine neue Art von ISPs entstanden. Man bezeichnet sie als WISPs (*Wireless ISPs*). Ein WISP ist in der Regel ein Unternehmen, das Hotspots bei seinen Kunden (wie z. B. Hotels, Flughäfen) installiert, diesen Hotspots über einheitliche Methoden den Internet-Zugang anbietet und diesen Zugang gemäß einem Vertrag mit dem Kunden abrechnet. Es gibt bereits mehrere WISPs, die ihre Hotspot-Dienste bundesweit anbieten.

WISP

Von großer Bedeutung ist die Möglichkeit, dass ein Benutzer mit seinem tragbaren Rechner von einem Hotspot zu einem anderen „wandern“ und diesen als Gast nutzen kann. In diesem Zusammenhang ist zwischen folgenden zwei Fällen zu unterscheiden:

- Ein Benutzer ist im Hotspot, in dem er sich gerade aufhält, auf Dauer bzw. auf eine lange Zeit registriert. In diesem *Heimat-Hotspot* verfügt er über gewisse Zugangsrechte bzw. über einen noch nicht verbrauchten Account. Nur in seinem *Heimat-Hotspot* wird lokal überprüft, ob das Internet für ihn freigeschaltet werden soll.

Benutzer im Heimat-Hotspot

*Benutzer
im Fremd-
Hotspot*

■ Ein Benutzer hält sich gerade nicht in seinem Heimat-Hotspot auf, sondern als „Gast“ in einem *Fremd-Hotspot*. Dort verfügt er über keine bevorzugten Zugangsrechte. Gilt aber eine Vereinbarung zwischen dem WISP des Fremd-Hotspot und dem WISP des Heimat-Hotspot, kann der Internet-Zugang für den Gastbenutzer im Fremd-Hotspot freigeschaltet werden.

*Mobilität
ohne
Roaming*

Schließt ein Benutzer einen Vertrag mit dem WISP, wird er dort registriert und bekommt für ein bestimmtes Entgelt die Berechtigung, die Hotspots des WISP für den Internet-Zugang zu nutzen. Die Benutzerdaten werden in diesem Fall im zentralen AAA-Server (*Authentifizierung, Authorisierung und Abrechnung*) beim WISP abgespeichert, sodass der Benutzer nicht nur in einem Hotspot beheimatet ist, sondern alle Hotspots seines *Heimat-WISP* nutzen kann. Abbildung 13.1-2 bringt dies zum Ausdruck

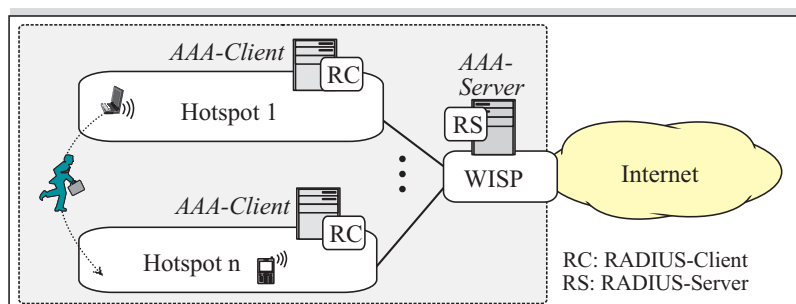


Abb.13.1-2: WISP als Betreiber mehrerer Hotspots

*Bedeutung
des RADIUS*

Hält sich der Benutzer in einem Hotspot seines WISP auf, muss eine Anfrage von dort an den WISP als Zentrale abgeschickt werden, um die Berechtigung des Benutzers zu überprüfen. Hierfür ist ein *AAA-Protokoll* nötig. Als derartiges Protokoll dient das Protokoll RADIUS (*Remote Access Dial-In User Service*), das nach dem Client-Server-Prinzip funktioniert [Abschnitt 12.5]. Wie Abbildung 13.1-2 zeigt, enthalten die einzelnen Hotspots jeweils einen RADIUS-Client als *AAA-Client*. Diese Clients übermitteln die entsprechenden Anfragen an den RADIUS-Server beim WISP, um die Berechtigung von Benutzern zu überprüfen. Der RADIUS-Server beim WISP stellt einen *AAA-Server* dar.

Abbildung 13.1-3 zeigt die Situation, in der ein Benutzer mit seinem tragbaren Rechner unterwegs ist. Hält er sich im Bereich eines Hotspot seines Heimat-WISP auf, wird eine Abfrage zur Überprüfung seiner Berechtigung direkt an den AAA-Server bei seinem Heimat-WISP übermittelt. Weil der Benutzer also weiterhin bei seinem Heimat-WISP ist, findet hier kein Hotspot-Roaming statt.

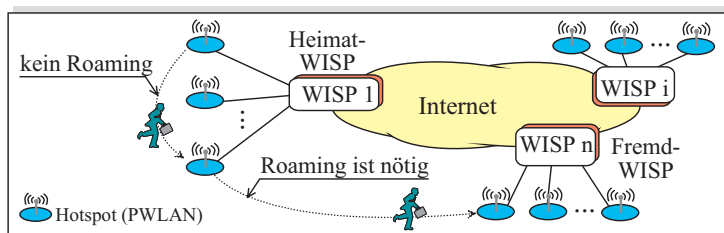


Abb.13.1-3: Notwendigkeit von Hotspot-Roaming (PWLAN-Roaming)

Ist der Benutzer unterwegs und hält sich in einem Hotspot eines Fremd-WISP auf, wünscht er sich natürlich, den Hotspot des Fremd-WISP so nutzen zu dürfen, als ob er in seinem Heimat-WISP wäre. Diesen Wunsch könnte man dadurch erfüllen, dass der Heimat- und der Fremd-WISP eine entsprechende *Roaming-Vereinbarung* unterzeichnen. Damit können die Benutzer, die bei einem von diesen beiden WISPs registriert sind, alle Hotspots von beiden WISPs so nutzen, dass sie keinen Unterschied merken, ob sie in einem Hotspot ihres Heimat-WISP oder des Fremd-WISP sind. Diese Möglichkeit wird in Abschnitt 13.2 ausführlicher dargestellt.

Hotspot-Roaming

13.1.2 Hauptproblem der Mobilität in IP-Netzen

Die Lokation von Rechnern in IP-Netzen wird bestimmt durch ihre IP-Adressen. Ein IP-Netz stellt im Allgemeinen eine Vernetzung mehrerer IP-Subnetze dar, die miteinander mithilfe von Routern verbunden werden. Um die Mobilität in IP-Netzen zu unterstützen, muss man die Lokation eines Rechners feststellen können, also in welchem IP-Subnetz sich ein mobiler Rechner aktuell aufhält.

Ein Rechner am IP-Netz mit einer IP-Adresse gehört immer zu einem IP-Subnetz. Dies bedeutet, dass der Rechner in einem IP-Subnetz *beheimatet* ist. Die IP-Adresse des Rechners in seinem Heimatsubnetz kann somit als *Heimatadresse* interpretiert werden. Bei einem mobilen Rechner muss man damit rechnen, dass er sein *Heimatsubnetz* verlässt und sich vorübergehend in einem *Fremdsnetz* aufhält. Dies führt zu einem *Subnetzwechsel*, der in Abbildung 13.1-4 illustriert wird.

Heimatsubnetz

Hier sendet ein Rechner am Internet ein IP-Paket an den Zielrechner im Subnetz 135.168.34/24. Der Router R_x im Internet leitet dieses IP-Paket anhand der Routing-Tabelle an den Router R_1 weiter. Da der Zielrechner sein Heimatsubnetz verlassen hat, kann das IP-Paket hier den Zielrechner nicht erreichen.

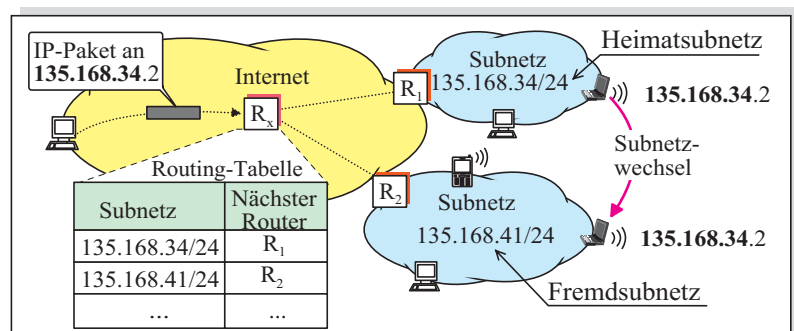


Abb. 13.1-4: Mobilität in IP-Netzen führt zu einem Subnetzwechsel
R: Router

Folge des Subnetzwechsels

Die an einen Rechner adressierten IP-Pakete werden immer in sein Heimatsubnetz übermittelt. Wenn ein mobiler Rechner sein Heimatsubnetz aber verlassen hat, müssen sie in das Fremdsubnetz, in dem der mobile Rechner sich gerade aufhält, weitergeleitet werden. Im Fremdsubnetz muss dem mobilen „Gastrechner“ eine neue vorläufige IP-Adresse zugewiesen werden, um ihn innerhalb dieses Fremdsubnetzes eindeutig lokalisieren zu können. Um einen Gastrechner in einem Fremdsubnetz zu erreichen, muss dieses Fremdsubnetz dem Router R_1 in seinem Heimatsubnetz bekannt sein.

Dies wird mit dem Protokoll *Mobile IP (MIP)* gelöst. Das MIP ermöglicht die Weiterleitung der IP-Pakete zu mobilen Rechnern, die sich in irgendwelchen Fremdsubnetzen aufhalten und ihre Heimatadressen weiter verwenden.

13.1.3 Die grundlegende Idee des Mobile IP

Internet und Postdienst

Das Internet ist kein einzelnes physikalisches Netz, sondern stellt einen Dienst für die Übermittlung von IP-Paketen in einem weltweiten Verbund unterschiedlicher physikalischer Übermittlungsnetze dar. Logisch gesehen stellt das Internet eine Nachbildung des weltweiten Briefpostdienstes dar, wobei ein IP-Paket einem Brief und eine IP-Adresse einer postalischen Adresse entspricht. Der Postdienst basiert auf einer Vernetzung von Postleitzahlgebieten. Das Internet stellt eine Vernetzung von IP-Subnetzen dar. Somit würde ein IP-Subnetz einem Postleitzahlgebiet entsprechen. Beim Postdienst findet eine Unterstützung der Mobilität statt. Sie besteht darin, dass ein Brief nach dem Umzug eines Adressaten an seine neue Adresse nachgeschickt werden kann.

Abbildung 13.1-5 zeigt dieses Prinzip der Nachsendung beim Postdienst.

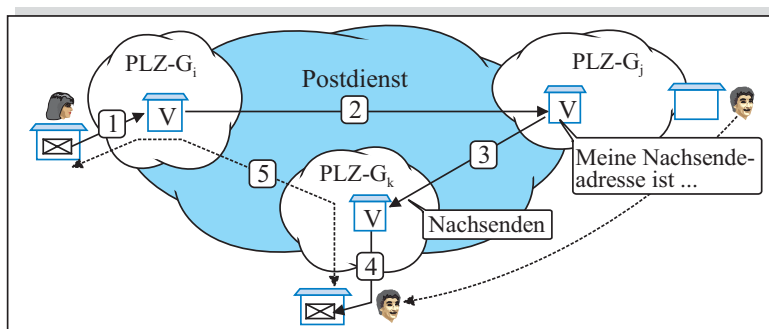


Abb. 13.1-5: Unterstützung der Mobilität beim Postdienst
 PLZ-G: Postleitzahl-Gebiet, V: Verteilungsstelle

Die Unterstützung der Mobilität beim Postdienst lässt sich wie folgt zusammenfassen:

1. Der Brief wird an eine Briefverteilungsstelle übergeben.
2. Der Brief wird von der Briefverteilungsstelle im Postleitzahlgebiet des Absenders an die Briefverteilungsstelle (an das Postamt) des Adressaten übermittelt. Der Adressat hat aber sein Postleitzahlgebiet verlassen und ist unter einer neuen Adresse zu erreichen. Er hat die *Nachsendeadresse* seinem Postamt mitgeteilt.
3. Der Brief wird an die Briefverteilungsstelle des Postleitzahlgebiets aus der Nachsendeadresse transportiert.
4. Der Brief wird an die Nachsendeadresse übergeben.
5. Es kann nun direkter Brieffaustausch zwischen den beiden Personen stattfinden.

*Mobilität
 beim Post-
 dienst*

Die Unterstützung der Mobilität in IP-Netzen basiert auf dem gleichen Prinzip, hier werden ähnliche Schritte unternommen. Es existieren jedoch Unterschiede zwischen dem Mobile IPv4 und dem Mobile IPv6.

*Nachsende-
 prinzip als
 MIP-Idee*

13.1.4 Die Idee des Mobile IPv4

Die IP-Adresse eines Rechners in seinem Heimatsubnetz ist seine *Heimat-IP-Adresse*, sie wird kurz als *HoA (Home Address)* bezeichnet. Wechselt ein mobiler Rechner nun das IP-Subnetz, müssen die an ihn gesandten IP-Pakete in ein *Fremdsbnetz* nachgeschickt werden. Dies entspricht der Nachsendung von Briefen beim Postdienst [Abb. 13.1-5].

Die Art und Weise der Unterstützung der Mobilität in Netzen mit dem IPv4 definiert das *Mobile IP* [RFC 3344/3220]. Man spricht hierbei auch vom *MIPv4 (Mobile IPv4)*. Beim MIPv4 kann man ähnliche Prinzipien erkennen wie beim Postdienst. Abbildung 13.1-6 illustriert das Prinzip der Mobilität nach dem MIPv4. Beim MIPv4 werden zwei Funktionsmodule, die sog. *Mobility Agents*, für die „Betreuung“ von mobilen Rechnern definiert. Ein Mobility Agent kann

*Prinzip des
 MIPv4*

- *Heimatagent (HA, Home Agent)* oder

- *Fremdagent (FA, Foreign Agent)* sein.

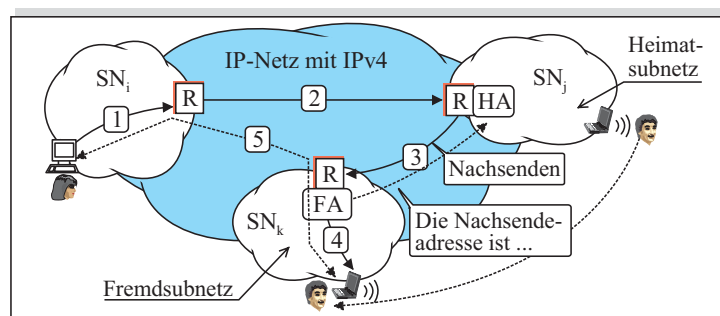


Abb. 13.1-6: Unterstützung der Mobilität in IPv4-Netzen nach MIPv4
FA: Fremdagent, HA: Heimatagent, R: Router, SN: Subnetz

Ein HA wird in der Regel als Funktionsmodul auf einem Router im Heimatsubnetz installiert. Er wird vom mobilen Rechner darüber informiert, in welchem Fremdsubnetz sich dieser gerade aufhält. Der HA leitet die im Heimatsubnetz ankommenden und an den mobilen Rechner adressierten IP-Pakete in das Fremdsubnetz, in dem der mobile Rechner sich aktuell aufhält, weiter. Der Mobility Agent, der für jeden mobilen Rechner, der sich in einem Fremdsubnetz aufhält, zuständig ist, stellt einen FA dar. Ein FA wird wie ein HA in der Regel als Funktionsmodul auf einem Router installiert.

*Mobilität
beim MIPv4*

Wie aus Abbildung 13.1-6 ersichtlich ist, unterscheidet man beim MIPv4 folgende Schritte beim Verlauf der Kommunikation zwischen einem stationären und einem mobilen Rechner:

1. Ein Paket, das an einen Rechner im Subnetz SN_j adressiert ist, wird an den Router im Subnetz SN_j übermittelt.
2. Das IP-Paket wird vom Subnetz SN_i des Quell-Rechners an das Subnetz SN_j des Zielrechners übermittelt. Der mobile Zielrechner hat aber sein Heimatsubnetz verlassen. Daher müssen die an ihn adressierten IP-Pakete in ein Fremdsubnetz, in dem dieser Rechner sich gerade aufhält, weitergeleitet werden. Im Fremdsubnetz wird dem mobilen Gastrechner eine vorläufige IP-Adresse zugewiesen, um ihn innerhalb dieses Fremdsubnetzes zu lokalisieren. Diese Adresse wird als *Care-of Address (CoA)* bezeichnet und sie stellt die *Nachsende-IP-Adresse* dar. Die CoA des Gastrechners ist dem FA bekannt und er übermittelt sie an den HA im Heimatsubnetz. Damit ist die CoA als Nachsendeadresse auch dem HA bekannt.
3. Das IP-Paket wird vom HA an die CoA weitergeleitet. Dieses IP-Paket, das im Header als Ziel-IP-Adresse die HoA des mobilen Rechners enthält, wird hierfür in ein anderes IP-Paket eingekapselt. Im Header des äußeren IP-

Pakets wird die CoA als Ziel-IP-Adresse eingetragen. Man bezeichnet dies als *IP-in-IP-Encapsulation*.

4. Das IP-Paket wird im Fremdsubnetz an den Gast-Rechner übermittelt.
5. Zwischen den beiden Rechnern kann nun eine indirekte Kommunikation – aber nur über den FA – stattfinden. Das ist ein Nachteil des MIPv4 im Vergleich zum MIPv6 [vgl. Abb.13.1-7], bei dem kein FA benötigt wird.

Das MIPv4 wird im Abschnitt 13.3 ausführlicher dargestellt.

13.1.5 Idee des Mobile IPv6

Beim MIPv6 wurden sowohl die Erfahrungen, die bei der Entwicklung des MIPv4 gesammelt wurden, als auch die zusätzlichen Möglichkeiten, die das IPv6 bietet, berücksichtigt. Das MIPv6 bietet neben allen Funktionen des MIPv4 viele Erweiterungen und kann im Gegensatz zum MIPv4 als ein integrierter Bestandteil des IPv6 angesehen werden. Das MIPv6 ist in RFC 3775 spezifiziert.

Die Mobilität in IP-Netzen mit dem IPv6 illustriert Abbildung 13.1-7. Hierbei ist auf den Unterschied zwischen MIPv4 und MIPv6 zu verweisen: Der Gastrechner im Fremdsubnetz SN_k übermittelt seine Nachsendeadresse CoA selbst an seinen HA im Heimatsubnetz, also ohne den FA in Anspruch zu nehmen.

Unterschied zwischen MIPv4 und MIPv6

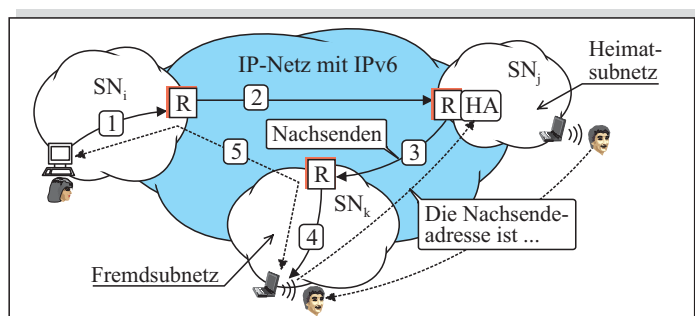


Abb. 13.1-7: Unterstützung der Mobilität in IPv6-Netzen nach dem MIPv6
HA: Heimatagent, R: Router, SN: Subnetz

Beim Verlauf der Kommunikation zwischen einem stationären und einem mobilen Rechner unterscheidet man nach dem MIPv6 folgende Schritte:

Mobilität beim MIPv6

1. Dieser Schritt entspricht dem Schritt 1 beim MIPv4 [Abb. 13.1-6].
2. Dieser Schritt entspricht dem Schritt 2 beim MIPv4.
3. Das IP-Paket wird vom HA an die CoA weitergeleitet. Dieser Schritt entspricht dem Schritt 3 beim MIPv4.