Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Burkhard Stiller, Pascal Kurtansky (Edt.)

PPS-Seminar:
Grundlagen der Internet-Technologie 4

TIK-Report
Nr. 128, February 2002

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Burkhard Stiller, Pascal Kurtansky (Edt.):
PPS-Seminar: Grundlagen der Internet-Technologie 4
February 2002
Version 1
TIK-Report Nr. 128

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology (ETH) Zurich

Institut für Technische Informatik und Kommunikationsnetze,
Eidgenössische Technische Hochschule Zürich

Gloriastrasse 35, ETH-Zentrum, CH-8092 Zürich, Switzerland

PPS-Seminar
Grundlagen der Internet-Technologie 4

Einleitung

Diese nun bereits vierte Auflage des PPS-Seminares sprach wiederum Studierende des Departements für Informationstechnologie und Elektrotechnik an, welche die Grundlagen und erste wichtige Begriffe des Internet erlernen möchten.
Das Seminar vermittelte dabei die wesentlichen Grundlagen für die Kommunikationstechnologie des Internet. Dabei wurden u.a. die folgenden Fragen aufgeworfen und Antworten hierzu gegeben: Was ist ein Netzwerk, was bezweckt die Adressierung, wie funktioniert E-Mail, welche Protokolle gibt es im WWW, was ist IP-Telefonie, wie werden drahtlose Web-Zugriffe möglich, was ist Mobile IP, was bedeutet Peer-to-Peer? Das Seminar vertiefte entsprechende Details der Internet-Technologien: Was ist die Internet-Architektur, welche Protokolle gibt es, welche Rolle spielt die nächste Generation der Internet-Protokolle, welche Entwicklungstendenzen zeigen sich? Insbesondere wurden einige Themen behandelt, die mit dem Auftritt des Internet als Daten- und Informationspräsentationsmedium zusammenhängen, u.a. das HTTP-Protokoll, die Beschreibungssprache HTML sowie die Datenstrukturierungssprache XML.

Ablauf

Die Studierenden erarbeiteten wie in den vergangenen Semestern dieses Mal zu 16 vorgegebenen Themen (siehe unten) eigenverantwortliche, schriftliche Zusammenfassungen, die in diesem TIK-Report zusammengestellt sind. Diese Ausarbeitungen basieren auf teilweise bereitgestelltem Material sowie Literatur, die die Studierenden aus eigenem Antrieb ermittelt und erarbeitet haben. Erstmalig wurde den Studierenden dieses Semester ein zu verwendendes Word-Template abgegeben, um die Form der Ausarbeitungen optisch ansprechend zu gestalten und zu vereinheitlichen. Neben dieser schriftlichen Arbeit, musste jeder Studierende einen Vortrag von genau 15 Minuten halten – mit einer maximalen Überzeit von zwei Minuten. Die Studierenden sollten lernen, in dieser Zeit den technischen Sachverhalt zusammenzufassen, das Essentielle herauszuschälen und anschliessend prägnant zu präsentieren. Ein nachfolgende kurze Diskussions- und Fragephase erlaubte das interaktive Behandeln von Unklarheiten, offenen Fragen sowie die Verknüpfung von den verschiedenen Themen.

Vorträge, Referenten und Titel

Vortrag	1	Zoltan Schlegel	Grundlagen des Internet
Vortrag	2	David Grünert	Netzwerktechnologien für das Internet
Vortrag	3	Christian Marggi	IP, Adressierung und Routing im Internet
Vortrag	4	Kaspar Giger	IPng – Die nächste Generation des Internet Protokolls
Vortrag	5	Andreas Grüter	MobileIP
Vortrag	6	Cyril Stutz	TCP/UDP
Vortrag	7	Stephan Rüegger	Das HTTP-Protokoll
Vortrag	8	Carlo Mathys	Die Beschreibungssprache HTML
Vortrag	9	Marius Staub	Die Datenstrukturierungssprache XML
Vortrag	10	Reto Felix	IP-Telefonie, VoIP
Vortrag	11	Robin Elsasser	Sichere Kommunikation – SSL, SHTTP
Vortrag	12	Luca Zimmermann	Elektronische Post im Internet
Vortrag	13	Tobias Rein	Drahtlose Kommunikation – WAP, WML
Vortrag	14	Michael Reiterer	Übertragungstechnologien
Vortrag	16	Valentin Schuler	Peer-to-Peer Netzwerke

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Grundlagen des Internet

Zoltan Schlegel
szoltan@ee.ethz.ch
07.November 2001

1 Der Begriff des Internet

Das Internet kann man auf verschiedene Weise betrachten und beschreiben: als einen unendlichen, virtuellen Raum, in welchem man alle möglichen Menschen "trifft" und alle möglichen Informationen findet, den sogenannten Cyberspace. Für einige Leute ist es ein neues Medium, welches die Kommunikation der Menschen untereinander revolutionieren wird. Man kann es etwas nüchterner als eine Konvention bezeichnen, als eine Abmachung darüber, wie verschiedene Computer miteinander kommunizieren.

1.1 Die technische Definition

Technisch betrachtet ist das Internet die Verbindung von Millionen von Computern beziehungsweise von abertausenden von Computernetzwerken. Durch diese Vernetzung ist es möglich, dass man von jedem angeschlossenen Computer jederzeit auf unzählige andere zugreifen und von dort die gespeicherten Texte, Töne, Bilder, Grafiken, Videofilme, Programme, oder was eben dort vorhanden ist, auf seinen eigenen Computer holen kann.

1.1.1 Der Computer im Internet und seine Funktion

Es gibt grundsätzlich zwei Sorten von Computern im Internet: Der erste Typ, der sogenannte Server, hält Informationen irgendeiner Art für den zweiten Typ, den Client oder Kunden, bereit. Dieser Server, der „Computerdiener“, lagert zum Beispiel die E-mails für den Benutzer in einem Postfach, bis dieser sie auf seinen Computer holt.

1.1.2 Das Angebot im Internet

- **E-mail**
  Dies ist ein Postservice, der an und für sich gleich funktioniert wie die Papierpost. Nachrichten werden übermittelt. Doch im Internet ist sind Form der Nachricht elektrische Impulse, die über ein Kabelwerk gesendet werden.
- **Online-Kommunikation**
  Der Dienst IRC (Internet Relay Chat), bietet die Möglichkeit direkt mit einer oder mehrerer Personen, egal wo sie sich auf der Welt befinden, zu kommunizieren. Die getippte Nachricht ist praktisch synchron auf dem Bildschirm des Adressaten zu sehen. Man schreibt sozusagen auf dessen Monitor.
- **Diskussionsforen**
  Usenet ist ein weltweites Netzwerk von Computern, die untereinander Artikel austauschen. Es gibt zu allen erdenklichen Themen sogenannte News-Gruppen. Die Teilnehmer können verfasste Berichte an einen zentralen Computer senden. Jeder, der sich für das Thema interessiert, kann die Artikel lesen und darauf eine Antwort schreiben. Entstanden ist diese Form von Kommunikation aus der Wissenschaft Dort hat man versucht, neue Erkenntnisse anderen so schnell wie möglich zugänglich zu machen.

- **WWW**
  Das sogenannte World-Wide-Web ist der meistgenutzte Dienst im Internet. Dort bietet sich die Möglichkeit nicht nur Texte zu lesen, sondern auch Bilder, ja sogar Videosequenzen zu sehen und gesprochene Sprache oder Musikstücke hören zu können. Das WWW ist der Grund, weshalb das Internet so populär geworden ist.
- **FTP**

  FTP bedeutet file-transfer-protokoll. Es ist ein Dienst, der es erlaubt Dokumente sicher über das Internet zu transportieren.
- **telnet**

  Dieser Dienst erlaubt es einem Benutzer sich von einer Maschine in eine andere einloggen zu können. Er „telefoniert“ sozusagen mit der entfernten Maschine. Das Problem dabei ist allerdings, dass der Datenverkehr im Sinne der Abhörbarkeit nicht sicher ist. Deshalb gibt es einen weiteren Dienst:
- **SSH**

  Die secure shell. Dieser Dienst erlaubt es, dass die übertragenen Daten verschlüsselt werden, und somit von dritten nicht gelesen werden können.

1.2 Kurze Entstehungsgeschichte des Internet

1969 startete das amerikanische Verteidigungsministerium das Projekt ARPA-net. Die strategischen Rechner sollten an verschiedenen Standorten installiert und miteinander vernetzt werden, damit nicht die ganze Computerkapazität mit einem Atomschlag ausgeschaltet werden könnte. Der Vorfahre des Internet war sehr klein. Drei Computer in Kalifornien wurden mit einem vierten in Utah verbunden. Da das Arphanet ganz gut funktionierte, wollten sich bald andere Institutionen daran beteiligen. Besonders die Universitäten waren daran sehr interessiert, den wissenschaftlichen Kontakt untereinander zu beschleunigen und zu vereinfachen. Sie erkannten, dass das Arpanet dafür die ideale Voraussetzung bot. So schlossen sie ihre Computer an das Netz an. Dieses wuchs so stark an, dass man das militärische vom zivilen trennen musste. Was blieb ist die gemeinsame Sprache, das Internet-Protocol (IP). Die Benutzer von damals hatten nur ein Terminal, eine Eingabestation zur Verfügung. Die Rechenarbeit wurde von den Computern im Internet ausgeführt. Dies darum, weil es zu dieser Zeit noch keine Personal-Computer gab. Bis zum Anfang der Neunziger-Jahre war das Internet weitgehend der wissenschaftlichen Arbeit vorbehalten, doch dann entstanden auch immer mehr kommerzielle Netzwerke, die den Zugang für jedermann öffneten.

1.3 Der Weg der Information zum Ziel

Als das Arpanet entwickelt wurde, war es das primäre Ziel, ein Netzwerk zu bauen, das auch unter widrigsten Umständen funktionieren würde. Deshalb baute man nicht eine feste Verbindung zwischen zwei Computern auf, sondern entwickelte ein Verfahren, bei dem die Daten sich selber einen Weg zum Ziel suchen. Sie finden ihn mit der Hilfe von sogenannten Routern . Das sind Wegweiser-Computer, welche ein Datenpaket empfangen und dank seiner Codierung (Architektur, Schichtung) wissen, wohin sie es weiterleiten sollen. Sie wissen, wo der nächste Router ist, der die Daten wiederum weiterleiten kann Welchen Weg die Datenpakete genommen haben ist zufällig. Teilweise nehmen die einzelnen Datenpakete einer einzigen Datei sogar verschiedene Wege. Am Ziel werden sie dann wieder zusammengefügt. Das ist die Aufgabe des TCP (Transmission-Control-Protocol).

1.3.1 Schichten und Hierarchie

Wie bereits oben erwähnt, müssen den zu übermittelnden Daten Transportinformationen beigefügt werden, damit ein Datenpaket eindeutig identifizierbar ist und seinen Weg zu Ziel findet.

1.3.2 Die Schichten in einer Dokumentübertragung

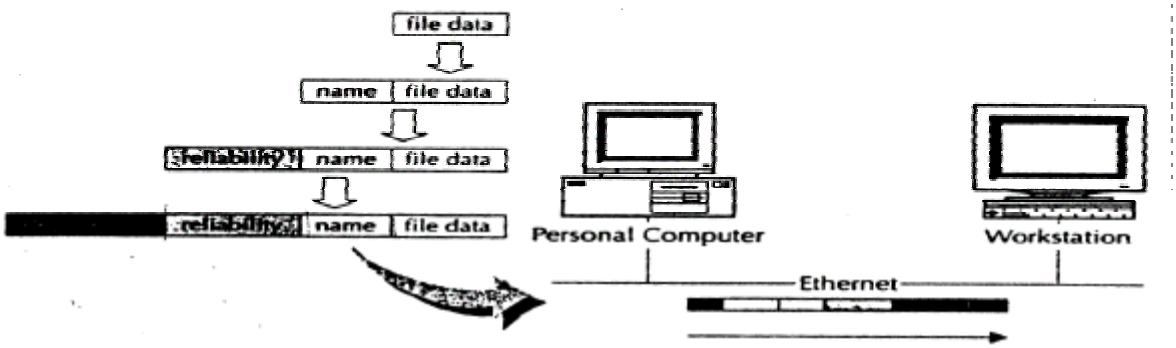


Fig. 1 Die Schichtung der Dokumentenübertragung

Der Inhalt (Schicht 1) des zu versendende Dokument muss mit einem Namen (Schicht 2) versehen werden. Um nun den Transport zu ermöglichen und Uebermittlungsfehler zu vermeiden ist die sogenannte „reliability"-Schicht (Schicht 3) notwendig. Nun muss das Netzwerk das Dokument zum richtigen Ziel befördern. Aus diesem Grund wird die Schicht 4 (forwarding, routing) angefügt. Beachte die Färbung der einzelnen Schichten. Das so verpackte Dokument wird nun wie folgt übermittelt. Zuerst kommt die allgemeinste Schicht, die die Adressierungsinformation enthält (Schicht 4), gefolgt von Schicht 3 und 2 und erst am Schluss die Daten. Eigentlich gleich wie bei der Paketpost: Zuerst wird das Verpackungsmaterial mit der Adresse des Empfängers abgearbeitet, auf dem Weg versucht man die Beschädigung des Paketes zu vermeiden, und zum Schluss gelangt die Ware in die Hände des Empfängers, der den Inhalt verwendet.

1.3.3 Hierarchien

Auch diese sind ein abstraktes Konzept, um Netzwerke zu organisieren. Hierarchien organisieren die Information und delegieren die Verantwortung. Ein Beispiel ist eine Telefonverbindung. Die zu wählende Nummer setzt sich hierarchisch zusammen: Landesvorwahl - Regionalvorwahl - charakteristische Zahl für die Gegend - die Nummer des Haushaltes. (z.B. 0043 - 1 - 643 - 3849) .

1.4 Die Schichten und die Hierarchie im Internet

1.4.1 Das Kommunikationsprotocol TCP/IP

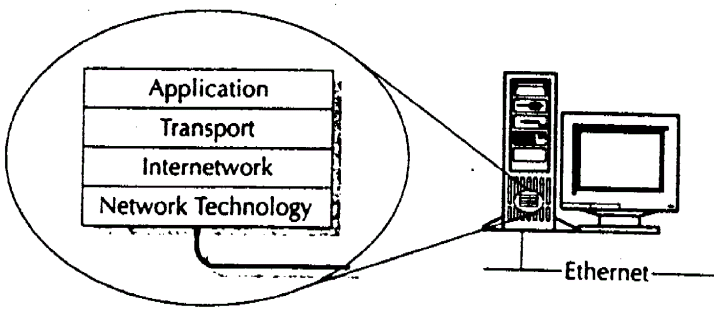


Fig. 2 Der Aufbau, oder die Architektur des TCP/IP

Das TCP/IP Protocol ist wie ein Stack, der TCP/IP-Stack organisiert. Die höchste oder auch die innerste Schicht enthält die Applikationen wie das file-transfer, remote terminal emulation, E-mail, News und die Uhrzeit. Unter dieser folgt die Transport-Schicht. Diese ist dafür zuständig, dass die Applikationen zur gewünschten Adresse gesendet werden. Das TCP ist ein solches Uebertragungs-protokoll. Im Internet arbeitet das Transportprotokoll mit dem Internet-Protokoll (IP) zusammen. IP ist ein verbindungsunabhängiges, optimiertes Protokoll zur Paketübermittlung, welches das Routen, Zerlegen und Zusammensetzen der Pakete übernimmt. Das IP stellt sicher, dass die Information sicher und zielsicher durch das Internet befördert wird. Damit dies möglich wird, ist das IP über den Aufbau, die Topologie des Internet unterrichtet. Die unterste Schicht ist dann die Technologie des Netzwerkes selbst, wo die miteinander verbundenen Systeme Daten austauschen. Das TCP/IP unterstützt diverse Netzwerktechnologien.

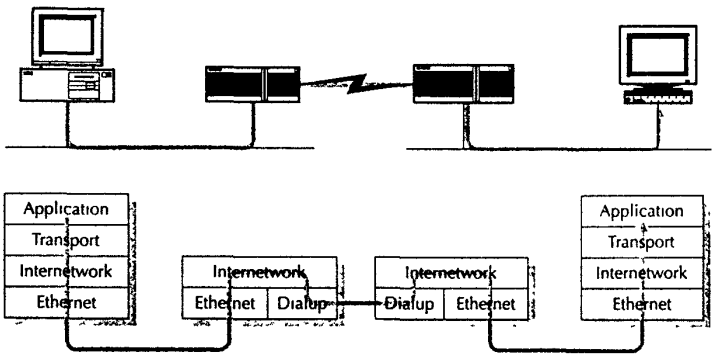


Fig. 3 Ein kleines Netzwerk am Internet

Grundsätzlich werden drei Arten von Uebertragungsdienste (Connetion delivery) unterschieden. Die erste ist die sogenannte Connectionless Delivery, die zweite ist die Connection-Oriented Delivery und die letzte die Kombination der ersten beiden.

1.4.2 Connectionless Delivery (C)

Diese Art Daten zu übermitteln ist die simpelste. Denn ein Protokoll diese Dienstleistung zur Verfügung stellt, braucht es weder Interaktion vor Annahme einer Nachricht, noch stellt es irgendwelche Zusatzinformationen betreffend der Nachricht zu Verfügung. Jede Nachricht wird völlig isoliert bearbeitet. Als Vergleich kann man sich die Briefpost denken. Die Nachricht wird isoliert in einem Umschlag (frankiert!) von der Poststelle angenommen, ohne dass diese weitere Auskünfte über den Inhalt des Briefes braucht. Die Nachricht wird anstandslos dem Empfänger zugestellt. Die Poststelle kümmert sich aber auch nicht darum, was der Empfänger mit dem Brief tut. Ob er ihn beantwortet oder nicht ist ihr egal. Die Poststelle liefert und Schluss. Anders sieht es beider folgenden Uebertragungsart aus:

1.4.3 Connection-Oriented Delivery (CO)

Bei dieser Art wird Zusatzinformation zur Nachricht beigefügt wie zum Beispiel eine Liefergarantie. Dazu muss das Protokoll mit dem Empfänger Kontakt aufnehmen, bevor es die Nachricht liefert und danach, um die Bestätigung für die fehlerfreie Uebermittlung zu erhalten. Als Vergleich kann man sich die Faxübertragung denken. Der Sendefax wählt die Nummer des Empfängers und wartet auf das O.K. desselben. Ist der Empfänger bereit, so werden die Daten überliefert. Danach erkundigt sich der Sendefax beim andern, ob die Uebertragung einwandfrei funktioniert hat. Ist dies der fall, so gibt der Sendefax einen Sendebeleg aus.

1.4.4 Beide Deliveries kombiniert

Die Verbindung beider Uebertragungsarten ist deshalb überhaupt möglich, weil es verschiedene, voneinander unabhängige Schichten im Netzwerk gibt. Die eine Schicht arbeitet Connectionless und eine andere Connection-Oriented.
Betrachten wir ein ATM-Netzwerk (Asynchronous Transfer Mode).

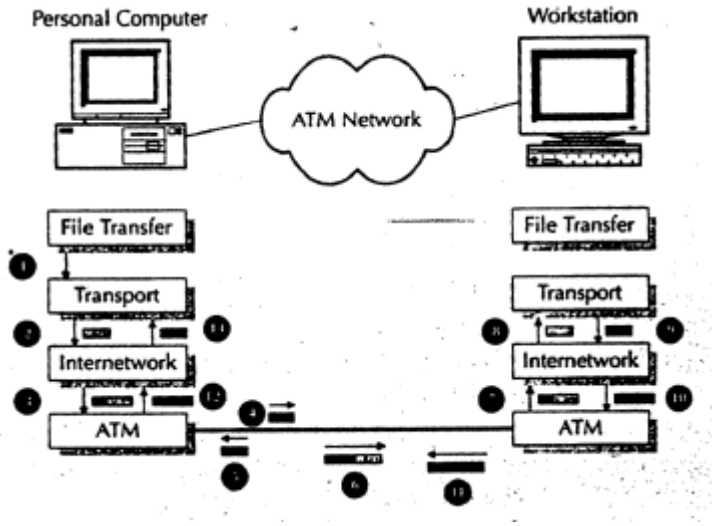


Fig. 4 Datentransport

Betrachte die Nummern zwischen den einzelnen Schichten.
1) Zuerst bittet das Applikationsprogramm, das TCP als Transportprotokoll benützt (CO), die Transportschicht, Kontakt mit der Workstation aufzunehmen, was erstere auch macht.
2) TCP baut die Nachricht und gibt sie der IP-Schicht (C) weiter. Es werden eigene Informationen der Nachricht hinzugefügt und das Ganze der ATM-Software weitergegeben.
3) Die IP-Schicht arbeitet (C). Sie gibt eigene Informationen zur Nachricht und gibt sie der ATM-Software weiter. Das ATM-Netzwerk arbeitet (CO) und baut darum eine eigene Verbindung zur Workstation auf.
4) Die Verbindung wird aufgebaut.
5) Die ATM Empfangsstation bestätigt den Verbindungsaufbau.
6) ATM fügt eigene Information der Nachricht zu und sendet diese.
7) - 13) Die Nachricht wird, diesmal rückwärts im Stack hochbefördert und landet schlussendlich in der Workstation.

1.5 Schlussbemerkung

Das Internet ist ein schnell wachsendes Kommunikationssystem, das es erlaubt, Daten irgendeiner Form schnell und zuverlässig zu übermitteln. Die lästige Papierpost entfällt. Allerdings ist zu beachten, dass das Internet auch Gefahren mit sich bringt. Nämlich können schlechte Programme oder sogar zerstörerische Programme schnell verteilt werden. Dies geschieht meistens über ein Email-Programm.
Hat sich jemand in das Internet eingewählt, kann ein anderer Benutzer in dessen Computer über das Netz eindringen und die Daten einsehen. Um dies zu verhindern werden Abwehrprogramme entworfen. Aber damit hat man ein richtiges Wettrüsten provoziert.
Das Internet ist eine ideale Plattform für jedermann, ob seine Absichten nun gut oder schlecht sind.

1.6 Literaturverzeichnis

S. Thomas: Ing and the TCP/IP Protocols; John Wiley & Sons, Inc., New York, USA, 1996
D. Borchers, M. Benning, J. Kuri: “Hätt ich dich heut erwartet…”; c’t, Heft 21, 1999
E. Wilde: World Wide Web – Technische Grundlagen; Springer Verlag, Berlin, Deutschland, 1999
Andrew S. Tanenbaum: Computernetzwerke, Prentice Hall, Toronto, New York, Sydney, 1998
http://www.ask.uni-karlsruhe.de/doc/talente/40117/HISTORY.HTM, 15.4.2001
http://www.hagen-roewer.de/internet/grundlagen-internet, 15.4.2001
http://www.igd.fhg.de/~jasnoch/fh-vorlesung/inhalt.htm, 15.4.2001
http://www.ikr.tuwien.ac.at/lehre/edv_rpl_2/internet_prinzip.htm, 15.4.2001
http://www.informatik.uni-osnabrueck.de/axel/talks/inet_jur/0/2.html, 15.4.2001
http://www.learnthenet.com/german/section/intbas.html, 15.4.2001
http://www.lrz-muenchen.de/services/schulung/unterlagen/grundlagen, 15.4.2001
http://www.uni-tuebingen.de/zdv/Termine/kursbeschreibung/kurs-internet.html, 15.4.2001

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Netzwerktechnologien für das Internet

David Grünert
davidgr@ee.ethz.ch
23. November 2001

1 Einleitung

Unter einem Netzwerk versteht man mehrere miteinander verbundene Computer, mit dem Ziel Ressourcen gemeinsam zu nutzen oder zu kommunizieren. Es handelt sich also um einen recht weit gefassten Begriff denn sowohl zwei verbundene PCs, wie auch das Internet als Ganzes ist ein Netzwerk.

1.1 WAN und LAN

Bei diesen unterschiedlichen Bedeutungen macht es Sinn den Begriff Netzwerk weiter aufzuspalten. Vielleicht am naheliegensten ist eine Unterteilung nach der Grössenordnung. Als LAN (Local Area Network) bezeichnet man Netzwerke mit einer Ausdehnung bis einige Kilometer. Grössere Netzwerke werden als WAN bezeichnet (Wide Area Network).

1.2 Anforderungen des Internets

Wie sehen die Anforderungen aus, die vom Internet an ein Netzwerk gestellt werden? Man könnte meinen nun müsse ein langer Anforderungskatalog folgen, doch dem ist nicht so. Ein solches Netzwerk muss lediglich in der Lage sein, Pakete bestimmter Grösse zu transportieren. Doch was genau sind Pakete?

2 Pakete und Protokolle

Um Daten zu versenden, können diese in Pakete aufgeteilt und einzeln verschickt werden. Wie ein solches Paket aussieht, wird durch das Protokoll festgelegt. So wie bei der Post benötigen Pakete eine Zieladresse und einen Absender. Dazu kommen oft noch einige Angaben über den Inhalt um sicher zu sein, dass auf der Reise nicht Daten abhanden gekommen sind. Es gibt aber auch noch andere Probleme. Durch die sehr offen formulierten Anforderungen ist eine Vielzahl zum Teil recht unterschiedlicher Netzwerke entstanden. Doch nicht alle haben die gleichen Protokolle für die Pakete. Diesem Umstand wird mit der Schichtung begegnet.

2.1 Was heisst Schichtung

Um die Kompatibilität zu gewährleisten, ist die Kommunikation geschichtet aufgebaut. Die Daten durchlaufen die Grafik (Fig. 1) in Pfeilrichtung. In Schicht 4 des Senders befinden sich die Daten, die gesendet werden sollen. Diese werden an Schicht 3 weitergeleitet. Schicht 3 ergänzt die Daten und leitet sie an Schicht 2 weiter usw. Nach der letzten Stufe habe die Daten ein internet verträgliches Paketformat. Auf der Empfängerseite werden die Stufen in umgekehrter Reihenfolge durchlaufen. In Schicht 4 des Empfängers sind die Daten dann wieder in der Ausgangsform. Die Schnittstellen zwischen den Schichten sind festgelegt. So ist es möglich unterschiedliche Hardware zu verbinden. Es müssen lediglich die richtigen Schichten zur Umwandlung der Daten verwendet werden.

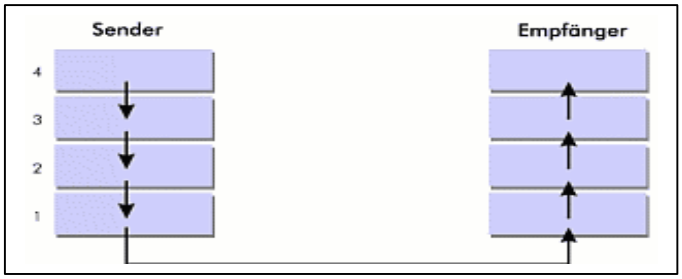


Fig. 1 Prinzip des Schichtenmodells

2.2 Konkretes Beispiel

Die obere Zeile in der Grafik (Fig.2) entspricht dem Ethernet Protokoll. Etwa so wird das Paket über das Netzwerk geschickt. Der Datenteil ist nach einem anderen Protokoll formatiert. In diesem Fall gemäss TCP/IP. Der Datenteil kommt also von einem höheren Schicht und wird entsprechend ergänzt.

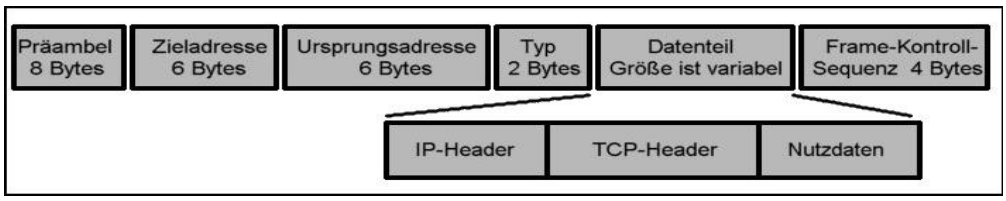


Fig. 2 Paket im Format des Ethernet Protokolls

3 Netzwerktypen

Wie bereits angedeutet, gibt es eine grosse Anzahl von Netzwerktypen. Es stellt sich natürlich die Frage, warum man sich nicht auf einen Standard geeinigt hat. Das Problem sind die sehr unterschiedlichen Anforderungen, welche an ein Netzwerk gestellt werden. Diese machen eine solche Vielfalt sinnvoll. In der Folge werden einige dieser Typen vorstellt.

3.1 Ethernet

Ethernet ist der am weitesten verbreite Netzwerktyp und wird im LAN Bereich eingesetzt. Es existieren verschiedene Versionen.

3.1.1 10Base2

Alle Computer hängen am selben Bus. Er ist mit Endwiderständen terminiert. Geschwindigkeiten bis 10 MBit sind möglich. Das Netzwerk ist anfällig auf Unterbrüche.

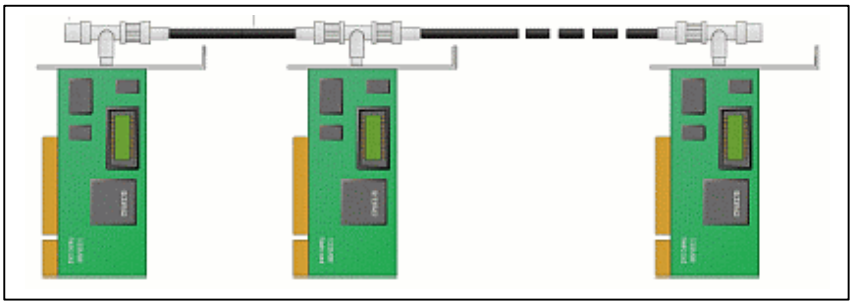


Fig. 3 10Base2 Ethernet

3.1.2 10BaseT

10BaseT funktioniert wie 10Base2, besitzt aber eine sternförmigen Aufbau. Der Bus befindet sich nun im „Kasten“, dem Hub (4.3). Die Zuleitungen wurden einfach verlängert. Geschwindigkeiten bis 10 Mbit sind möglich. Unterbrüche haben kleinere Auswirkungen.

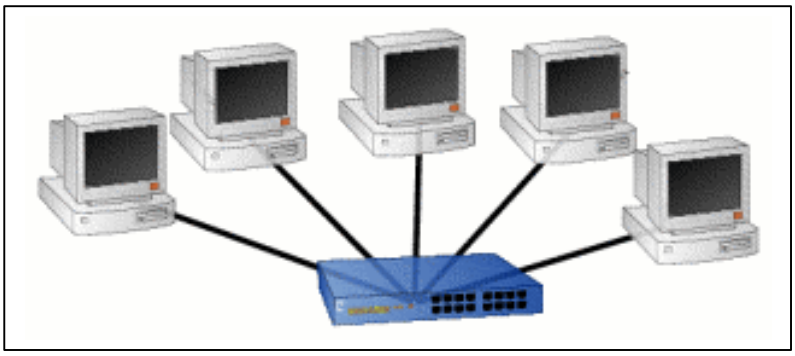


Fig. 4 10BaseT Netzwerk mit sternförmigem Aufbau

3.1.3 Fast Ethernet und Gigabit Ethernet

Beide haben den gleichen Aufbau wie Netzwerke vom Typ 10BaseT. Die höheren Geschwindigkeiten machen bessere Kabel und aufwendigere Hubs nötig. Wie der Name sagt, beträgt die Geschwindigkeit 100 MBit resp. 1GBit.

3.1.4 Funktionsweise

Alle Ethernet Netzwerke funktionieren nach dem gleichen Prinzip. Das Versenden der Daten erfolgt mittels Paket Brodcasting. Dabei werden die Daten auf den gemeinsamen Bus gesendet. Alle am Bus angeschlossenen Stationen können die Pakete sehen, aber nur ein festgeleger Empfänger nimmt sie entgegen. Wie ist es geregelt, dass nicht mehrere Geräte gleichzeitig senden?

- Die Stationen überwachen den Bus ständig. Sie senden erst, wenn der Bus frei ist.
- Beginnen zwei Stationen gleichzeitig mit dem Senden, kommt es zu einer Kollision. Weil die Stationen auch während dem Senden den Bus überwachen, erkennen sie diese Situation. Sie brechen die Übertragung ab und senden ein Kollisionssignal. Die Übertragung wird nach einer zufällig gewählten Zeit wieder aufgenommen. Es ist wichtig, dass der Sender noch während der Übertragung die Kollision erkennt. Ansonsten sind die Pakete bereits angekommen. Dieser Umstand schränkt die Grösse eines Ethernet Netzwerks erheblich ein.

3.2 Token Ring

Das Token Ring Netzwerk ist ringförmig aufgebaut und wird im LAN eingesetzt. Es befindet sich immer nur ein Paket im Ring. Kollisionen sind dadurch ausgeschlossen. Bildlich kann man sich vorstellen, dass im Ring immer ein Zug kreist, das Token. Der Zug ist entweder frei oder besetzt. Kommt der Zug an einer Station vorbei, prüft diese ob der Zug Daten für sie enthält und nimmt diese entgegen. Ist der Zug leer, hat die Station die Möglichkeit Daten dem Zug zu übergeben.

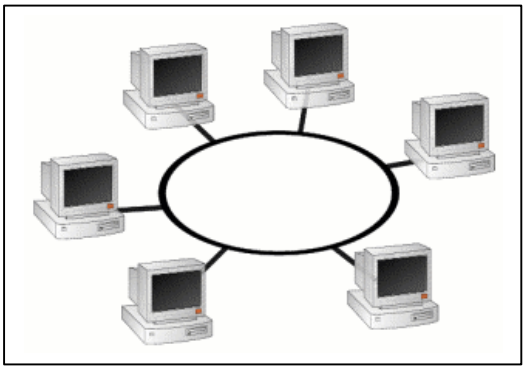


Fig. 5 Token Ring Netzwerk

3.3 ATM

ATM Netzwerke werden im WAN und LAN Bereich eingesetzt. Sie bestehen meist aus einem Netz von Switches (4.4) an welchem die Rechner angeschlossen sind. Zur Verbindung werden Glasfaserkabel eingesetzt. Der grösste Unterschied zu den zuvor genannten Netzwerktypen ist, dass ATM leitungsvermittelnd arbeitet. Will eine Station Daten senden, muss eine Verbindung aufgebaut werden. Zuerst wird eine ID festgelegt. Dann erhält jeder Switch auf der Verbindungsstrecke das Ziel für Pakete mit dieser ID. Nach der Übertragung wird die Verbindung wieder aufgehoben. Anders als bei den meisten paketvermittelnden Netzwerken ist eine feste Bandbreite vorhanden, was für die Übertragung von Bild und Ton wesentlich ist.

4 Netzwerkhardware

Die oben beschriebenen Netzwerktypen können nicht beliebig gross werden. Mit jedem weiteren Teilnehmer sinkt die Geschwindigkeit. Es wird spezielle Hardware benötigt, um Netzwerke zu verbinden und die vorhandenen Bandbreiten sinnvoll zu verteilen.

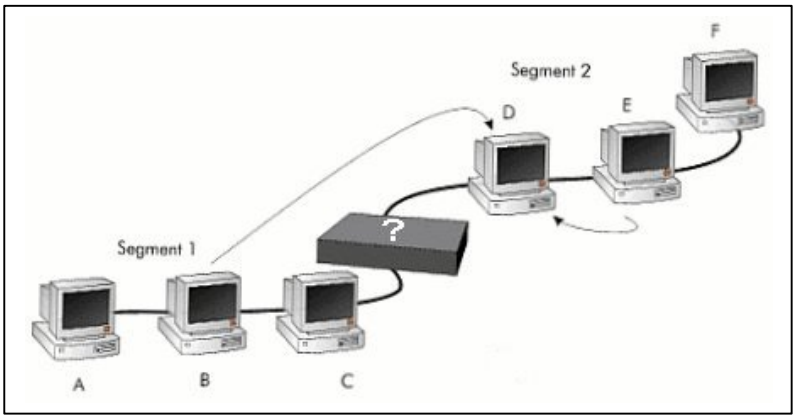


Fig. 6 Anordnung für Repeater und Briges

4.1 Repeater

Repeater werden vor allem in busförmigen Ethernet Netzwerken eingesetzt. Sie haben die Aufgabe ein Signal zu verstärken und zu glätten. Anordnung gemäss Fig.6.

4.2 Bridges

Bridges haben eine Art Schleusenfunktion. Bei einer Anordnung wie sie in Fig.6 dargestellt ist, werden nur jene Pakete weitergeleitet, die sich auch wirklich auf der anderen Seite der Bridge befinden. Dies hat den Vorteil, dass Stationen der Segmente1 und 2 gleichzeitig den Bus verwenden können, wenn sie innerhalb ihres Segmentes bleiben. Doch wie weis die Bridge wie das Netzwerk aufgebaut ist? Es gibt zwei Bridge Techniken:

- **Transparent Bridging:** Die Bridge unterhält eine Tabelle, welche jeder Adresse einen Ausgang zuordnet. Sobald die Bridge ein Paket empfängt, kennt sie die Seite des Absenders. Wenn zu einer Adresse noch keine Zuordnung existiert, leitet die Bridge die Nachricht weiter.
- **Source Route Bridging:** Die Bridge muss keine Tabelle unterhalten. Der Sender eines Pakets gibt explizit an, über welche Segmente das Paket gesendet werden soll. Dazu muss der Sender den Weg natürlich kennen. Zu diesem Zweck sendet er sogenannte Discovery Pakete aus. Diese werden mittels Brodcasting (3.1.4) an alle Stationen gesendet. Der Zielknoten retourniert das Paket wieder mit Brodcasting. Auf dem Rückweg zeichnen die Pakete den Weg und die Verzögerungen auf. Mit diesen Daten kann der Sender den optimalen Weg berechnen.

4.3 Hubs

Hubs wurden bereits unter 3.1.2 erwähnt. Die Anordnung entspricht Fig.4. Der Hub leitet Pakete vom Eingang an alle Ausgänge weiter.

4.4 Switches

Der Switch ist ein Mix aus Hub und Bridge. Die Anordnung entspricht Fig.4. Wie ein Hub verfügt ein Switch über mehrere Ein- und Ausgänge. Pakete werden aber nur an jenen Ausgang weiter geleite, an welchem sich der Empfänger befindet. Es gibt wieder die Möglichkeiten von Transparent Bridging und Source Route Bridging (4.2).

4.5 Router

Ein Router erfüllt ähnliche Aufgaben wie eine Bridge. Die Anordnung entspricht Fig.6. Anders als bei der Bridge, können sich aber in Segment1 und 2 unterschiedliche Netzwerktypen befinden. Router sind oft Rechner mir zwei Netzwerkkarten.

5 Zusammenfassung

- Im WAN und im LAN kommen unterschiedliche Netzwerktypen zum Einsatz. Es ist sinnvoll mit verschiedenen Netzwerktypen zu arbeiten, weil die Anforderungen an diese sehr unterschiedlich sind.
- Die Anforderungen welche das Internet an die Netzwerke stellt, sind sehr gering.
- Die kompatiblen Protokolle ermöglichen es Daten durch verschiedene Netzwerktypen zu transportieren.
- Dank dem geschichteten Aufbau von Sender und Empfänger können die Daten einfach verpackt und wieder herausgelöst werden.
- Um den Verkehr in einem Netzwerk zu optimieren, sind intelligente „Kreuzungen“ nötig.

Quellenangabe

1. S. Thomas: Ipng and the TCP/IP Protocols; J. Wiley & Sons, Inc.,New York, Seitn 43-76.
2. T. Braun Ipng – Neue Internet Dienste und virtuelle Netze; dpunkt Verlag, Heidelber Deutschland 1999 Seten 5-26.
3. ETH Zürich TIK-Report Nr.113, Juni 2001.
4. ETH Zürich TIK-Report Nr.104 Februar 2001.
5. http://www.bszh.de/projekte/netze/zugriffs/zugriff.htm
6. http://www.physio.mu-luebeck.de/schulung/_private/netzwerk/

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Internet Protokoll, Adressierung und Routing

im Internet

Christian Marggi
marggic@ee.ethz.ch
7. Dezember 2001

1 Einleitung

Will man ein grosses Netzwerk aus Computern erstellen, kann man nicht einfach alle Computer direkt miteinander verbinden. Es ergeben sich dann aber einige Probleme: Wie werden die verschiedenen angeschlossenen Systeme voneinander unterschieden? Auf welche Art sollen die Daten verschickt werden und welcher Weg wird dabei am besten gewählt? Wie sollen sich die vernetzten Computer bei Fehlern verhalten? etc.
Es braucht also etwas, das allen vernetzten Geräten zu Grunde liegt.

Egal ob man WWW-Seiten aufruft, e-Mails versenden will, mit FTP Downloads macht oder mit Telnet auf einem fremden Rechner arbeitet, immer werden die Daten auf die gleiche Weise adressiert und transportiert. Nämlich mit TCP/IP. TCP bedeutet **Transfer Control Protocol** (Protokoll für Übertragungskontrolle) und IP bedeutet **Internet Protocol**.

Ich bitte zu beachten, dass in dieser Arbeit oft das Internet stellvertretend für jedes Netzwerk genannt wird, das mit TCP/IP arbeitet.

2 Internet Protokoll – (IPv4)

2.1 Einleitung

Das Internet Protokoll ist die grundlegende Voraussetzung für einen Datentransfer in Netzwerken. Es ist arbeitet verbindungslos, das heisst es stellt die Informationen aufs Netzt, auch wenn es aus irgend einem Grund keinen Empfänger gibt. Das IP hat unter anderem folgende Aufgaben zu bewältigen:

- Adressierung
- Fragmentieren und Fehlererkennung
- Wegwahl und Reassemblierung
- Lebenszeitkontrolle der Datenpakete (ttl – time to live)

2.2 IP-Adressierung

Immer wenn man sich mit dem Internet (oder jedem anderen TCP/IP basierenden Netzwerk) verbindet um zu surfen, Mails zu lesen etc. muss man sich bei einem Provider anmelden, wodurch man (wahrscheinlich ohne es zu merken) eine IP-Adresse zugewiesen bekommt. Der Rechner wird dabei zu einem sogenannten Host. Das ist nötig, weil jedes Gerät in einem Netzwerk eindeutig identifizierbar sein muss, um es ohne Konflikte ansprechen zu können.
Dazu muss man ein Adressierungs-Schema einführen - das sogenannte IP - Adressierungsschema.

Eine typische IP-Adresse im Dezimalsystem sieht etwa so aus: 149.174.211.5 Es sind also immer vier Zahlen getrennt durch Punkte. Die Punkte haben die Aufgabe die über- und untergeordneten Netzte voneinander zu trennen. Das ist ähnlich wie bei internationalen Telefonnummern, wo es Landes-, Ortvorwahlnummer und schliesslich die Teilnehmernummer und eventuell eine Durchwahlnummer gibt.

Jede IP-Adresse wird in zwei Teile getrennt, wobei der erste Teil die Netzwerknummer ist und der zweite Teil die Hostnummer. Der Telefon-Vorwahl entspräche hier die Netzwerknummer und die Durchwahl wäre dann die Hostnummer.
Bei welchem Punkt sich nun die Trennung zwischen Netzwerk- und Hostnummer befindet, bestimmt das folgende Klassifizierungsschema für Netztypen:

Netztyp	IP-Adressierung	Typische IP-Adresse
Klasse A-Netz	**xxx**.xxx.xxx.xxx	**103**.234.123.87
Klasse B-Netz	**xxx.xxx**.xxx.xxx	**151.170**.102.15
Klasse C-Netz	**xxx.xxx.xxx**.xxx	**196.23.155**.113

Tabelle 2.2.1: IP-Adressierungsschema

In der Hierarchiestufe stehen die **Klasse A-Netze** ganz oben. Nur gerade die erste Zahl der IP-Adresse bestimmt die Netzwerknummer. Die Netzwerknummer kann bei Klasse A-Netzen zwischen 1 und 126 liegen. Es gibt also nur gerade 126 Klasse A-Netze im Internet. Man erkennt ein Klasse A Netzwerk also daran, dass die erste Nummer zwischen 1 und 126 liegt. Alle anderen Zahlen dahinter kann der Netzwerkbetreiber frei zur Adressierung von Hostrechnern verwenden und können zwischen 0 und 255 liegen. Auf diese Weise ist es also möglich in einem solchen Netz ungefähr 16.7 Millionen Hosts gleichzeitig teilnehmen zu lassen.
Das lohnt sich nur für ganz grosse Organisationen wie zum Beispiel das Amerikanische Militär oder IBM.

Die **Klasse B-Netze** bilden die zweite Hierarchiestufe. Die Netzwerknummer eines Klasse B-Netzes erstreckt sich über die ersten beiden Nummern und wird daran erkannt, dass die erste Nummer zwischen 128 und 191 liegt. Für die zweite Nummer sind Werte zwischen 0 und 255 erlaubt. Daraus sind rund 16'000 solcher Netze realisierbar mit jeweils bis zu 65'000 Hostrechnern. Klasse B Netze werden von grossen Firmen, Universitäten und Online-Diensten betrieben.

Die **Klasse C-Netze** sind die unterste Stufe, wobei man die ersten 3 Zahlen als Netzwerknummer verwendet. Man erkennt ein Klasse C-Netzwerk ebenfalls an der ersten Zahl, die zwischen 192 und 223 liegen muss. Nur gerade die letzte Zahl kann als Hostnummer verwendet werden. Es können so maximal 256 Hostrechner pro Klasse C-Netzwerk verwaltet werden. Das ist vor allem für kleine Firmen mit direktem Internetzugang und kleinere Internet-Provider interessant.

Es sei der Vollständigkeit halber noch erwähnt, der aufmerksame Leser wird's bereits gemerkt haben, dass es noch ungebrauchte IP-Adressen gibt. Sie heissen **Klasse D- und Klasse E-Netze** gibt. Diese beginnen dann mit den noch nicht aufgeführten Nummern 127, 244 und 255. Diese Netze dienen ganz verschiedenen Anwendungen, wie etwa das Durchführen von Tests. Darauf soll an dieser Stelle jedoch nicht weiter eingegangen werden.

Wenn man mit diesem Schema adressiert ist es möglich über fast 16.8 Mio. Rechner bzw. Hosts zu verwalten. Das ist aber nicht mehr genug! IP Version 6, kurz IPv6 stellt bereits 2^{128} Adressen zur Verfügung. Das wären $4.5*10^{28}$ Adressen pro Mensch auf der Erde oder rund 1500 IPs pro Quadratmeter Erdoberfläche! Mehr dazu im Thema IPng (IP next generation).

2.2.1 Sub-Net Mask

Sub-Netz Masken werden oft in grösseren lokalen Netzwerken benötigt um organisatorische und topologische Probleme zu lösen. Rundsendungen (Broadcasts) beispielsweise müssen von jedem Host eines Netzwerks geprüft werden, ob sie die Sendung etwas angeht oder nicht. So wird vor allem in grossen Netzwerken, wo es viele Broadcasts gibt enorm viel Rechenzeit verschwendet. Darum will man mit Sub-Nets das Netzwerk weiter unterteilen. Die Subnetzmaske ist analog zur IP-Adresse aufgebaut und folgt den gleichen Regeln. Die Sub-Netz Maske wird mit der IP-Adresse durch eine logische UND-Funktion verknüpft und ist nur für Rechner in eben diesem Sub-Netz sichtbar.

2.3 Fragmentierung und Fehlererkennung

Verschiedene Faktoren zwingen uns bei einer Datenübertragung die Daten in Segmente zu unterteilen, weil eine maximale Länge für IP - Datenpakete existiert. Gründe dafür sind Hard- und Softwarebeschränkungen, Beschränkungen auf Grund verwendeter Standards, aber vor allem auch um eine Möglichkeit zu erhalten Fehlerkontrolle durchzuführen. Alle Daten werden wie folgt in Segmente, wir sagen Fragmente, unterteilt:

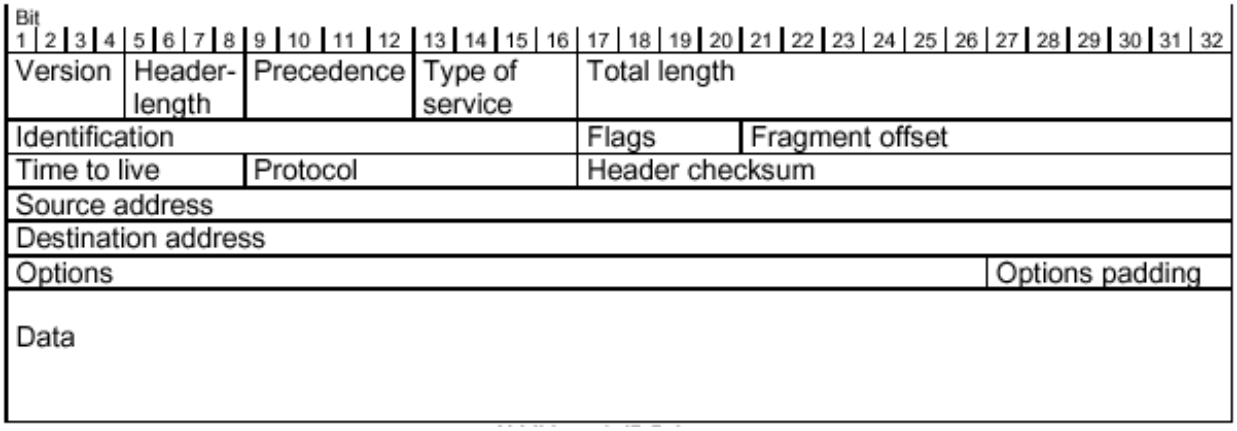


Bild 2.3.1: Aufbau eines Datenpakets

Feld	Aufgabe
Version	Versionsnummer des Protokolls
Header length	Länge des IP-Kopfes in 32-Bit-Worten
precedence	Prioritätsinformationen, Routing-Protocol-Daten
Type of service	Angabe eines Diensttyps (Gesichtspunkte zum Weiterleiten etc.)
Total length	Gesamtlänge in Byte inklusive Kopf und Daten
Identification	Identifikationswert von Paketen
Flags	z.B. Anzeige. ob Fragmentierung zugelassen
Fragment offset	Position im reassemblierten Teil des gesamten Pakets
Time to live (ttl)	Maximaler Hop-Count des Pakets
protocol	Kennung des Schicht-4-Protokolls z.8. 7 für TCP
Header checksum	Prüfsumme über den Kopf eines IP-Pakets
Source adress	Internet-Adresse des Herkunftssystems
Destination address	Internet-Adresse des Zielsystems
Options	Enthält Optionen wie Weginformationen etc.
Options Padding	Füllt die durch die Options angegebenen Daten auf ein Vielfaches von 32 auf
Daten	Beinhaltet die effektiven Nutzdaten

Tabelle 2.3.2: Erklärung zu Bild 2.3.1

In jedem Knoten (vgl. Routing), welche die Datenpakete durchlaufen, muss der Datenkopf neu berechnet werden, weil sich einige Einträge wie zum Beispiel „time to live“ (vgl. time to live) und dadurch die „header checksum“ im Knoten ändern.
Nur mit der Hilfe der header checksum hat man die Möglichkeit einen Hinweis auf einen Übertragungsfehler zu entdecken. So können aber nur allfällige Fehler im Header des Datenpakets gefunden werden. Es gibt auf dieser Protokoll-Stufe keine Möglichkeit das Datenfragment selbst auf seine Richtigkeit zu prüfen!

2.4 Wegwahl und Reassemblierung

Das IP ist ein verbindungsloses Protokoll. Das heisst die verschiedenen Pakete werden nicht zwangsläufig über den gleichen Weg gesendet! Der Vorteil daran ist eine flexible und schnelle Datenübertragung auch Engpässen und Ausfällen gewisser Teilnetze. Der Nachteil ist, dass dadurch nicht alle Pakete wieder in der richtigen Reihenfolge am Zielort eintreffen. An der Zieladresse kann dann mit Hilfe der Headerinformation die Daten wieder richtig zusammengefügt werden. Dieser Vorgang heisst Reassemblierung.

2.5 Lebenszeitkontrolle - time to live

Fehlfunktionen im Netzwerk oder fehlerhafte Domainaufrufe können dazu führen, dass Datenpakete ihren Bestimmungsort nicht erreichen. Diese werden dann von einem Knoten zum anderen immer weitergegeben. Diese „verirrten" Daten belasten dann nur sinnlos das System und man muss einen Weg finden, dass diese Daten nicht ewig weiter kursieren.
Denkbar wäre es jedem Datenpaket eine maximales Lebensalter in Sekunden zuzuweisen. Durchgesetzt hat sich aber der sogenannte Hop-Count, der bei jeder Passierung eines Netzknotens (vgl. Routing) um eins dekrementiert wird. Gelangt dieser Counter bei Null an und hat das Paket seine Zieladresse noch nicht gefunden, wird das Paket automatisch gelöscht. Eine Fehlermeldung wird mittels ICMP (Intenet Contoroll Message Protocol) erzeugt. Der Sender des Pakets wird dadurch veranlasst das Paket erneut auf einem besseren Weg zu verschicken.

3 Domain Name Service - DNS

Ein Computer kann besser mit Zahlen umgehen, ein Mensch hingegen besser mit Namen. Darum hat man mit dem Domain Name Service ein System entwickelt, welches die oben genanten Netz- und Hostnummern in Namensadressen übersetzt, die uns viel vertrauter sind als die IP-Nummern.

Auch dieses System ist hierarchisch aufgebaut. Jede IP-Adresse gehört innerhalb einer **Top-Level-Domain** auch zu einer **Sub-Level-Domain** welche selbst nochmals untergeordnete Domains haben kann. Auch hier werden die verschiedenen Ebenen mit einem Punkt voneinander getrennt. Zum Beispiel: ee.ethz.ch

Die Top-Level Domain steht immer zuhinterst und wäre in diesem Beispiel **ch.** Top-Level Domains sind einigermassen sprechende Abkürzungen. Die Abkürzungen können einen Hinweis auf das Herkunftsland oder den Typ geben. Einige Beispiele:

de: Deutschland
au: Österreich
ch: Schweiz
it: Italien
my: Malaysia
com: Kommerziell orientierter Namensinhaber
org: Organisation
net: Allgemeines Netz
edu: amerikanische Hochschulen
gov: amerikanische Behörden
mil: amerikanische Militäreinrichtungen

Jeder dieser Top-Level-Domain wird von einer eigenen Verwaltungsbehörde beaufsichtigt, welche wiederum für die Namensvergabe der verschiedenen Sub-Domains verantwortlich ist.

Da man nicht direkt mit einer aussenstehenden Adresse Kontakt aufnehmen kann wird man in der Sub-Netz-Hierarchie bis zur Top-Level-Domain durchgeleitet und von dort an die erwünschte Ziel-Top-Level-Domain weiterverbunden. Dort geht's es dann die Hierarchie bis zur gewünschte Zieladresse wieder runter. Falls die geforderte Zieladresse gar nicht existieren sollte wird die Anfrage nach Ablauf des Zeitlimits gelöscht, damit sie nicht beliebig lange im Netz „herumirrt". Dem Benutzer wird eine Fehlermeldung angezeigt.

4 TCP – Transfer Control Protocol

Wir haben gesehen, das es nötig ist die Daten in kleine Teile zu segmentieren - man sagt fragmentieren. Wir haben auch gesehen, dass diese Fragmente verschiedene Wege nehmen können und dadurch auch nicht unbedingt in der richtigen Reihenfolge beim Empfänger eintreffen. Deshalb braucht man ein Protokoll, welches solche Fehler erkennt und in der richtigen Reihenfolge wieder zusammenfügt oder im schlimmsten Fall ein fehlerhaftes oder verlorengegangenes Datenpäckchen erneut anfordert. Dafür ist das auf dem Internet Protocol (IP) aufbauende Transfer-Controll-Protocol (TCP) zuständig. Gerade weil diese zwei Protokolle so eng zusammen arbeiten werden sie oft in einem Atemzug genannt: TCP/IP.

Damit das TCP seine Aufgabe erfüllen kann muss zwischen Empfänger (Client) und Sender (Server) eine Verbindung bestehen. Es ist also im Gegensatz zum IP verbindungsorientiert. Die fragmentierten Datenpakete werden mit Sequenznummern versehen und verschickt. Beim Empfänger werden sie auf Fehler untersucht und wieder in der richtigen Reihenfolge zusammengefügt. Danach wird eine Bestätigung an den Server geschickt, dass die Übertragung erfolgreich war. Um die Sendung nicht unnötig zu verlangsamen wird aber nicht jedes einzelne Paket bestätigt, sondern immer nur Pakete in sogenannten Fenstern. Erst bei Erhalt dieser Bestätigung kann der Sender mit dem Verschicken der nächsten Gruppe von Datenpaketen fortfahren.

5 User Datagram Protocol – UDP

Oft ist es gar nicht nötig, dass alle Informationen wirklich fehlerfrei ankommen. Es kann sogar störend wirken wenn man ein Video-Stream ansehen möchte, denn ein Übertragungsfehler würde eine Neuübertragung der fehlerhaften Sequenz verlangen. Das wiederum lässt dann aber das Video ruckeln was sicher unerwünschter ist als ein kurzfristiger Bildfehler.
Darum gibt es das verbindungslose User Datagram Protocol. Es hat keine Fehlerbehebungsmechanismen ordnet die ankommenden Daten auch nicht der Reihe nach. Der UDP Header beinhaltet lediglich Port des Senders und Empfängers, die Länge des Pakets und optional die Prüfsumme.
UDP bietet also so ein sehr einfachen Demultiplex-Dienst an. Der hat aber eben in der Realzeitkommunikation einen grossen Vorteil weil die Daten direkt an den Empfänger ausgeliefert werden. Auch wird so Gruppenkommunikation möglich, denn Multicast Adressen als Empfängeradressen sind erlaubt!

6 Dynamic Host Configuration Protocol - DHCP

Speziell in Klasse A und B Netzten, wo unglaublich viele Hosts angeschlossen sind, wird es enorm aufwendig das Netz einzurichten und zu warten. Mit TCP/IP muss nämlich jeder Host manuell konfiguriert werden. DHCP erlaubt es nun alle TCP/IP Konfigurationsparameter zentral zu verwalten. DHCP ist einerseits ein Protokoll, welches die Übertragung der Konfigurationseinstellungen von einem DHCP-Server zu den Clients steuert. Andererseits ist DHCP auch eine Funktion für die Zuweisung von Hostadressen innerhalb des Netzwerks. DHCP kann IP-Adressen auf drei Arten zuweisen:

- dynamisch
- automatisch
- manuell

Wählt man sich Beispielsweise ins Internet ein, so bekommt (**dynamisch)** man eine beliebige Adresse zugewiesen die gerade frei ist. Die Adresse ist darum bei jedem erneuten Einwählen eine andere. So können mit relativ wenigen Hostadressen (z.B. in einem Klasse C Netzwerk) relativ viele Host an dem Netzwerk Teilnehmen (natürlich nicht alle gleichzeitig). Bei der **automatischen** Zuweisung wird dem Host ebenfalls eine beliebige Adresse zugewiesen, aber dann reserviert. Beim erneuten Verbindungsaufbau zum Netz bleibt die Adresse erhalten. Das macht beispielsweise Sinn für ein Netzwerk einer Firma, die ein Intranet betreibt. Bei **manueller** Zuweisung muss der Netzwerkadministrator selbst bei jedem Host eine IP-Adresse zuweisen und selbst eine Tabelle über die Vernetzung erstellen. Auf diese Weise kann man ein kleines Familiennetzwerk zu Hause einrichten. In diesem Fall reduziert sich das DHCP auf den Transport.

7 Routing und Gateways

Zunächst ist es im Internet nur möglich etwas direkt an eine IP-Adresse zu verschicken, die im selben Sub-Netz vorhanden ist! Will man aber darüber hinaus, so braucht man Rechner die den Verkehr zwischen diesen Subnetzen regeln. Solche Rechner bezeichnet man als Gateways oder Router.

Gateways leiten Daten seiner eigenen Hostrechner an andere Gateways anderer Sub-Netzwerke weiter. Er empfängt aber auch von anderen Gateways Daten, die für einen Hostrechner des Netzwerks hinter ihm bestimmt sind. Ohne Gateways gäbe es kein Internet!

Das Weiterleiten von Daten zwischen Sub-Netzen wird als Routing bezeichnet. Auf jedem Gateway - Rechner findet man in den sogenannten Routing - Tabellen alle möglichen Verbindungen zu anderen Gateways (und deren Netzwerken). Kann auf dem standardmässigen Weg nicht kommuniziert werden, gehört es zu den Aufgaben eines Gateways bzw. Routers alternative Wege zu finden. Aus diesem Grund versucht ein Router ständig neue, verkehrsarme Wege zu entdecken und bemerkt, wenn ein schon bekannter Weg unterbrochen wurde.

Es ist also gut möglich, dass selbst zwei aufeinanderfolgende Datenpäcken der selben Datei während des Datentransfers zwei völlig verschiedene Wege nehmen. Weder der Browser noch der Benutzer merk das aber.

7.1 Routing Protokolle

Die Routingprotokolle beruhen auf folgenden zwei Verfahren:

- **Distanz – Vektor – Routing:**
  Jeder Router besitzt Informationen über seine Entfernung zu jedem anderen Router und sendet seine Routing – Informationen an die benachbarten Router.

- **Link – State – Routing:**
  Der Router besitzt Informationen über jeden Link der Domäne und berechnet daraus die komplette Netztopologie dieser Domäne.

7.1.1 Routing Inforamtion Protocol - RIP

Das RIP gehört zu den Distanzvektorprotokollen und ist weit verbreitet. Jeder Router sendet in zyklischen Zeitabständen seine Routing-Informationen über die angeschlossenen Links und teilt benachbarten Routern mit wie gross sein Distanz (in Anzahl Knoten) zu anderen Netzwerken ist.

7.1.2 Open shortest Path first - OSPF

OSPF ist ein Link-State-Routing-Protokoll. Die Router einer Domänen tauschen die Beschreibungen ihrer direkt angeschlossener Links aus. Mittels dem Hello-Protokoll lernen die Router die Existenz benachbarter und direkt über einen Link erreichbaren Router kennen. Haben sich 2 benachbarte Router erkannt, so tauschen sie die Beschreibungen ihrer Links aus. Jeder Router speichert fortwährend die neuen Beschreibungen der anderen Router in seinen Routing-Tabellen. So kann der Router ausrechnen wie die Struktur der Domäne aussieht.

7.1.3 Border Gateway Protocol - BGP

Innerhalb von Domänen werden Interior Gateway Routing Protokolle wie OSPF und RIP verwendet. Mehrere Routing-Domänen sind durch einen sogenannten Edge-Router verbunden. Wobei diese wiederum auf einer höheren Ebene ein weiteres Protokoll ausführen. Im Internet ist dies momentan das Border-Gateway-Protokoll. Dieses Protokoll hat neben der Aufgabe Wege zwischen Domänen zu finden zusätzlich die Aufgabeadministrative Randbedingungen bei der Wegefindung zu beachten, so müssen z.B. Zugriffsrechte auf einzelne Netze beachtet werden. Um das BGP-Protokoll durchführen zu können, sind die Edge-Router über TCP miteinander verbunden. Wird zwischen zwei Routern eine Verbindung über TCP erstellt, sendet der eine Router ein Open-Nachricht und der zweite bestätigt den Aufbau mit einer Keepalive-Nachricht, falls er in Verbindung treten will. Im folgenden werden Keepalive- sowie Update-Nachrichten periodisch ausgetauscht. Im Gegensatz zu RIP beschreibt BGP nicht nur die Distanzen zwischen Routern, sondern die Routing-Nachrichten enthalten die vollständigen Wege von einem Router zu einem Ziel.

8 Schlusswort

Der Vorteil am IP ist seine Einfachheit, Übersichtlichkeit und seine grosse Verbreitung
Für die Zukunft stellt IPv4 jedoch eindeutig zu wenig IP-Adressen zur Verfügung. Auch die ganze Konstruktion ist zu statisch. IPv6 wird vorübergehend Abhilfe schaffen können.

Es ist klar dass all diese Protokolle nichts nützen, wenn man keine Anwendungen wie FTP, versch. Browser etc. hat die darauf aufbauen. Man muss also kluge Anwendungen entwickeln die das richtige Protokoll für den richtigen Zweck einsetzen. Also je nach Vorhaben mehr oder weniger Fehlerkontrolle (z.B. TCP vs. UDP)

Daten die mit TCP/IP verschickt werden liegen quasi 1:1 vor. Im Prinzip kann sie jeder Abfangen und so an eventuell sensible Informationen kommen. Es existieren zwar Protokolle mit denen Datenverschlüsselung möglich ist, diese sind aber noch nicht wirklich befriedigend.

9 Quellenangabe

- T.Braun: lPng – Neue Internet-Dienste und virtuelle Netze; dpunkt Verlag, Heidelberg, Deutschland, 1999, Seiten 27-44.
- S.Thomas: lPng and the TCP/IP Protocols; John Wiley & Sons, Inc., New York, U.S.A, 1996, - Seiten 27-41.
- M.Zitterbart, T.Braun: Hochleistungskommunikation 2, Oldenburg Verlag München, Deutschland, 1996, Seiten 46-51.
- M.Hein: TCP/IP Internet-Protokolle im professionellen Einsatz, International Thomson Publishing, 1996.
- http://selfhtml.teamone.de/intro/internet/standards.htm
- http://www.dbg.rt.bw.schule.de/lehrer/ritters/info/bagintra/tcpip.htm

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

IPng
Die nächste Generation des IP

Kaspar Giger
gigerk@ee.ethz.ch
im November 2001

1 IPng – die nächste Generation des IP

IP – Internet Protocol – ist das Internet Protokoll schlechthin. Es verbindet einzelne Computer, LANs, Supercomputer und Server miteinander. Gerade weil so viele verschiedene Technologien mit dem IP zusammen kommunizieren, muss es extrem leistungsfähig, skalierbar, aber auch sicher sein. Und besonders in der heutigen Zeit, wo überall ein PC mit Internetanschluss und anspruchsvollem Benutzer wartet, muss das IP auch schnell Daten transportieren können und viel Platz für Erweiterungen bieten.
Im Folgenden wird darüber diskutiert, welches die Probleme des aktuellen und die Vorteile des zukünftigen IPs sind und wie das IPv6 strukturiert ist.

1.1 Wieso ein neues IP?

1.1.1 Geschichtlicher Hintergrund

Seit Beginn der 90-er Jahren ist des Internet extrem schnell zu einem weltweiten Informations- und Unterhaltungsnetz herangewachsen. Heute können sich schon viele Menschen ein Leben ohne Internet gar nicht mehr vorstellen. Aber genau so schnell wie das Internet wuchs, stieg auch die Anzahl der angeschlossenen Endgeräte und Verbindungsknoten. Waren es vor zehn Jahren erst wenige langsamste Rechner die sich ins Abenteuer Internet stürzten, sind es heute bereits schon Handys, Palmtops oder Spielekonsolen.
Die Internetgemeinde erkannte deshalb schon bald, dass irgendwann mal das IP an seine Grenzen stossen würde und schrieb einen Wettbewerb aus, bei dem Vorschläge für die Zukunft des Internet Protocols ausgearbeitet werden sollten. 1992 wurden mehrere Lösungen eingereicht und diskutiert.
1994 wurde schliesslich beschlossen, IPv6 zum neuen Standard zu machen (der Name IPv6 deshalb, weil der Name IPv5 bereits für ein Streaming Protocol vergeben war). Durch die Abwärtskompatibilität zu IPv4, konnte der Umstieg seither langsam vorbereitet und vollzogen werden. Der Endbenutzer wird aber erst in ein paar Jahren definitiv den Umstieg machen. So ist z.B. Windows XP das erste Microsoft-Betriebssystem, das IPv6 zur Kommunikation benützen könnte.

1.1.2 Wo IPv4 an die Grenzen stösst

Dem Internet Protocol liegt die Philosophie zugrunde, dass jedes Gerät, welches ans Internet angeschlossen ist, eine eindeutige, virtuelle Adresse besitzt. Daraus ergibt sich ein Problem des heutigen IPs: Die Adressen (so genannte IP-Adressen) werden mit 32-Bit kodiert. Deshalb können „nur“ 2^{32} (ca. 4.2 Mia.) verschiedene Adressen vergeben werden. Doch glaubt man heutigen Studien, werden es in den nächsten Jahren nicht mehr nur reine PCs sein, die eine solche Adresse brauchen, sondern auch Computermodule in Autos, Küchengeräten, Stereoanlagen etc. Man rechnet mit etwa 100 IP-Adressen pro Person. Man erkennt also schnell einmal, dass die heutigen 32-Bit irgendwann nicht mehr ausreichen werden. Im neuen IPv6 bestehen die Adressen deshalb aus 128-Bit. Daraus ergeben sich 2^{128} (ca. 3.4×10^{38}) verschiedene Kombinationen, theoretisch also 6.7×10^{23} Adressen pro Quadratmeter Erdoberfläche!
Ein weiterer Punkt ist die Rationalität. Das IPv4 ist zu umständlich für eine schnelle Kommunikation. Die Router und Verbindungsknoten müssen immer noch zu viel selbst überlegen und zusätzlich wird unnötiger Ballast mitgeschleppt. Im neuen IPv6 ist dieser kleine Engpass behoben. Zum einen wurden die Header auf ein Minimum reduziert und zum anderen wurde den Routern und Verbindungsknoten ein grosser Teil der Arbeit abgenommen.

Der dritte Kritikpunkt am alten IP waren die fehlenden Sicherheitsfunktionen. Gerade in der Zeit, in der sich Interneteinkäufe und Internetbanking grosser Beliebtheit erfreuen sind solche Mechanismen von zentraler Bedeutung. Deshalb wurde in der Entwicklung auch viel Gewicht auf die Sicherheit gelegt, wie z.B. die Authentifizierung von Servern und die Verschlüsselung von Paketen.
Weiter wurde viel auf die Flexibilität und Skalierbarkeit geachtet. So gibt es heute diverse Erweiterungsheader, die man beiziehen kann, aber nicht muss. Das hat den Vorteil, dass gewisse Funktionen (z.B. Sicherheit, Fragmentierung, etc.) bei Gebrauch implementiert werden können, aber keinen unnötigen Ballast darstellen, falls man sie nicht braucht.

1.1.3 IP-verwandte Protokolle

Im Zuge der Erneuerung des Internet Protokolls, wurden andere Protokolle, die ebenfalls mit dem IP stark verbunden sind, auch erneuert. So stehen uns DNS (Domain Name Server), PPP (Point-to-Point Protocol), DHCP (Dynamic Host Configuration Protocol)… in einer neuen Version zur Verfügung (DNSv6, PPPv6, DHCPv6, …). Auf diese Protokolle soll im Folgenden aber nicht weiter eingegangen werden.

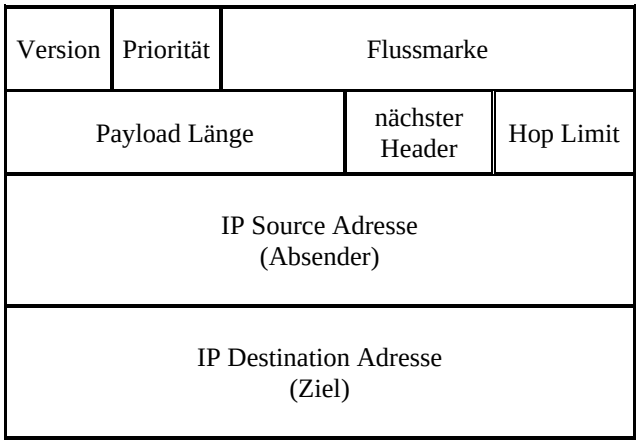
1.2 Aufbau von IPv6

Im Wesentlichen ist IPv6, genau wie sein Vorgänger IPv4, durch einen Basic-Header definiert. Neu gibt es aber noch weitere so genannte Erweiterungs-Header (Extension-Headers), mit denen sich fast beliebig viele weitere Funktionen implementieren lassen.

1.2.1 Basic Header

Der Basic Header, mit einer Bandbreite von 32 Bit besteht aus verschiedenen Variablen, wie unten dargestellt. Kurz zu den einzelnen Optionen und deren Funktionen:
Version, 4 bits:
Im Versionsfeld wird die verwendete Version des IP-Headers angegeben. (Genau gleich wie bei IPv4)

Priority (Priorität), 4 bits:
Definiert den relativen Prioritätswert des jeweiligen Paketes. Je höher dieser Wert, desto schneller wird das Paket verarbeitet und weiterverschickt. Hier einige Beispiele für Prioritätswerte: Hintergrundaktionen (z.B. News)=1, Email=2, Dateitransfer=4. Diese Liste geht offiziell von 0 bis 7. Neuerdings gibt es auch Programme (so z.B. Real-Time Audio), die auch Werte über 7 gebrauchen.
Zu bemerken ist, dass man auch hier die Zukunft und ihre Veränderungen einplante. So gibt es Werte, die für zukünftige Funktionen reserviert sind.



IPv6 Header

Flow Label (Flussmarke), 24 bits:
Definiert die spezielle Behandlung bei der Übermittlung des jeweiligen Pakets. Die Flow Labels sorgen für einen effizienteren Datenfluss. Wenn ein Router ein Datenpaket erhält, muss er zuerst die Adresse des Zielrechners ausfindig machen und so den besten Weg dorthin suchen. Bisher machte das der Router bei jedem Paket von neuem. Mit Flow Labels versucht man das nun zu verhindern. Der Router merkt sich während einer Zeitdauer von sechs Sekunden, wohin er die Pakete mit welchem Flow Label schickte. Kommen nun weitere Daten mit der gleichen Flussmarke kann er auf die im Cache gespeicherte Zieladresse zugreifen und die Pakete auf diese Weise schneller abfertigen.
Flow Labels sind quasi typisch für das Internet Protokoll. Da das IP ein Netzwerkprotokoll ohne permanente Bindung (also nicht wie z.B. ein Telefongespräch) ist, kommt kein ständiger Datenfluss zustande, was zur Folge hat, dass eben für jedes Paket wieder ein Weg zur Zieladresse gesucht werden muss.

Payload-Länge, 16 bits:
Diese Variable gibt die Anzahl Bytes an, die das Paket nach dem IP-Header noch beinhaltet. Sollte einmal das Paket grösser sein, als in diesem Feld angegeben werden kann, gibt es eine zusätzliche Funktion, die dieses Übel beheben kann. In diesem Fall würde die Payload-Länge auf den Wert 0 gesetzt. Dazu ist aber noch anzumerken, dass hardware-technisch im Normalfall keine derart grossen Pakete gesendet werden können.

Next Header, 8 bits:
Definiert den Header, der auf den Basic-Header folgt. Das kann ein Header des nächst unteren Netzwerkprotokolls (z.B. Ethernet) oder auch ein IPv6-Erweiterungsheader sein. Die Angabe des Headers wird mit vordefinierten Zahlenwerten übergeben.
In diesen Erweiterungsheadern liegt eine der Stärken von IPv6. Der Basic-Header kann um beliebig viele weitere Header erweitert werden. Doch weil sie nur optional sind, bleibt das Grundgerüst trotzdem schlank.

Hop Limit, 8 bits:
Diese interessante Eigenschaft definiert die Lebensdauer eines gesendeten Pakets. Der Wert wird zu Beginn vom Sender auf einen beliebigen Startwert gesetzt. Jeder Router, der nun dieses Paket verarbeitet, dekrementiert dieses Limit um eins. Erreicht nun das Hop Limit den Wert Null, wird das Paket zerstört und nicht weiter verarbeitet und gesendet.
Eine Anwendung dafür ist z.B. die Suche innerhalb eines IPv6 Netzwerks. Sucht man den nächstgelegenen Server mit bestimmten Eigenschaften sendet man zuerst Pakete mit dem Hop Limit 1. Kommt keine Antwort zurück werden Pakete mit dem Limit 2 gesendet, usw. bis eine Antwort zurückkommt.
Mit dieser Option wird auch verhindert, dass Pakete, die ihr Ziel nicht finden können, ewig im Internet herum schwirren und so das Internet unnötig belasten und somit bremsen. Weil nun diese Zahl endlich ist (2^8=256) ist es auch nicht möglich, eine derart grosse Zahl hineinzuschreiben, dass die Pakete während mehrerer Tage unterwegs sind, ohne ihr Ziel zu erreichen.

IP-Source Adresse, 128 bits:
Erstmals in 128 kodierte Adresse des Absenders. Die Source-Adresse wird dazu gebraucht, dass der Server auch weiss, von wo die Anfragen kommen, wohin er also die Antwort schicken muss. Zudem können so auch Fehler zurück an den Sender geschickt werden.
Auf die IPv6-Adressen soll aber weiter unten noch eingegangen werden.

IP-Destination Adresse, 128 bits:
Genau gleich wie die Source-Adresse ist auch die Ziel-Adresse durch 128 Bits dargestellt. Die Aufgabe der zwischengeschalteten Router ist es, das gesamte Paket an diese Adresse zu schicken.

1.2.2 IPv6-Adressen

Für Otto-Normalverbraucher wird sich sowieso nach der Umstellung auf IPv6 nicht viel ändern. Das einzige, was ihm, als Hobby-Netzwerkbastler ins Auge sticht, sind die anderen IP-Adressen. Bisher (IPv4) wurden die Adressen mit 32 Bits angegeben und zwar im Format xxx . xxx . xxx . xxx (wobei xxx für eine Zahl, bestehend aus 1 Byte steht, also 0 bis 255, in Dezimalzahlen angegeben). Wie werden die 128 Bits aber jetzt gefüllt? Man unterteilt die Adresse neu in 8 Teile, also vom Format x : x : x : x : x : x : x : x (wobei x für eine Zahl, bestehend aus 2 Bytes, in hexadezimaler Schreibweise, also 0 bis FFFF).
Neu hinzu kommt, dass man zwischen drei Arten von Adressen unterscheidet: Unicast, Anycast und Multicast, welche im folgenden beschrieben werden sollen. Welcher Typ jeweils gebraucht wird, wird in den ersten Bits der Adressen angegeben.
Unicast: Jeder Internetteilnehmer erhält genau eine, eindeutige Adresse. Darin kann z.B. die Netzwerkuntergruppe, die physikalische Adresse (MAC) etc. gespeichert werden.
Anycast: Generell gleicher Aufbau wie bei Unicast. Nur können mehrere Netzwerkteilnehmer die gleiche Adresse erhalten. Sobald also zweimal die gleiche Unicast-Adresse verwendet wurde, wird die Konfiguration automatisch auf Anycast umgestellt. Das Paket wird in diesem Fall an den Teilnehmer geschickt, der sich am nächsten beim Sender befindet.
Multicast: Mit Multicast-Adressen werden Benutzergruppen adressiert. Pakete die an diese Gruppenadresse geschickt werden, erhalten alle anderen Teilnehmer mit der gleichen Multicast-Adresse ebenfalls. Auf diese Art ist es möglich, relativ einfach Video-Konferenzen o.ä. zu veranstalten, ohne aufwändige Software zu schreiben, die alle Pakte an alle Teilnehmer schickt.
Eine zusätzliche Schwierigkeit bei den IPv6-Adressen ist, dass die Internetgemeinde natürlich nicht von heute auf morgen umstellen und ihre IPv4-Adressen durch solche der neueren Version auswechseln kann. Deshalb wurde speziell darauf geachtet, dass es einen Abwärtskompatibilitätsmodus zur Vorgängerversion gibt. In diesem Modus werden die ersten 96 Bits der IPv6-Adresse auf Null gesetzt und in den restlichen 32 Bit die alte Adresse gespeichert.

1.3 QoS – IP und Quality of Service

1.3.1 Was ist Quality of Service?

Beim Datenaustausch in einem Netzwerk werden ständig mit unglaublich grosser Geschwindigkeit Pakete gesendet. Doch wer garantiert mir, dass die Pakete ihr Ziel wirklich erreichen? Es kann sehr gut möglich sein, dass Pakete unterwegs für immer verloren gehen und ich so keine Antwort erhalte. Pakete können z.B. verschwinden, wenn sich in einer Warteschlange eines Routers zu viele Pakete angestaut haben, und er nicht mehr alle abarbeiten kann. In einem solchen Fall werden die ankommenden Päckchen zerstört. Besonders bei wissenschaftlichem und geschäftlichem Gebrauch ist es aber dringend notwendig, dass alle Daten ihr Ziel schnell erreichen.

1.3.2 QoS in IPv4

In der vierten Version des Internet Protocols gibt es eigentlich keine Lösung zum Problem des Quality of Service. Man kann den Paketen nicht eine spezielle „Wichtigkeit“ zuordnen. Eine Art Lösung, unabhängig vom Protokoll wäre, die Bandbreite zu erhöhen. Doch das ist natürlich wieder mit höheren Kosten verbunden. Aber auch hier gilt: Die Bandbreite des gesamten Netzwerks kann nicht grösser sein als die, des langsamsten Teilnehmers (Router, Server, etc.). Deshalb ist die Erhöhung der Bandbreite auch keine wirklich sinnvolle Lösung.

1.3.3 QoS in IPv6

Da man bereits frühzeitig einen Engpass beim QoS erkannte, wurde diese Thematik in die Entwicklung des neuen Internet Protocols einbezogen. Gelöst wurde das Problem mit zwei Einträgen im Header von IPv6: Priorität und Flow Label. Die Flow Label Option sorgt für eine speditive Weiterleitung der Daten und die Prioritäts Option führt dazu, dass wichtigere Pakete schneller verarbeitet werden als weniger wichtige. So ist es z.B. möglich, dass man wichtige Daten mit einem höheren Prioritäts-Wert versieht und somit sicherer sein kann, dass die Daten auch schnell ihre Destination erreichen.

1.4 Sicherheit in IPv6

Sicherheit in einem Netzwerk besteht aus zwei Teilen: Zum einen muss sichergestellt werden, dass die Daten korrekt vom gewollten Server kommen und zum anderen sollte ausser dem Server und dem Client kein anderer Rechner mehr die Daten lesen können.

1.4.1 Authentifizierung

Ein Erweiterungsheader von IPv6 ist der Authentication-Header. In diesem Header wird die Information gesendet, von welchem Server die Pakete kommen. Bevor der Datentransfer beginnt, bestimmen Client und Server einen Schlüssel, den nur sie beide kennen. Dieser Schlüssel wird verschlüsselt und mitgesendet. Der Client entschlüsselt ihn wieder und kann so die Echtheit der Daten feststellen.
Bei der Version 4 gab es eine solche Option noch nicht. Deshalb kam es oft vor, dass Hacker einen anderen Host vortäuschten und so an sichere Daten herankamen.

1.4.2 Verschlüsselung

Bei der Verschlüsselung der Pakete gibt es zwei Varianten. Die eine setzt den Verschlüsselungs-Erweiterungsheader unmittelbar vor die Daten und verschlüsselt danach alles (Transport-Modus). Die andere Variante verschlüsselt das gesamte IP-Paket inklusive Daten und Headers und generiert dazu einen neuen Basic-Header und Verschlüsselungsheader.
Verwendet wird dabei das DES-CBC Verfahren (Data Encryption Standard, Cipher Block Chaining), welches auf einem zufälligen Initialisierungsvektor aufbaut. Die Tatsache, dass dieser Vektor zufällig zustande kommt, macht die Verschlüsselung nochmals sicherer.

1.5 Schlusswort

Man kann doch sagen, dass mit dem IPv6 ein Standard kommt, bei dem wirklich aus den „Fehlern“ der vergangenen Versionen gelernt wurde. Besonders bemerkenswert ist die Skalierbarkeit des Protokolls. Für gewöhnliche Anwendungen ist der Header extrem schlank und beschleunigt so den Datenfluss. Dabei wurde aber nicht vergessen, optionale Sicherheitsfunktionen etc. einzubauen.
Trotz aller Kompaktheit wurde auch noch Platz reserviert für zukünftige Funktionen und Optionen, welche uns heute noch nicht beschäftigen oder wir noch gar nicht kennen (so geschehen beim QoS in IPv4).
Erstmals werden im Internet Protocol Version 6, neben erweiterten Sicherheitsmechanismen, auch Quality of Service Optionen enthalten sein. Doch hier ist noch zu bemerken, dass das zwar ein logischer und vernünftiger Schritt ist, sich diese Funktionen aber noch immer in der Testphase befinden und man noch nicht weiss, wie sie sich im „Alltag“ bewähren werden.

Referenzen

- S. Thomas: IPnG and the TCP/IP Protocols
- Matthias Hein, Michael Reisner: TCP/IP Ge-Packt
- http://www.stardust.com/ipv6
- http://www.geocities.com/SiliconValley/Foothills/7626/defin.html
- http://www.ietf.org/rfc/rfc2405.txt

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Mobile IP

Grüter Andreas
gruetera@ee.ethz.ch
12. Dezember 2001

1 Einführung und Problemstellung

Nachdem sich das Internet rasend schnell verbreitet und weltweit Fuss gefasst hat, möchte man die Vorteile und Annehmlichkeiten, welche das Internet bietet, nicht mehr missen und überall zu jeder Zeit nutzten können. Darum nimmt die Zahl mobiler Endgeräte (z.B. Laptops und Handys mit Internetzugriffsmöglichkeit) stetig zu. Diese mobilen Knoten *(mobile node MN)* im Netzwerk führen im Zusammenhang mit der Funktionsweise der herkömmlichen Internetprotokollen zu Problemen: Ein Rechner mit der IP-Adresse 129.12.31.145 liegt im Subnetz 129.12.31. Ein IP-Paket wird von den Routern nach dem Kriterium des am besten passenden Präfixes weitergeleitet. Wird dieser Rechner nun in ein anderes Subnetz verschoben, kann er keine Pakete mehr empfangen, da das Präfix des neuen Subnetzes nicht mit jenem der IP-Adresse des Rechners übereinstimmt.

1.1 Lösungsmöglichkeiten ohne Mobile IP

Im Folgenden werden zwei Möglichkeiten beschrieben, die dieses Problem lösen, ohne dass ein zusätzliches Internetprotokoll eingeführt wird.

- Ein Vorschlag ist das Ändern der IP-Adresse eines mobilen Endgerätes, bei jedem Wechsel des Subnetzes. Das bedeutet, dass auch alle *Domain Name System (DNS)*-Einträge für das Endsystem aktualisiert werden müssen, damit das Gerät gefunden werden kann.
- Spezifische Wege

Diese Möglichkeiten sind zu langsam respektive zu aufwendig. Also hat die *Internet Engineering Task Force* (IETF) ein neues Internetprotokoll namens Mobile IP entworfen.

1.2 Anforderungen an Mobile IP

Ein mobiler Knoten muss in der Lage sein, mit anderen Knoten zu kommunizieren, auch wenn er seine Lokation im Internet ändert. Dabei soll er seine IP-Adresse behalten können. Das ganze soll so aufgebaut sein, dass Endgeräte nichts von der Mobilität ihres Kommunikationspartners bemerken, wenn sie mit einem mobilen Rechner Daten austauschen. Der Austausch soll wie zwischen festen Knoten erfolgen. Zudem dürfen keine Änderungen an herkömmlichen Endsystemen und Routern nötig sein. Des weitern muss die Sicherheit gewährleistet sein, was Authentifizierungsmechanismen erfordert. Schlussendlich ist es erstrebenswert, ein effizientes Internetprotokoll zu kreieren, wobei möglichst wenig zusätzliche Daten mitgesendet werden müssen. Der Grund liegt auf der Hand: es sollen möglichst viele mobile Endsysteme unterstützt werden können, welche allenfalls nur über eine schmalbandige Anbindung zu erreichen sind.

2 Arbeitsweise

In diesem Kapitel wird erläutert, wie Mobile IP funktioniert. Zuerst werden die Begriffe, die im Zusammenhang mit Mobile IP auftreten erläutert, dann wird der Ablauf beim Transferieren von Daten aufgezeigt.

2.1 Komponenten

Die folgenden Begriffe sind für das Verständnis wesentlich:

- Der mobile Knoten *(mobile node MN)* ist ein Endgerät, das nicht an einen Netzwerkanschluss gebunden ist, sondern seinen Internetzugangspunkt wechseln kann und dabei die permanente IP-Adresse behält.
- Der Heimatagent *(home agent HA)* ist ein Router im Heimatnetz (Subnetz zu welchem ein mobiler Konten gemäss IP-Adresse gehört), welcher weiss, wo sich die mobilen Knoten des Heimatnetzes momentan aufhalten. Er empfängt Daten, die an die mobilen Knoten adressiert sind und leitet sie per Care-of-Adresse weiter.
- Als Fremdagent *(foreign agent FA)* wird ein Router in einem Fremdnetz bezeichnet, der die Pakete an den mobilen Knoten weiterleitet und Pakete vom mobilen Knoten entgegennimmt und versendet. Fremdnetz nennt man das Subnetz in dem sich der mobile Knoten befindet, falls es nicht sein Heimatnetz ist.
- Mit Heimatadresse eines mobilen Knotens meint man die permanente IP-Adresse, welche auch fix bleibt, wenn der Knoten verschiedene Subnetze im Internet durchwandert. Der vordere Teil der Adresse entspricht dem Präfix des Heimatnetzes.
- Die Care-of-Adresse *(care of adress COA)* ist nur eine temporäre Adresse für einen mobilen Knoten. Sie ändert sich, wann immer der mobile Knoten ein anderes Netz besucht. An diese Adresse werden vom Heimatagent aus Pakete, die für den mobilen Knoten bestimmt sind, gesendet. Es gibt zwei verschiedene Typen von Care-of-Adressen:
  Entweder ist die Care-of-Adresse auf einem Fremdagenten (*foreign agent care-of address)* und kann von mehreren mobilen Knoten genutzt werden, oder sie ist auf dem mobilen Knoten selber *(co-located care-of address).*

2.2 Datentransfer

In Figur 1 ist der Ablauf beim Datentransfer aufgezeigt. Wenn ein Endgerät Daten an einen mobilen Knoten senden will, schickt es die IP-Pakete an die bekannte Heimatadresse (1.). Die Router im Internet leiten die Pakete weiter, bis sie beim Heimatagent ankommen (2.). Falls der mobile Knoten im Moment am Heimatnetz angeschlossen ist, werden die Daten wie bei einem festen Knoten zugestellt. Die Mobile IP Funktionalität kommt also gar nicht zum Tragen. Ist der mobile Knoten aber in einem Fremdnetz, so werden die Paket vom Heimatagent gekapselt (wird im Abschnitt 2.2.1 erklärt) und durch einen Tunnel an die Care-of-Adresse des mobilen Knotens weitergeleitet, also nicht über den üblichen Weg mit den Routern (3.). Dieser Vorgang wird *Tunneling* genannt. Das Ende des Tunnels ist der Fremdagent, bei co-located Care-of-Adressen ist es der mobile Knoten, welcher die Pakete entkapselt und an den mobilen Knoten leitet (4.). Wenn der mobile Knoten Pakete versendet, gehen diese ohne Kapselung über die Router zum Zielendsystem (5.).

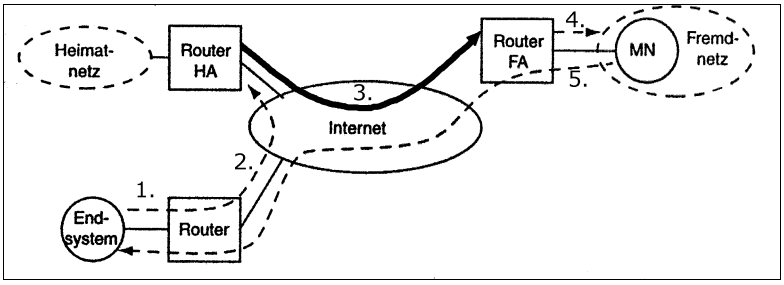


Fig. 1 Ablauf des Datentransfers zwischen einem fixen Endsystem und einem mobilen Knoten

2.2.1 IP-in-IP Kapselung

Wie oben erwähnt, werden IP-Pakete beim *Tunneling* gekapselt. Man nennt diesen Vorgang IP-in-IP Kapselung. Einem IP-Paket wird einfach ein neuer Header verpasst, wobei der alte Header unverändert erhalten bleibt. Auf diese Weise kann der Fremdagent beim entkapseln das Originalpaket wiederherstellen und an den mobilen Knoten senden. Weil bei den beiden Header viele Felder doppelt vorkommen, gibt es auch eine *Minimal Kapselung*, welche die doppelten Felder eliminiert, um die Datenmenge zu minimieren.

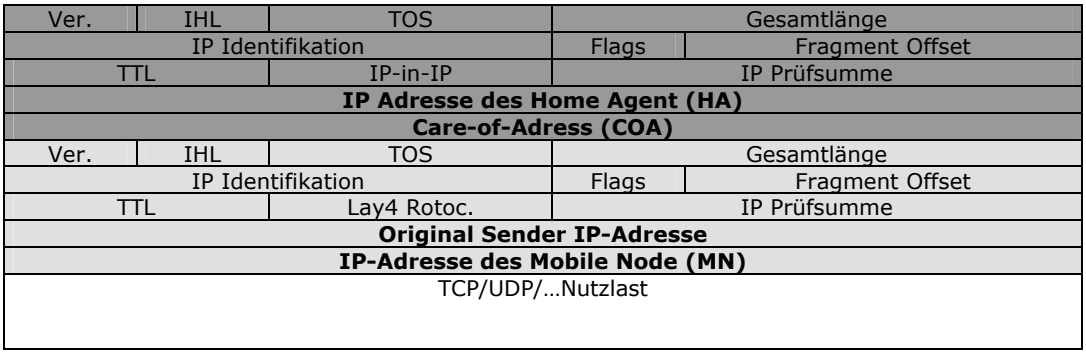


Fig. 2 IP-in-IP-Kapselung: Der neue Header (dunkel) ist fast identisch mit dem alten Header (hell). Unterschiedlich sind die Absender- und Empfängeradressen, sowie das Feld mit dem Protokolltyp: Im neuen Header steht „IP-in-IP“, was bedeutet, dass die Nutzlast wiederum ein ganzes IP-Packet ist.

2.3 Agent Discovery

Damit der Datentransfer von und zu einem mobilen Knoten möglich wird, muss dieser ins Internet integriert werden. Aus Sicherheitsgründen gibt es gewisse Vorkehrungen bei der Integration. Zunächst muss der mobile Knoten herausfinden, in welchem Subnetz er sich befindet. Diesen Prozess nennt man *Agent Discovery*. Darunter versteht man, dass die Agenten periodisch Nachrichten, sogenannte *Agent Advertisements*, in ihre jeweiligen Netze senden. Der mobile Knoten hört diese und kann anhand der Informationen im Header erkennen, ob er im Heimatnetz oder einem Fremdnetz ist. Das geschieht indem der Knoten den Präfix des Absenders mit dem Präfix seiner Heimatadresse vergleicht: stimmen die Präfixe überein, befindet er sich im Heimatnetz. Falls der mobile Knoten sich in einem Fremdnetz befindet, kann er aus der Nachricht die Care-of-Adresse ablesen. Mobile Knoten, denen die Periode der Übermittlung von *Agent Advertisements* zu lang geht, weil sie ihre Lokation sehr schnell wechseln, können Nachrichten von einem Agenten mittels *Agent Solicitations* erzwingen. *Agent Solicitations* werden vom mobilen Knoten gesendet und veranlassen die Agenten sofort *Agent Advertisements* zu senden.

2.4 Registrierung

Im zweiten Schritt der Netzwerkintegration muss der Heimatagent über den Aufenthaltsort des mobilen Knotens informiert werden. Man nennt dies Registrierung. Der Heimatagent führt ein Protokoll, in welchem er allen mobilen Knoten des Heimatnetzes ihre zugehörige Care-of-Adresse zuordnet, damit er weiss, wohin ankommende Pakete weitergeleitet werden müssen. Der mobile Knoten muss sich also vom Heimatagenten registrieren lassen und gegebenenfalls seine neue Care-of-Adresse bekannt geben. Es gibt drei mögliche Szenarien:

- Der mobile Knoten besucht ein neues Netz und lässt mit seiner *co-located care-of adress* beim Heimatagent registrieren.

- Der Knoten lässt sich über einen Fremdagenten registrieren. So erhält der Heimatagent die Care-of-Adresse des Fremdagenten.
- Der Knoten meldet sich beim Heimatagenten ab (Deregistrierung), wenn er gerade das Heimatnetz betritt. Jetzt läuft alles wie bei einem fixen Knoten ab.

Bei der Registrierung sendet der mobile Knoten einen *registration request* und der Heimatagent schickt ein *registration reply* zurück, falls alles in Ordnung ist.

2.5 Optimierung

Das Senden von Paketen über den Heimatagenten ist ein Umweg und ist somit nicht optimal. Eine Verbesserung erreicht man, wenn ein Sender die Lokation des mobilen Knotens lernt. Er bekommt vom Heimatagenten also die Care-of-Adresse nachdem er einmal via Heimatagent gesendet hat (Figur 3, Schritt 1.). Dann kann der Sender die Pakete direkt tunneln (2.). Ein anderes Problem sind Pakete, die noch zur alten Care-of-Adresse unterwegs sind, während der mobile Knoten seine Lokation bereits geändert hat. Das Verlieren der Pakete kann verhindert werden, wenn der neue Fremdagent den alten benachrichtigt (3.). So kann der alte Fremdagent noch eintreffende Pakete weiterleiten (4.) und dann den Sender warnen (5.). Worauf dieser den Heimagenten nach der neuen Care-of-Adresse fragt (6.).

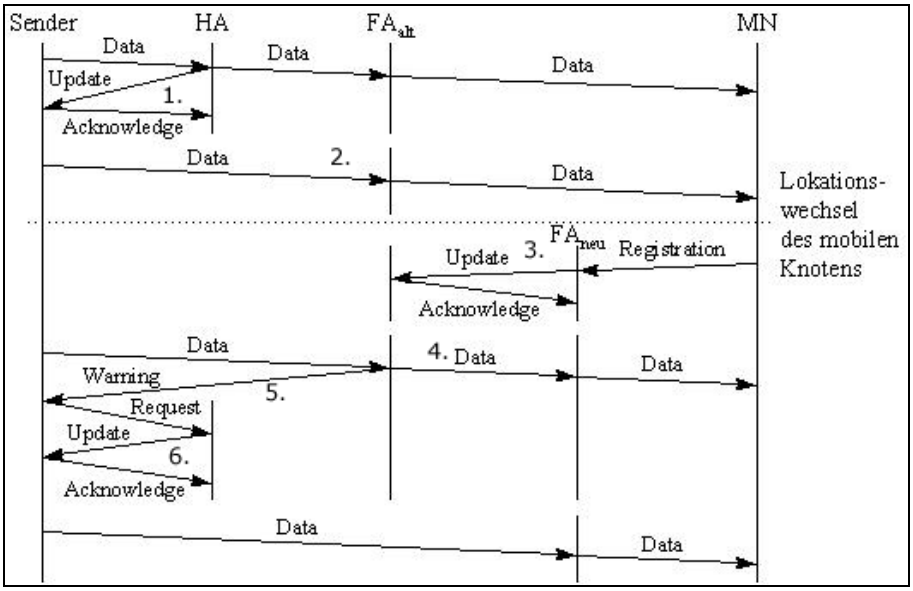


Fig. 3 Zeitlicher Ablauf der Optimierungsvorgänge (die Zeitachse geht von oben nach unten)

2.6 Reverse Tunneling

Im Abschnitt 2.2 über den Datentransfer wurde erwähnt, dass das Senden vom mobilen Knoten zu den festen Endsystemen über den einfachen Weg mit den Routern geht. Der Nachteil dabei ist, dass die Nachrichten eine topologisch falsche Absenderadresse haben, weil sie aus einem Subnetz abgeschickt werden, welches ein anderes Präfix hat als der Absender. Aus Sicherheitsgründen kontrollieren in der heutigen Zeit Router auch die Absenderadressen und halten somit alle Pakete von mobilen Knoten für Nachrichten von unbefugten Eindringlingen. Diese werden alle gelöscht. Damit dies nicht passiert wurde das *Reverse Tunneling* entwickelt. Das erlaubt einem mobilen Knoten die Care-of-Adresse als Absenderadresse zu verwenden und an den Heimatagenten zurück zu tunneln. Dieser kann die Pakete dann mit der richtigen Absenderadresse weiterleiten.

Referenzen

1. rfc2002.txt, Charles Perkins, T. J. Watson Research Center, IBM Corporation
2. http://www.telematik.informatik.uni-karlsruhe.de/lehre/alt/vorlesungen/Tele1-Folien_WS9697/K12-Mobi/sld014.htm
3. http://www.darmstadt.gmd.de/mobile/mobileip/mobileIP.html

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

UDP & TCP

Cyril Stutz
stutzcy@ee.ethz.ch
01. Dezember 2001

1 Internet Transportprotokolle

Das Internet umfasst für die Transportschicht des IP-Modells zwei verschiedene Protokollarten, eine verbindungslose und eine verbindungsorientierte. Bei der verbindungslosen handelt es sich um das User Datagram Protocol (UDP), bei der verbindungsorientierten um das Transmission Control Protocol (TCP). Die meisten Netzwerkanwendungen nutzen für die Datenübertragung eines dieser zwei Protokolle.

1.1 User Datagram Protocol (UDP)

UDP ist ein einfaches Transporschichtprotokoll. Es dient mit wenig Potokollaufwand dem Nachrichtenaustausch zwischen Programmen. Der Austausch von Daten erfolgt verbindungslos, d.h. dass Pakete unabhängig voneinander weitergeleitet werden. UDP gilt als unzuverlässig, denn es übernimmt keine Garantie, dass die Dateneinheiten ihr Ziel erreichen und in einer bestimmten Reihenfolge ankommen. Bei Bedarf kann eine Fehlerbehandlung, die eine Zeitsteuerung, eine Bestätigung von empfangenen Datenpaketen sowie eine wiederholte Übertragung von Datenpaketen beinhalten sollte, in die Applikation selbst integriert werden.

UDP eignet sich vor allem für Client/Server-Anwendungen, die auf der Grundlage eines einfachen Anfrage/Antwort-Paares aufgebaut sind und bei denen auf einen aufwendigen Verbindungsaufbau und -abbau verzichtet werden soll. Im Unterschied zu TCP besteht die Möglichkeit, das Broadcasting und Multicasting des Internet-Protokolls zu nutzen. Typische Beispiele dafür sind Konferenzanwendungen mit Audio- und Videodaten.

1.1.1 UDP Header

Die einzelnen Dateneinheiten des UDP werden als Nutzer-Datagramme bezeichnet. Jede Ausgabeoperation auf das Netzwerk durch einen Anwendungsprozess erzeugt genau ein Nutzer-Datagramm, das wiederum in ein IP-Datagramm verpackt wird und einen eigenen Protokollkopf beistzt.

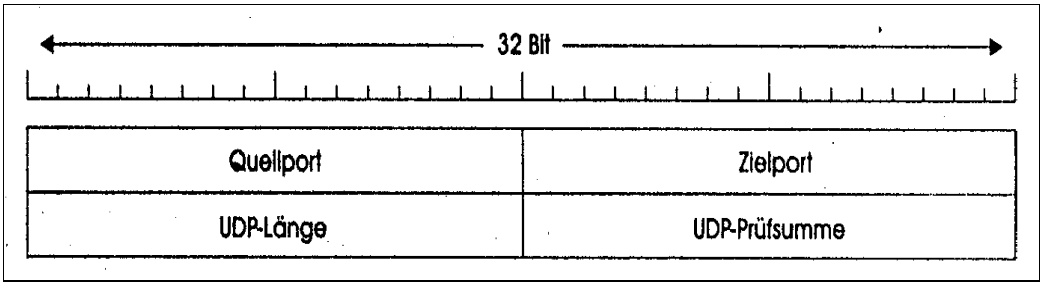


Fig. 1 UDP Header mit einer Grösse von 8 Byte (4 Steuerfelder)

- *Source Port und Destination Port:* Ankommende Daten werden dem jeweiligen Zielprozess über eine Portnummer zugewiesen. Für Standarddienste sind einige Portnummern zentral festgelegt.
- *Length:* Gibt die Grösse des Nutzer-Datagramms inklusive der Kopfinformation in Byte an.
- *Checksum:* Bildet eine Prüfsumme über die Daten.

1.2 Transmission Control Protocol (TCP)

TCP wurde entwickelt, um einen zuverlässigen Datenaustausch zu gewährleisten. Um eine solche Zuverlässigkeit zu bieten, müssen die verschiedensten Merkmale von Verbundnetzen jeweils dynamisch angepasst werden. Nur so kann das gesamte Gebilde robuster gemacht werden. TCP stellt daher den Anwendungen wesentlich umfangreichere Dienstleistungen als das UDP zur Verfügung.

1.2.1 Typische Eigenschaften des TCP

- *Verbindungsorientiert:* Applikationen, die über TCP kommunizieren wollen, bauen eine virtuelle Verbindung zueinander auf. Nach dem Datenaustausch wird die Verbindung wieder abgebaut.
- *Byte-Strom-Orientierung:* TCP überträgt in der Phase des Datenaustausches einen Byte-Strom, ohne diesen zu interpretieren oder Datensätze voneinander zu trennen.
- *Zuverlässigkeit:* Jedem zu übertragenden Byte wird eine Sequenznummer zugeordnet. TCP kann mehrere Bytes gleichzeitig senden. Solche Pakete werden als Segmente bezeichnet. Der Empfänger bestätigt jedes ankommende Segment mit einem sogenannten Acknowledgement. Bleibt eine Bestätigung aus, so wiederholt TCP die Übertragung des unbestätigten Segments. Mit einer Prüfsumme über Header und Daten (ähnlich wie bei UDP) überwacht TCP das Auftreten von Übertragungsfehlern, wie zum Beispiel das Ankommen der Daten in falscher Reihenfolge oder das senden von Duplikaten.
- *Flusskontrolle:* Jede TCP-Instanz besitzt für die Zwischenspeicherung der Segmente einen Puffer beschränkter Grösse. Um einen Überlauf zu vermeiden, teilt der Empfänger dem Sender mit, wie viele Bytes er aufnehmen kann, was als sogenanntes Fenster (Window) bezeichnet wird.
- *Vollduplex:* Die über TCP kommunizierenden Instanzen können gleichzeitig senden und empfangen.

1.2.2 TCP Header

Ähnlich wie bei UDP kapselt das Internet-Protokoll die Dateneinheiten, bei TCP die Segmente, die aus einem 20-Byte-Header sowie den Datenbytes bestehen, in IP-Datagramme. Jedes Segment beginnt mit einem Header mit festem Format. Dem festen Header können Header-Optionen folgen, wobei sich die ersten 20 auf den IP-Header und die zweiten auf den TCP-Header beziehen. Segmente nur mit einem Header, also ohne Daten, sind zulässig und werden normalerweise für Bestätigungen und Steuernachrichten benutzt.

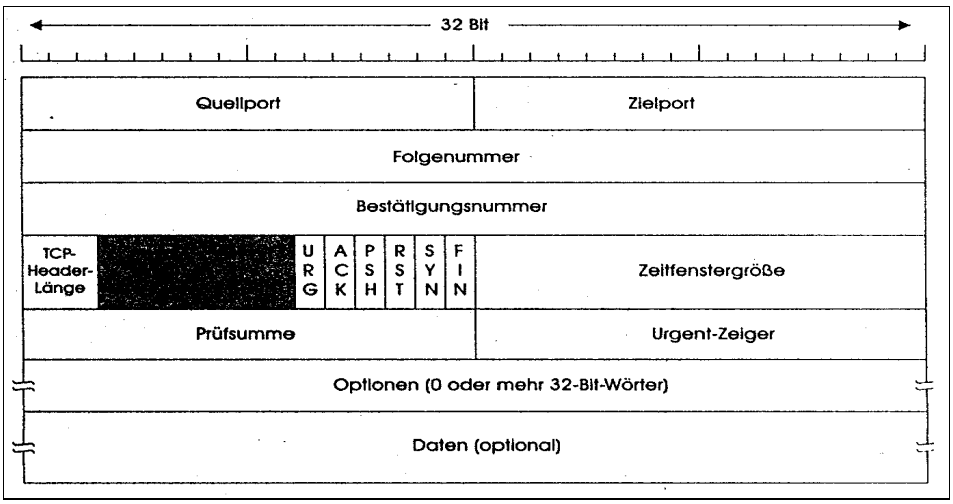


Fig. 2 TCP Header, Grösse: 20 Byte

- *Source Port und Destination Port:* Wie bei der Verwendung von UDP definieren die Portnummern die lokalen Endpunkte der Verbindung. Nummern unter 256 zählen zu den Standarddiensten (well-known-ports), wie zum Beispiel FDP mit Zielhost Port 21 oder Telnet mit Port 23.
- *Sequence Number:* Enthält jeweils die Folgenummer des ersten Daten-Bytes eines Segments.
- *Acknowledgement Number:* Gibt die Sequenznummer des nächst erwarteten Daten-Bytes an, also die Sequenznummer des letzten erfolgreich empfangenen Daten-Bytes plus eins.
- *TCP Header Length:* Repräsentiert die Grösse des TCP-Heades in 32-Bit-Worten. Diese Informationen sind erforderlich, weil das Options-Feld eine variable Länge hat.
- *Reserved:* Dieses Feld wird nicht benutzt. Es ist für zukünftige Verwendungen reserviert. Alle sechs Bits sind mit Null initialisiert.
- *Flags:* Die als Flags bezeichneten Bits haben, je nachdem ob auf eins gesetzt oder nicht, folgende Bedeutung:
  - URG: Wird auf 1 gesetzt, wenn Urgent Pointer benutzt wird. Dringende Daten werden mit diesem Flag initialisiert.
  - ACK: Wird auf 1 gesetzt, um anzugeben, dass die Bestätigungsnummer gültig ist. Ist ACK auf 0, enthält das Segment keine Bestätigung. In diesem Fall wir das Feld Acknowledgement Number ignoriert.
  - PSH: Mit dem Push-Bit wird der Empfänger aufgefordert, die Daten der Anwendung sofort bereitzustellen und sie nicht erst zwischenzuspeichern.
  - RST: Tritt eine Störung auf, beispielsweise ein abgestürzter Host, wird mit dem gesetzten Reset-Bit die Verbindung zurückgesetzt.
  - SYN: Verbindungsaufbau entsteht über das SYN-Bit, indem die Sequenznummer synchronisiert wird.
  - FIN: Analog zum Verbindungsaufbau wird mit dem FIN-Bit die Verbindung abgebaut. Vom Sender kommen keine Daten mehr.
- *Window:* Die Flusssteuerung erfolgt in TCP anhand eines Schiebefensters. Der Wert gibt die Anzahl der Bytes an, die der Absender des TCP-Segments im weiteren Verlauf der Datenübertragung akzeptieren kann. Ein Window-Feld von 0 ist zulässig und besagt, dass die Bytes bis einschliesslich zur Acknowledgement Number empfangen wurden, dass der Empfänger aber im Moment keine Daten mehr aufnehmen kann.
- *Checksum:* Wie beim UDP berechnen die Instanzen des TCP eine Prüfsumme über den TCP-Header und den Datenteil des Segments. Dies soll für optimale Zuverlässigkeit sorgen.
- *Urgent Pointer:* Dient zum Tranport von Vorrangdaten, die im unmittelbarn Anschluss an den Header folgen sollen.
- *Options:* Das Options-Feld bietet die Möglichkeit, zusätzliche Funktionen bereitzustellen, die im Header nicht zur Verfügung stehen. Die wichtigste dieser Optionen spezifiziert für jeden Host die Maximum Segment Size, die er annehmen kann oder will. Möglichst grosse Segmente sind effizienter als kleine, da ein 20-Byte-Header sowohl über ein kleines als auch ein grosses Segment stehen muss. Andererseits können kleine Host sehr grosse Segmente nicht bearbeiten. Wird diese Option von einem Host nicht benutzt, wird die Standardgrösse von 536 Byte herangezogen.

1.2.3 TCP-Verbindungsphasen

In TCP werden die Verbindungen nach dem sogenannten Dreiwege-Handshake aufgebaut. Die eine Seite, der Server, muss passiv auf eine ankommende Verbindung warten, indem er die Option LISTEN und ACCEPT ausführt.

Die andere Seite, der Client, führt eine CONNECT-Operation aus und gibt die IP-Adresse und den Port des Empfängers an. Gleichzeitig gibt er die maximale Grösse des TCP-Segments an, die er akzeptieren kann. Die CONNECT-Operation sendet ein TCP-Segment mit gesetztem SYN-Bit und ausgeschaltetem ACK-Bit und wartet auf eine Antwort.

Kommt dieses Segment am Ziel an, prüft die TCP-Einheit, ob sich der Server im LISTEN-Zustand befindet. Falls dies nicht der Fall ist, wird ein RST-Bit zurückgesendet und die Verbindung wird abgewiesen.

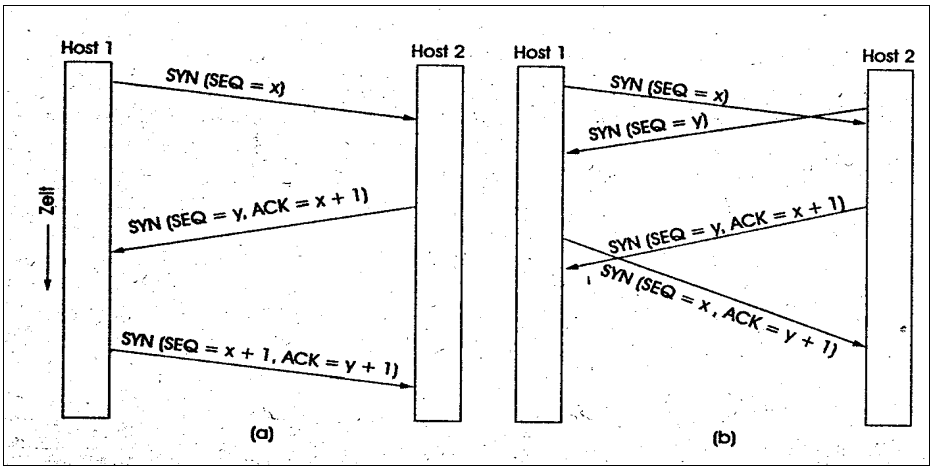


Fig. 3 TCP-Verbindungsaufbau (a) im Normalfall; (b) bei Kollision

Wird ein Prozess angenommen, so wird vom Server ein Bestätigungssegment zurückgeschickt. Die normale Folge von Abläufen ist in Fig. 3(a) dargestellt.

Versuchen zwei Hosts gleichzeitig eine Verbindung zueinander aufzubauen (Simultaneous Open), tritt die in Fig. 3(b) dargestellte Situation ein. Dabei werden nicht zwei, sondern nur eine Verbindung aufgebaut, weil die Verbindungen durch ihre Endpunkte gekennzeichnet sind.

Der Datenfluss innerhalb einer TCP-Verbindung erfolgt unabhängig voneinander in beide Richtungen. Der Datentransfer wird auch unabhängig voneinander für beide Richtungen beendet, was den Austausch von vier Segmenten erforderlich macht.

Die Instanz des Clients führt ein sogenanntes Active Close aus, indem sie ein FIN-Flag abschickt und in den Zustand FIN_WAIT_1 übergeht. Der Instanz des Servers wird somit mitgeteilt, dass aus dieser Richtung keine Daten mehr zu erwarten sind. Der Server reagiert mit einem Passive Close , sendet eine Bestätigung für das erste FIN-Segment und geht in den Zustand CLOSE-WAIT über. Nachdem der Client wiederum diese Bestätigung empfangen hat, befindet er sich im Zustand FIN_WAIT_2 und wartet noch auf das FIN-Segment des Servers. Sendet der Server das FIN-Segment ab, so wechselt er in den Zustand LAST_ACK. Sobald der Client dieses letzte Segment empfängt schliesst der Server. Der Client wechselt vor dem CLOSED noch in einen Zustand TIME_WAIT, um auf eventuelle Verluste von Segmenten zu reagieren.

Wie bei einem Telefongespräch, bei dem sich die zwei Teilnehmer gleichzeitig verabschieden, können beide Enden einer TCP-Verbindung FIN-Segmente senden. Diese werden auf die übliche Weise bestätigt und die Verbindung wird abgeschaltet.

Die zum Aufbau und Abbau von Verbindungen erforderlichen Schritte können in einem Modell (Fig. 4) dargestellt werden. In jedem Zustand sind bestimmte Ereignisse zulässig

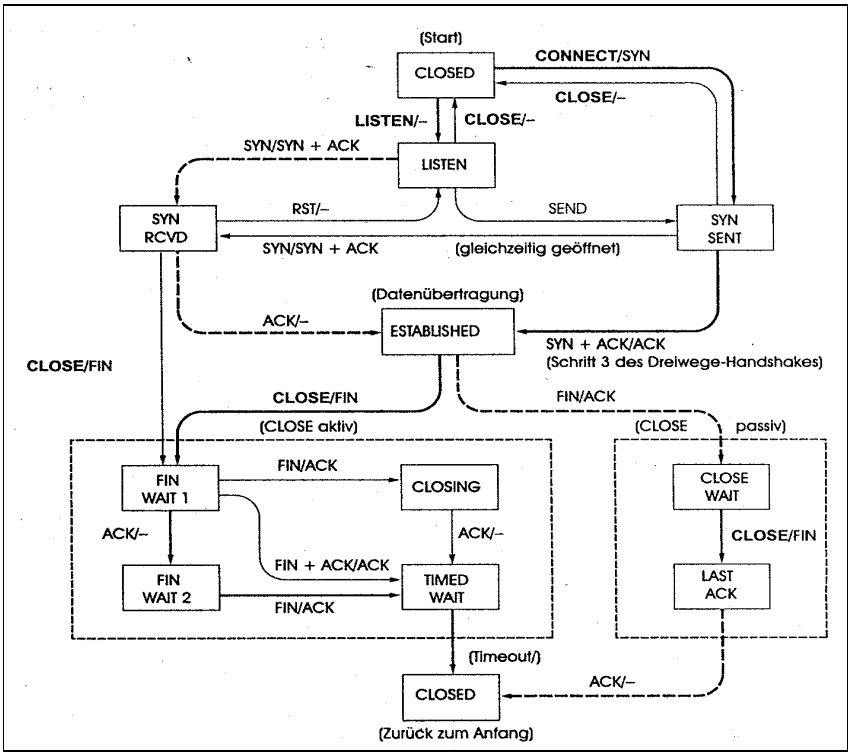


Fig. 4 TCP-Verbindungsmanagement; fette Linie: normaler Pfad des Client, fette gestrichelte Linie: normaler Pfad des Servers, feine Linie: ungewöhnliche Ereignisse, Linien sind jeweils mit einem Ereignis/Aktions-Paar gekennzeichnet

Folgende Zustände treten im Modell auf:

Zustand	**Beschreibung**
CLOSED	Keine Verbindung aktiv
LISTEN	Der Server wartet auf eine ankommende Verbindung
SYN RCVD	Ankunft einer Verbindungsfrage und Warten auf Bestätigung
SYN SENT	Verbindung ist am öffnen
ESTABLISHED	Normale Datenübertragung
FIN WAIT 1	Die Anwendung möchte die Übertragung beenden
FIN WAIT 2	Die andere Seite ist einverstanden, die Verbindung abzubauen
TIMED WAITE	Warten, bis keine Pakete mehr kommen
CLOSING	Beide Seiten haben versucht, gleichzeitig zu beenden
CLOSE WAIT	Dei Gegenseite hat den Abbau eingeleitet
LAST ACK	Warten, bis keine Pakete mehr kommen

Tab. 1 Zustände im TCP-Verbindungsmanagement

1.3 Schlusswort

UDP und TCP sind Transportprotokolle, die in der Schichtenarchitektur oberhalb des Internet-Protokolls stehen. Sie benutzen den Dienst des IP, verfügen jedoch über zusätzliche Funktionen. Der wesentliche Unterschied zwischen UDP und TCP besteht in der Zuverlässigkeit der Datenübertragung. UDP ermöglicht eine schnelle und flexible Übertragung, da eventuelle Fehler übergangen werden. TCP bietet hingegen die Gewissheit, dass die Daten korrekt übertragen werden.

Quellenverzeichnis

1. T. Braun: IPng – Neue Internet-Dienste und virtuelle Netze; dpunkt Verlag, Heidelberg, Deutschland, 1999
2. S. Thomas: IPng and the TCP/IP Protocols, John Wiley & Sons, Inc., New York, USA, 1996
3. http://www.networkmagazine.com

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Das Hypertext Transfer Protocol (HTTP)

Stephan Rüegger
sruegger@unlike.ch
12. Dezember 2001

1 Einführung [2]

Beim World Wide Web handelt es sich im Grunde um ein verteiltes Hypermedia System, bei dem Informationen in Form von Web-Seiten gespeichert werden, die über Links (URL's) abrufbar sind. Das erfordert Regeln der Kommunikation zwischen den Kommunikationspartnern, eine Definition, wie sie auf bestimmte Nachrichten zu reagieren haben. Dies geschieht üblicherweise mit Hilfe eines Protokolls, in diesem Fall mit Hilfe des HTTP Protokolls. Das Protokoll legt fest, auf welche Weise die Kommunikation gestartet und beendet werden kann, welche Konstrukte Teile der Kommunikation sein dürfen und wie die Kommunikationspartner auf die unterschiedlichen Konstrukte reagieren sollen.

1.1 Kommunikationspartner

Im Rahmen des Transfers von Dokumenten findet die Kommunikation zwischen genau einem Dienstnutzer und einem Diensterbringer, der das Dokument zur Verfügung stellt, statt. Der sogenannte "User Agent" (im Allgemeinen ein Browser, Editor oder anderes Endnutzer-Programm) übersetzt die Anforderungen des Dienstnutzers in die verwendete Kommunikationssprache, der "Server" verarbeitet diese Anfrage und generiert darauf eine Antwort. (Siehe Abb.1) So benötigen weder Dienstnutzer noch Diensterbringer eine genaue Kenntnis des zur Kommunikation verwendeten Protokolls. Das ist auch der Grund weshalb ein durchschnittlicher Web-User nie direkt mit HTTP in Berührung kommt. Er macht höchstens ab und zu Bekanntschaft mit Statuscodes, die dem Dienstnutzer Fehlermeldungen anzeigen. Bsp: "404 Not Found".

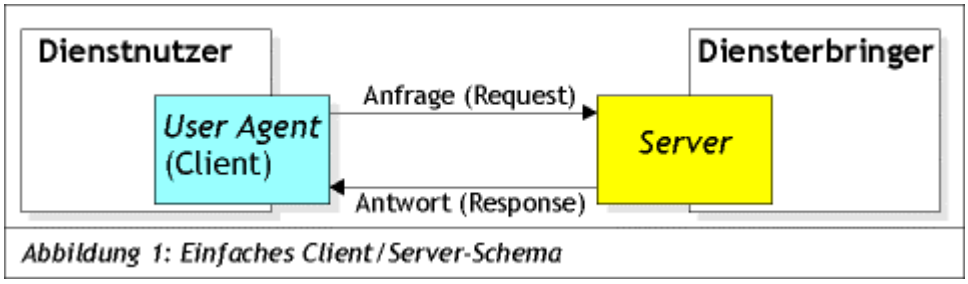


Abbildung 1: Einfaches Client/Server-Schema

1.2 Anforderungen an das Protokoll

- **Pull-Kommunikation:** Die Kommunikation soll durch den Client und nicht durch den Server veranlasst werden. Der Server hat jederzeit auf Anfragen von Clients antworten zu können.
- **Uniforme Identifizierung:** Ressourcen müssen in einem globalen Netz uniform, eindeutig und verlässlich identifizierbar sein, sie dürfen nicht von Standort oder Kontext des Nutzers abhängen.
- **Uniforme Adressierung:** Ressourcen müssen in uniformer und eindeutiger Weise adressierbar sein, so dass sie von jedem beliebigen Client abgerufen werden können. (Adressierung ist nicht gleich Identifizierung! Die Identifizierung muss konstant auf die gleiche Ressource zeigen, dagegen können Adressen während ihrer Lebenszeit ändern.)
- **Zuverlässiger Transport:** Es muss sicher gestellt sein, dass eine durch den Nutzer angeforderte Ressource vollständig und unverfälscht bei diesem ankommt.
- **Keine Authentifizierung:** Die zur Verfügung gestellten Daten sollen generell ohne Authentifizierung des Nutzers verfügbar sein (anders als bei Dateidiensten wie FTP).
- **Einfachheit / Kompatibilität:** Das Protokoll sollte für den Transfer von Ressourcen möglichst einfach sein, um auf allen denkbaren Plattformen eingesetzt werden zu können und als Grundlage für andere Anwendungen zu dienen.

2 Was ist HTTP? [2]

Angenommen ein Student möchte ein Buch bestellen. Dazu muss der Besteller (Client) das gewünschte Buch (die gewünschte Ressource) dem Verkäufer (Server) gegenüber spezifizieren. Dieser besorgt das Buch (die Daten) und sendet sie, abhängig von den Wünschen (Parametern) des Kunden (z.B. Zeit der Versendung, Verpackung, Lieferant etc.) über den Lieferanten an diesen (vgl. Abbildung 2). Entscheidend ist, dass für dieses Protokoll irrelevant ist, wie der Lieferant das Buch ausliefert (z.B. mit welchem Verkehrsmittel, auf welcher Route etc.). Von Interesse ist einzig und allein, dass der Lieferant das Buch zuverlässig in einer geeigneten Zeitspanne ausliefert.

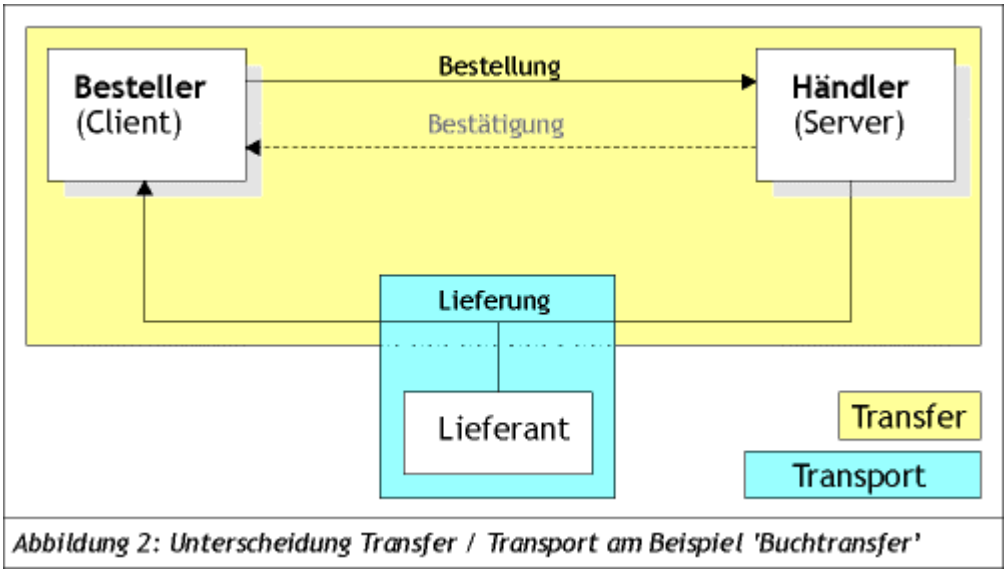


Abbildung 2: Unterscheidung Transfer / Transport am Beispiel 'Buchtransfer'

2.1 Kommunikationsstruktur

HTTP verwendet grundsätzlich drei Formen der Verbindung, wie sie in Abbildung 3 dargestellt sind:

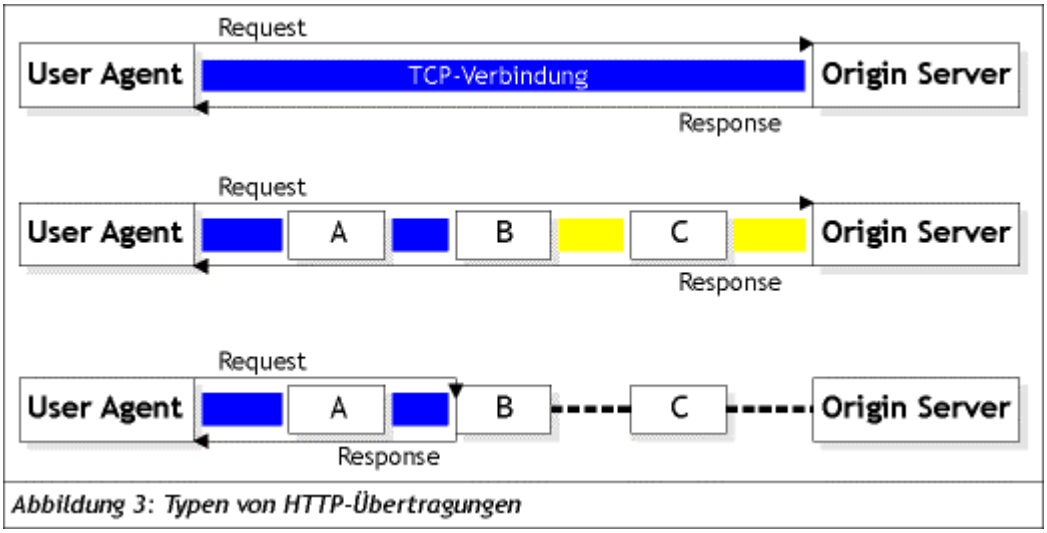


Abbildung 3: Typen von HTTP-Übertragungen

1. Direkte Verbindung: In diesem einfachsten Fall wird unmittelbar eine direkte Verbindung zwischen dem User Agent und dem Origin Server aufgebaut.

2. Mittler-Verbindung: Verbindung über einen oder mehrere Mittler (Proxy, Gateway oder Tunnel), wobei die Mittler die Anfragen des User Agents interpretieren und u.U. verändern können.

3. Verbindung unter Nutzung eines Caches: Der Client nimmt Verbindung mit einem als Cache dienenden Mittler (Proxy) anstelle des Origin Servers auf. Ist die Ressource bereits gecached und noch aktuell, kann der Cache die Anfrage beantworten ohne den Origin Server zu kontaktieren.

Tatsächlich hat sich gerade diese Vielfalt an Verbindungsmöglichkeiten als ein entscheidender Faktor für den raschen Erfolg von HTTP erwiesen, da durch die zweite Verbindungsform etablierte Internetprotokolle (wie FTP) und Legacy-Systeme leicht integriert werden konnten, während die dritte Verbindungsform den Netzverkehr (vor allem aus grossen Organisationen) deutlich reduzieren kann.

2.1.1 Mittler

Es gibt also, wie bereits festgestellt, drei Typen von Mittlern (Intermediates), wie in Abbildung 4 dargestellt:

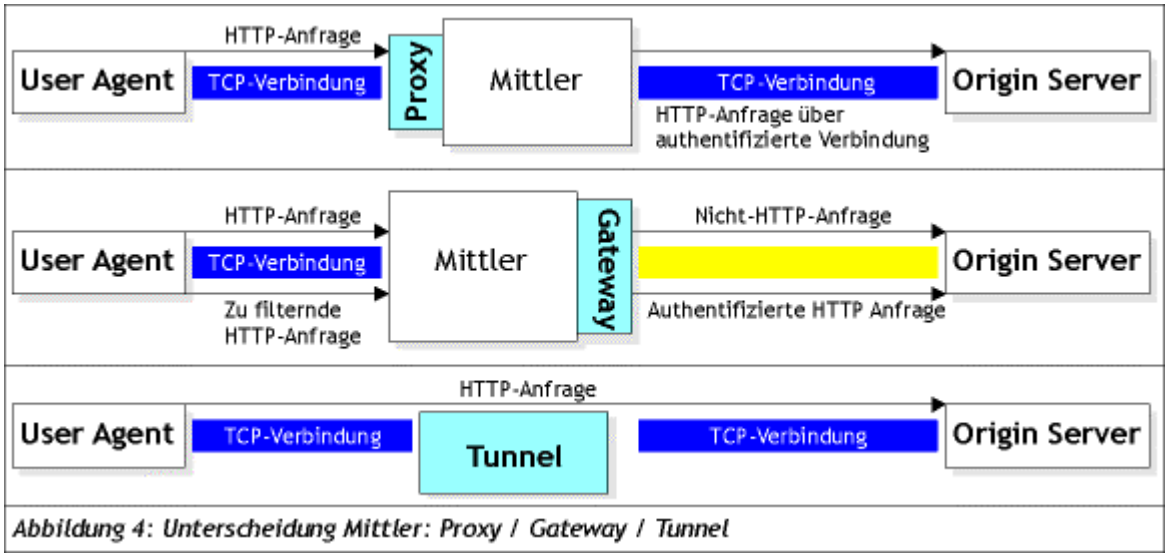


Abbildung 4: Unterscheidung Mittler: Proxy / Gateway / Tunnel

Proxy: Ein Proxy ist ein zwischengeschaltetes Programm, dass sowohl als Server, als auch als Client dient, um Anfragen im Auftrag anderer Clients zu stellen. Anfragen werden entweder intern verarbeitet (z.B. bei Caching) oder möglicherweise nach vorheriger Transformation an andere Server weitergeleitet. Ein Proxy muss eine Anfrage interpretieren und, wenn nötig, umschreiben, bevor sie weitergesendet wird. Zur Veranschaulichung betrachte man Abbildung 3 bzw. 4. Proxies werden häufig als *Client-seitige* Portale (Durchgänge) durch Firewalls oder als Hilfsanwendung zur Verarbeitung von Anfragen über Protokolle, die der User Agent nicht kennt, verwendet. Durch Proxies ist also eine sichere Anbindung von organisationsinternen Netzen (LANs) an das Internet möglich. Gleichzeitig dienen sie häufig als Caches.

Gateway: Ein Gateway ist ein Server, der als Mittler für andere Server dient: Anders als ein Proxy empfängt ein Gateway Anfragen als wäre er der Server, auf dem die Ressource liegt. Dabei müssen die Clients keine Kenntnis davon haben, dass die Kommunikation mit Hilfe eines Gateways stattfindet. Üblicherweise werden Gateways als Server-seitige

Portale durch Firewalls und als Mittler für den Zugriff auf Ressourcen, die auf Nicht-HTTP-Servern gespeichert sind (vgl. Abbildung 3 und 4).

Tunnel: Unter einem Tunnel schliesslich versteht man ein Mittlerprogramm, das eine blinde Weiterleitung vornimmt. Ist die Verbindung einmal hergestellt, wird der Tunnel nicht länger als Partner in der HTTP-Kommunikation betrachtet, er wird geschlossen, sobald beide Enden der Verbindung geschlossen werden. Üblicherweise wird ein Tunnel benutzt, wenn ein Portal nötig ist und der Mittler die versendeten Nachrichten nicht interpretieren kann oder soll.

2.1.2 Nutzung von Caches

Nutzer (insbesondere innerhalb einer Organisation) neigen dazu, auf Ressourcen, auf die sie bereits zugegriffen haben, erneut zuzugreifen. Darum macht es Sinn sogenannte Caches einzusetzen. In einem Cache kann ein Programm Antwort-Nachrichten (z.B. Ressourcen wie Hypertext-Dokumente) verschiedener Server speichern, so dass bei einem erneuten Zugriff auf die gleiche Ressource nicht die komplette Antwort-Nachricht erneut übertragen werden muss. Dadurch kann der Verkehr im Internet deutlich reduziert werden.

2.2 Transport der Daten

Einer der grossen Vorteile eines sauber definierten Protokolls ist es, dass seine Dienste selbst wieder zur Implementierung von Protokollen für Dienste höherer Ebene realisiert werden können. Dadurch ergibt sich eine Hierarchie oder Schichtung der benötigten Protokolle. Im Rahmen des Internets ist dies der TCP/IP-Protokollstack, innerhalb dessen HTTP ein Protokoll der Anwendungsschicht (Application Layer), wie in Abbildung 6 dargestellt:

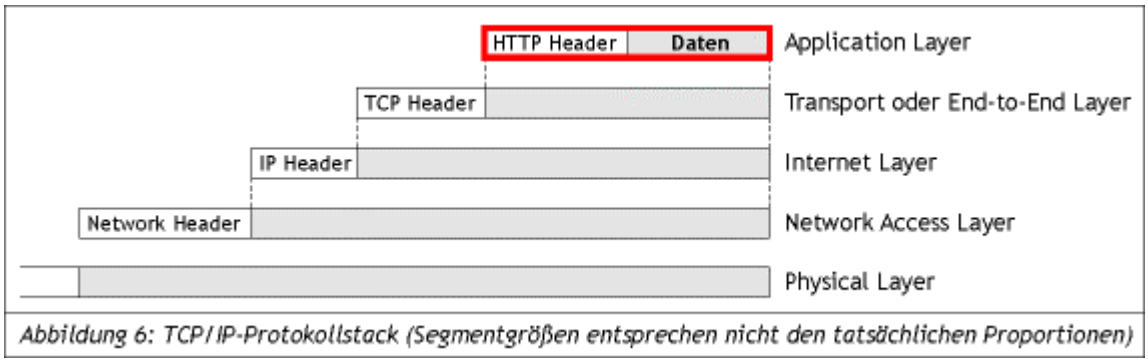


Abbildung 6: TCP/IP-Protokollstack (Segmentgrößen entsprechen nicht den tatsächlichen Proportionen)

2.2.1 HTTP Header

Der HTTP Header beginnt, je nach dem ob es sich um einen Request oder um einen Response handelt, mit einer request-line bzw. einer status-line. Anschliessend enthält der Header null oder mehrere Header-Felder, die sich in vier Gruppen zusammenfassen lassen:

- **General Header:** Hat keinen Einfluss auf die zu übertragende Daten. Enthält generelle Informationen wie Datum und Uhrzeit.
- **Entity Header:** Enthält Informationen über den Entity Body (Nachrichtenkörper) wie Länge, Codierung, letzte Änderung, ...
- **Request Header:** Enthält Informationen über den Request und den Client selbst.
- **Response Header:** Zusätzliche Informationen für den Client, die nicht in der status-line angegeben werden können

3 Entwicklung von HTTP [1]

HTTP ist ein verbindungsorientiertes Protokoll und wurde ursprünglich entworfen um einfache textbasierte Dokumente von einem Server abrufen zu können. Die damaligen Hauptziele des Protokolls haben sich bis heute nicht geändert: Einfachheit des Protokolls, damit es nicht viele Ressourcen beansprucht und Schnelligkeit des Protokolls, da eine grosse Anzahl von Dokumenten auf einer grossen Anzahl von Servern verteilt ist und trotzdem das schnelle Abrufen von Informationen gewährleistet sein soll.

3.1 HTTP 0.9

Begonnen hat die Entwicklung von HTTP mit Tim Berners-Lee am CERN in den Jahren 1990/1991. Die Spezifikationen der ersten Version des Protokolls, einfach HTTP genannt, bestand lediglich aus einigen Seiten, im Vergleich zu heute, wo sie etwa 180 – 230 Seiten umfasst. Die Version 0.9 unterstützte ausschliesslich die Methode “GET“. Ein Client musste eine Verbindung zum Server aufbauen und eine Zeile mit dem Schlüsselwort GET und dem Namen des Dokuments senden. Der Server antwortete dann mit dem Dokument selbst und brach unmittelbar danach die Übertragung ab, um das Ende des Dokuments zu signalisieren.

Die Nachteile dieser Version wurden sofort klar, als sich das Web immer schneller ausbreitete und sich daraus eine grossflächige Anwendung zu entwickeln begann: HTTP 0.9 war nur in der Lage Texte zu übertragen und ein Client konnte keine Daten an den Server übermitteln. Darum war es nötig diese Version zu überarbeiten und Lösungen für eine neue Version zu finden.

3.2 HTTP 1.0

1992 wurde mit der Entwicklung von HTTP 1.0 begonnen. Die endgültige Version wurde jedoch erst 1996 freigegeben. Die Spezifikationen waren aber lediglich zur Information gedacht und noch nicht als Standarddokument. Es war offensichtlich, dass auch diese Version bald durch eine neue ersetzt werden sollte. Trotzdem stellte HTTP 1.0 wesentliche Verbesserungen gegenüber der alten Version 0.9 dar. Es war möglich verschiedene Medientypen auszutauschen und auch Informationen über die Medientypen mitzusenden. HTTP 1.0 definierte ein flexibles Nachrichtenformat, welches aus Anfangszeile sowie aus Header-Feldern mit einer frei wählbaren Anzahl von Zeilen bestand. Weiter wurde neben der Methode GET die Methode POST zugefügt. Damit wurde es dem Client möglich, dem Server Daten zu übermitteln. Weiter wurde auch ein Konzept der Benutzerauthentifizierung vorgestellt, welches die Einschränkungen des Ressourcenzugriffs beinhaltet. Dieser Mechanismus war jedoch noch sehr unsicher.

Im Netzwerkbereich hat sich gegenüber der Version 0.9 aber nicht viel geändert. Die neue Version basierte weiterhin auf einer einzelnen Request/Response-Interaktion, die dazu zwang, nach jedem Versenden eines Requests die Verbindung abzubrechen, was natürlich sehr ineffizient war und erst in der Version 1.1 geändert wurde durch das sogenannte Persistent-HTTP.

3.3 HTTP 1.1

Die grundlegende Idee hinter dem Persistent-HTTP besteht darin, eine nach einer Request/Response-Interaktion aufgebaute Verbindung bestehen zu lassen und abzuwarten, ob weitere Requests an denselben Server gesendet werden. Dies hat zur folge, dass eine TCP-Verbindung nicht mehr so häufig geschlossen und wieder aufgebaut werden muss. Diese neue HTTP-Version wurde im Januar 1997 freigegeben.

Weitere Neuerungen waren die Einführung neuer Request-Methoden, namentlich DELETE, OPTIONS, PUT und TRACE, Unterstützung der fragmentierten Übertragung von Daten, Wahl zwischen verschiedenen Darstellungsformen einer Ressource, die sich in Bezug auf Sprache, Qualität, Codierung oder anderer Parameter unterscheiden können, aber keinen Einfluss auf den Inhalt einer Ressource haben, ausgereiftes Caching in grösserem Umfang und Verbesserung der Sicherheit des Authentifizierungsschemas.

3.4 Zusammenfassung

Es ist deutlich erkennbar, dass sich HTTP in der Version 1.1 weit von seinen einfachen Ursprüngen entfernt hat. So gilt HTTP 1.1 heute eher als unhandlich und ineffizient. Zwar hat HTTP 1.1 einen grossen Teil der Probleme der ersten Version gelöst, ist aber in vielen Bereichen inkonsequent. Das grösste Problem stellt aktuell die fehlende oder fehlerhafte Unterstützung der Erweiterungen in HTTP 1.1 durch die üblichen HTTP-Server und User Agents dar. Neue Methoden wie OPTIONS werden häufig nicht konform umgesetzt, andrerseits werden permanent neue Features hinzugefügt, ohne die möglichen Folgen zu überblicken.

Quellen:

1. Erhaltene Literatur
2. http://www.pms.informatik.uni-muenchen.de/lehre/seminar/html-metamorphosen/00ss/ausarbeitungen/HTTP/

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Die HTML Beschreibungssprache

Carlo Mathys
mathysc@ee.ethz.ch
08 Dezember 01

1 Einführung

HTML (**H**yper**t**ext **M**arkup **L**anguage) ist jedem von uns schon mehr oder weniger bewusst begegnet. HTML ist die Programmiersprache, die verwendet wird um Internetseiten zu gestalten. Ein Browser (Explorer, Netscape Navigator, Opera usw.) interpretiert die Sprache und gibt diese formatiert aus.

Die Sprache wurde 1990 im Rahmen eines Projektes im CERN in Genf erfunden und ist nach ständiger Weiterentwicklung, Verbesserungen und Erweiterungen mittlerweile bei Version 4.0 angelangt.

Die in frühen Veröffentlichungen verfassten Richtlinien bzw. Ziele, welche jedoch ohne weiteres auch heute noch gelten, lauteten wie folgt:

- **Leistungsfähigkeit**

  Es soll eine grosse Anzahl möglicher Anwendungen unterstützt werden. Dies wird erreicht, indem die Sprache allgemein gehalten wird. Also mit gewissen Freiheiten. Tritt zum Beispiel ein Schreibfehler im Code auf versucht der Browser trotzdem irgendwie die Seite darzustellen.

- **Einfachheit**

  Die Sprache soll nicht zu komplex sein um es vielen zu ermöglichen, die Sprache in angemessener Zeit zu erlernen und ihre eigenen Seiten zu gestalten.

- **Zugänglichkeit und Plattformunabhängigkeit**
  Die Sprache soll inhaltsorientiert sein. Einfach ausgedrückt geht es primär darum, Informationen einem breiten Publikum zugänglich zu machen, wobei das Layout eher zweitrangig ist. Zudem soll die Sprache auf allen Rechnern verstanden oder geschrieben werden können, egal welche Plattform.

2 Geschichte

1989 wurde HTML im Rahmen eines Projektvorschlags im CERN in Genf veröffentlicht. 1990 begann schliesslich die Entwicklung eines Prototyps der Sprache, welche noch binnen Jahresfrist fertig gestellt wurde. An einer Konferenz 1992 lagen bereites die Basiskonzepte des Web, also URLs, HTTP und HTML vor. Einfache Gestaltungsmöglichkeiten wie Überschriften und Listen waren bereits implementiert.

2.1 HTML 2.0

Da nun verschiedene Teams an Weiterentwicklungen und insbesondere Erweiterungen an eigenen Browsern arbeiteten, drohte bereits ein Wirrwarr an verschiedenen Funktionen der einzelnen Browser. Um dem entgegenzuhalten wurde 1994 die Sprache ein erstes Mal standardisiert. Zur selben Zeit wurde auch die Firma Netscape gegründet, welche gleich begann, neue Funktionen in ihrem ersten Browser hinzuzufügen, was unweigerlich zu einer schnellen Veraltung des soeben neuen Standards führte aber auch massgeblich zum Erfolg der Firma beigetragen hat.

2.2 HTML 3.2

Die Empfehlung der Version 3.0 durch das 1994 gegründete World Wide Web Consortium (W3C) wurde noch während des Ratifizierungsprozesses durch neue Browser welche bereits mit neuen Elementen erweitert waren, verdrängt, obwohl einige Elemente der Version 3.0 nicht vollständig unterstützt wurden.
Somit blieb vorerst HTML 2.0 der Standard, obwohl bereits Tabellen, Frames und Skripte in neuen Browsern implementiert wurden. Im Januar 1997 wurde schliesslich HTML 3.2 freigegeben. Erstmals waren Tabellen, Applets, Textfluss um Bilder und weitere Funktionen, die bereits stark verbreitet waren, offiziell zum Standard hinzugefügt. Frames waren trotz Wunsch seitens Netscape im Jahre 1995 noch nicht integriert.

2.3 HTML 4.0

Da HTML 3.2 bei der Veröffentlichung bereits veraltet war, dauerte es diesmal nicht mal ein Jahr bis die neue Version 4.0 im Dezember 1997 veröffentlicht wurde. Wie in den vorangegangenen HTML Versionen implementierte das W3C die vorgeschlagenen Funktionen in ihrem Test-Browser (Amaya).
Zu den wichtigsten neuen Funktionen zählen die Unterstützung von **Style Sheets**, **Frames**, ein wesentlich **verbessertes Tabellenmodell**, die Einbindung von **Multimediaobjekten** und besser ausgestatte **Formulare** sowie eine **Internationalisierung.**
Zudem wurde HTML 4.0 ISO-standardisiert. Die ISO-Version ist strenger in der Durchsetzung eines „schönen“ Programmierstils, weil die Anwendung von veralteten Elementen, die nicht mehr eingesetzt werden sollen, nicht erlaubt ist.
Da HTML-4.0-Implementierungen in der Lage sein sollten, auch ältere Konstrukte zu verstehen, wurden drei so genannte **D**ocument **T**ype **D**efinitions (DTDs) definiert.

2.4 Die Document Type Definitions

DTDs sind Listen, die sämtliche erlaubte Attribute enthalten, die im jeweiligen DTD erlaubt sein sollen. DTDs werden vom W3C definiert und veröffentlicht.

- **Transitional DTD**

  Hierbei handelt es sich um eine DTD, die ausschliesslich zum Interpretieren und nicht zum Erstellen von HTML 4.0 Dokumenten verwendet werden soll. Sie enthält Elemente, die noch gültigen HTML-Code darstellen aber nicht mehr verwendet werden sollten. Der grösste Teil dieser Elemente kann und sollte heute mit **C**ascading **S**tyle **S**heets (CSS) ersetzt werden. Es handelt sich hier also grösstenteils um Attribute die mit der Formatierung der Seite zu tun haben.

- **Strict DTD**

  Diese DTD dient der Herstellung von HTML-Seiten. Da eine Vielzahl von Attributen des Transitional DTD nicht mehr verwendet werden sollen, wurden diese bei der Definition von Strict DTD gestrichen. Sämtliche Elemente können mit CSS oder anderen Attributen programmiert werden.

- **Frameset DTD**

  Falls Frames verwendet werden, wird der Inhalt einer Seite eigentlich in verschiedenen HTML-Dokumenten definiert. Wobei ein Dokument die Struktur der Frames definiert (das Frameset) und eine Reihe anderer Dokumente den eigentlichen HTML-Code der Frames enthalten. Dieses DTD enthält deshalb die für die Spezifikation des Framesets verwendeten Attribute.

3 Aufbau eines HTML-Dokuments

Wie bereits erwähnt dient Transitional DTD nur der Interpretation und das Frameset DTD im Falle einer Herstellung einer Seite mit Frames. Daher sollte Strict DTD zum erstellen von HTML-Seiten verwendet werden. HTML Seiten bestehen aus zwei Teilen. Dem Document Head, der Informationen über das Dokument enthält und dem Document Body, der den eigentlichen Inhalt der Seite enthält. Also das, was wir in einem Browser dargestellt bekommen.

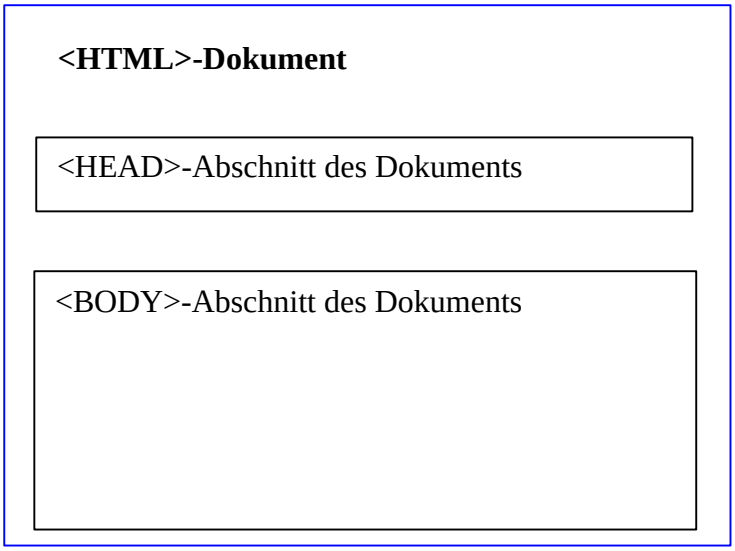


Fig. 1 Der Grundlegende Aufbau eines HTML-Dokuments

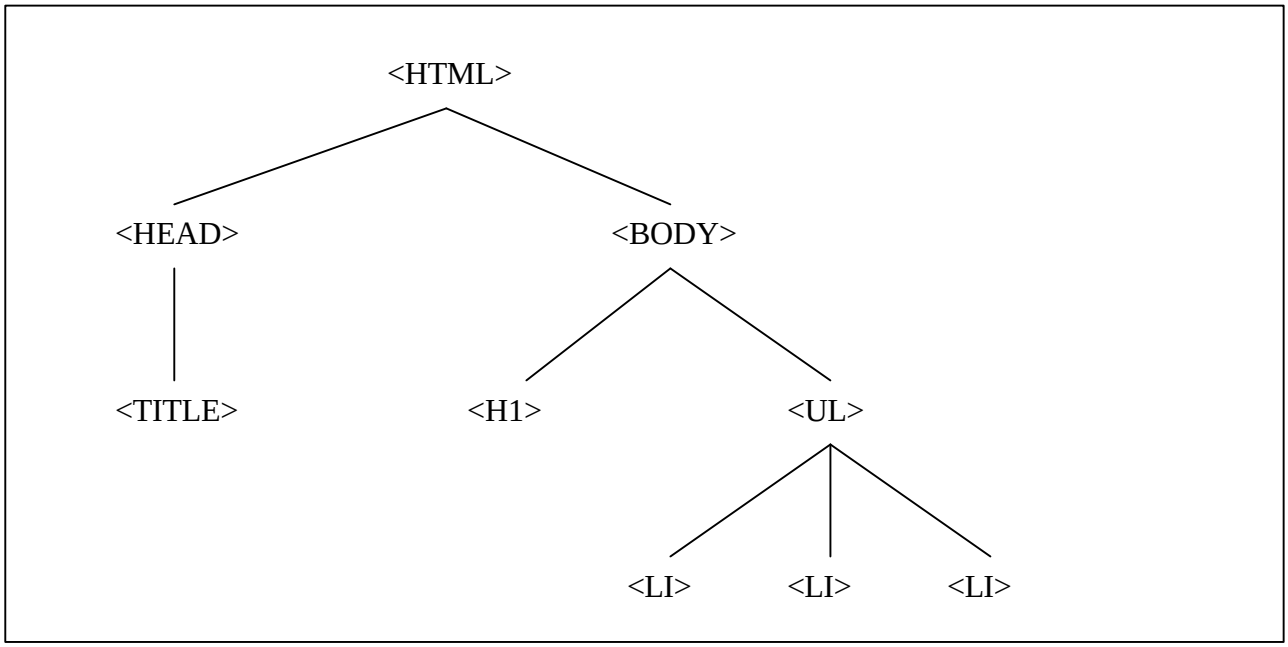


Fig. 2 Die Hierarchie eines HTML-Dokuments. H1 = Überschrift, UL = Liste, LI = Listenelemente

3.1 Der Document Head

Der Head enthält Informationen über das Dokument, wie den Titel (dieser wird im Browserfenster angezeigt), Schlüsselwörter (für Suchmaschinen), Style Sheets, Skripte (z.B. JavaScript) oder auch eine einfache Beschreibung der Seite. Sämtliche Informationen dienen praktisch ausschliesslich irgendwelchen Clients wie Suchmaschinen. Er kann als eine Art Inhaltsverzeichnis verstanden werden. <head>….</head> beinhaltet den ganzen head. Sehr schön zu sehen auf dem Screenshot [Fig. 4]

3.2 Der Document Body

Der Body enthält den eigentlichen Inhalt der Seite also Text, Bilder, Multimediaobjekte (z.B. Musik) inklusive deren Formatierung. Hier werden auch die Farben für Schrift, Links, Hintergrund usw. definiert. Konkrete Beispiele sind weiter unten aufgeführt. Hier erfolgt nun die genaue „Platzierung“ der Style Sheets oder Skripte. Der Body beginnt logischerweise mit dem <body> - tag und wird schliesslich mit </body> beendet. Die [Fig. 3] zeigt den ganzen Inhalt einer sehr einfachen Homepage. Es ist relativ einfach den Code parallel zu vergleichen und die einzelnen Attribute herauszulesen und was diese bewirken.

3.3 Der direkte Vergleich zwischen HTML und Screen

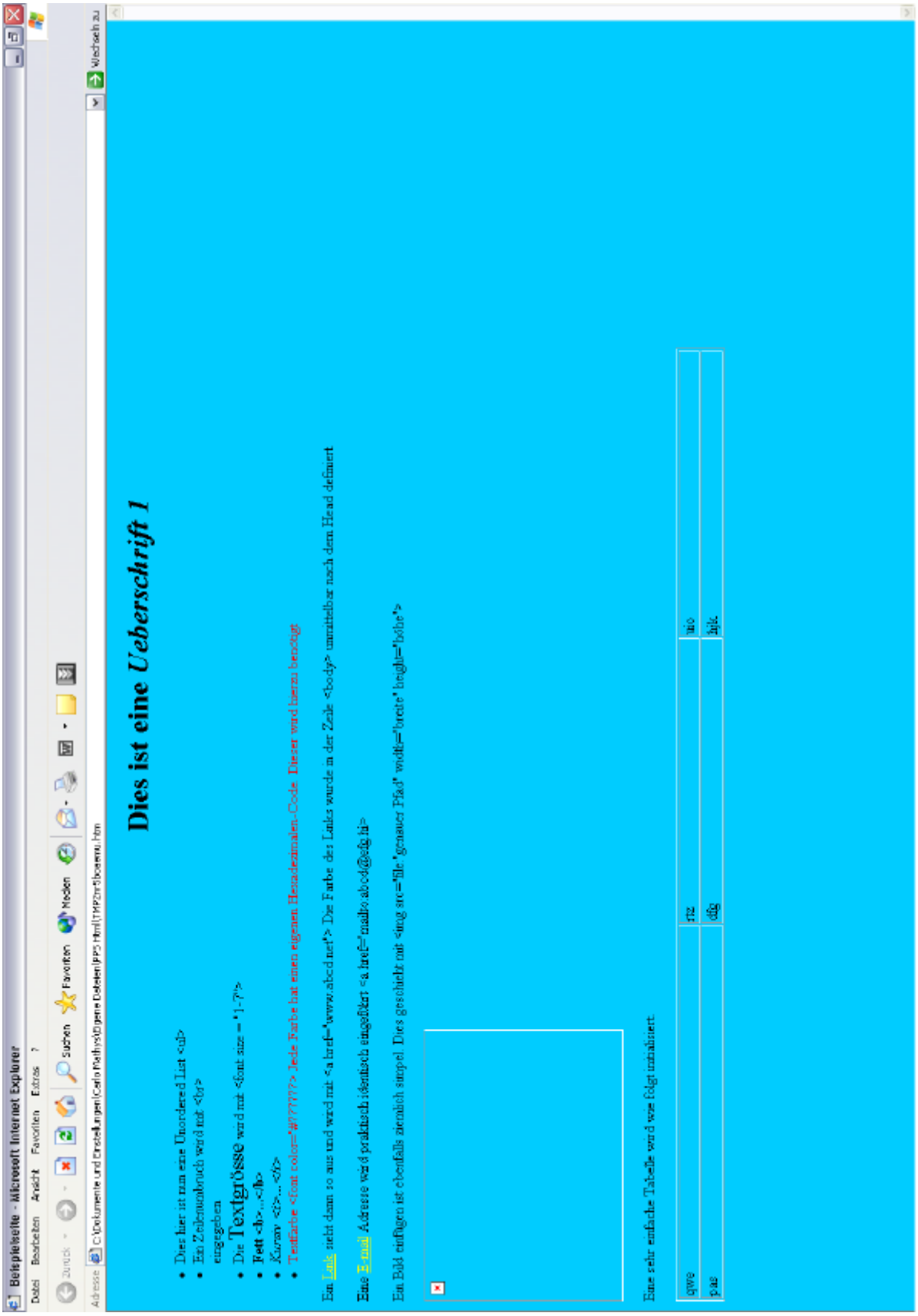
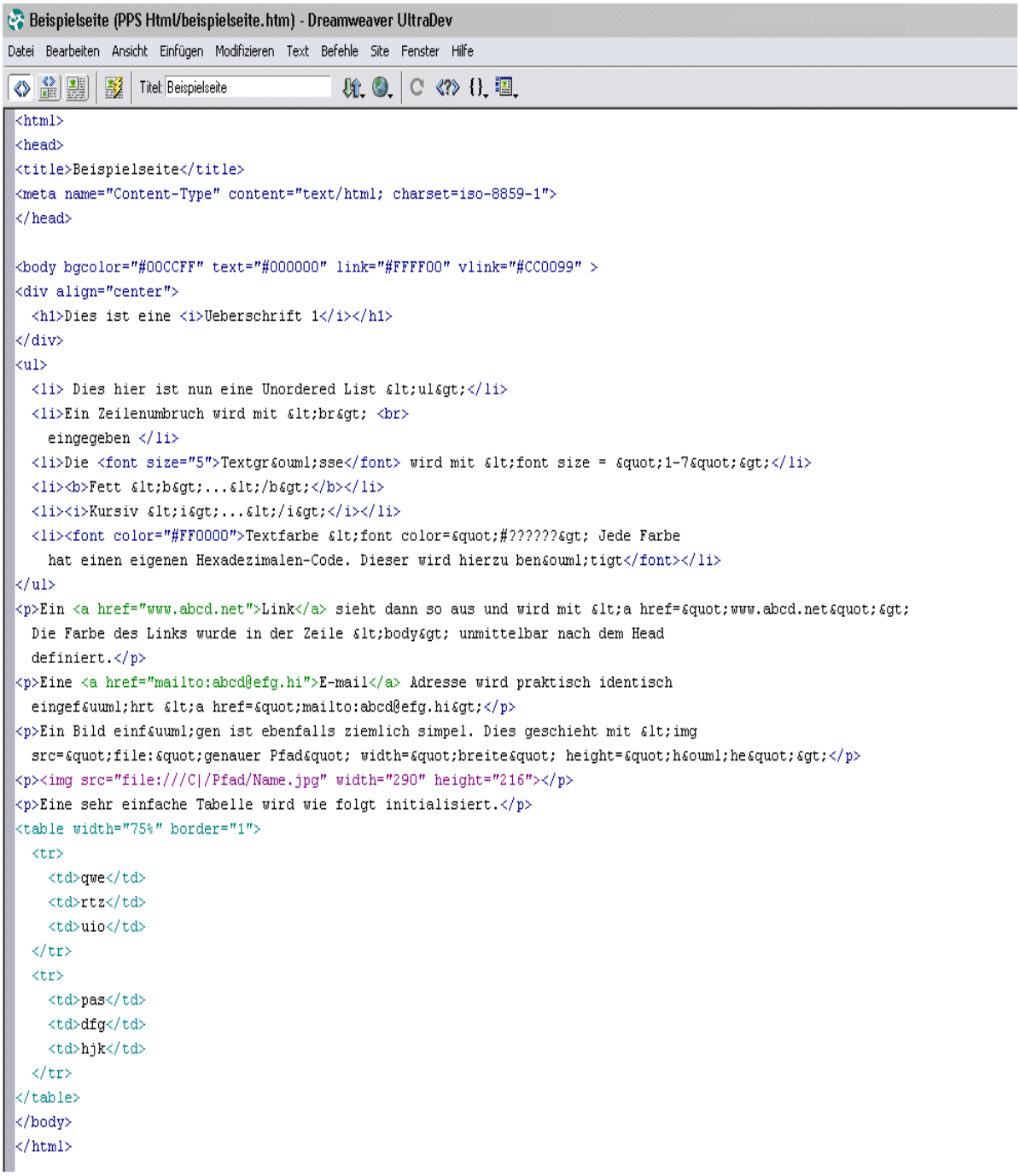


Fig. 3 Screenshot der Beispielpage mit dem Internet Explorer




```
<html>
<head>
<title>Beispielseite</title>
<meta name="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#00CCFF" text="#000000" link="#FFFF00" vlink="#CC0099" >
<div align="center">
  <h1>Dies ist eine <i>Ueberschrift 1</i></h1>
</div>
<ul>
  <li> Dies hier ist nun eine Unordered List &lt;ul&gt;</li>
  <li>Ein Zeilenumbruch wird mit &lt;br&gt; <br>
    eingegeben </li>
  <li>Die <font size="5">Textgr&ouml;sse</font> wird mit &lt;font size = &quot;1-7&quot;&gt;</li>
  <li><b>Fett &lt;b&gt;...&lt;/b&gt;</b></li>
  <li><i>Kursiv &lt;i&gt;...&lt;/i&gt;</i></li>
  <li><font color="#FF0000">Textfarbe &lt;font color=&quot;#??????&gt; Jede Farbe
    hat einen eigenen Hexadezimalen-Code. Dieser wird hierzu ben&ouml;tigt</font></li>
</ul>
<p>Ein <a href="www.abcd.net">Link</a> sieht dann so aus und wird mit &lt;a href=&quot;www.abcd.net&quot;&gt;
  Die Farbe des Links wurde in der Zeile &lt;body&gt; unmittelbar nach dem Head
  definiert.</p>
<p>Eine <a href="mailto:abcd@efg.hi">E-mail</a> Adresse wird praktisch identisch
  eingef&uuml;hrt &lt;a href=&quot;mailto:abcd@efg.hi&gt;</p>
<p>Ein Bild einf&uuml;gen ist ebenfalls ziemlich simpel. Dies geschieht mit &lt;img
  src=&quot;file:&quot;genauer Pfad&quot; width=&quot;breite&quot; height=&quot;h&ouml;he&quot;&gt;</p>
<p><img src="file:///C|/Pfad/Name.jpg" width="290" height="216"></p>
<p>Eine sehr einfache Tabelle wird wie folgt initialisiert.</p>
<table width="75%" border="1">
  <tr>
    <td>qwe</td>
    <td>rtz</td>
    <td>uio</td>
  </tr>
  <tr>
    <td>pas</td>
    <td>dfg</td>
    <td>hjk</td>
  </tr>
</table>
</body>
</html>
```

Fig. 4 Der HTML-Code erstellt mit Macromedias Dreamweaver 4 Evaluation

4 Zusammenfassung/Schlussfolgerungen

HTML hat bestimmt zum rasanten Wachstum des Webs beigetragen. Die heutige Version ist im Grossen und Ganzen „komplett“. Mängel sind vor allem für professionelle Webpublisher sichtbar. HTML erlaubt keine Individualität bei den verschiedenen Funktionen. Wobei es sich hier um kleinere Details handelt. Die Zukunft von HTML heisst sehr wahrscheinlich XML (e**X**stensible **M**arkup **L**anguage). XML bietet die Möglichkeit individuelle DTDs zu erzeugen. Diese können dann als Erweiterung zu HTML verwendet werden. Zukünftige HTML-Versionen werden vermutlich mit XML definiert werden.
Den WYSIWYG-Editoren stehen wohl auch rosige Zeiten bevor, da XML nicht so einfach zu lesen ist wie HTML. Mit Hilfe dieser Editoren ist es möglich ohne jegliche HTML-Kenntnisse eine doch ansprechende Seite zu kreieren. Den Code erzeugt das Programm selbst aufgrund der Gestaltung.
CSS ist ebenfalls ein „Stylingstool“ das in Zukunft noch weiterentwickelt wird. Sowie DHTML (Dynamic-HTML) welches eine Kombination von mehreren Techniken wie HTML, CSS, Scripting und objektorienierte Programmierung.
Für graphisch anspruchsvolle Seiten mit Animationen und Effekten wird heute oft Flash verwendet. Der Nachteil ist vor allem die Dateigrösse. Da man aber davon ausgehen muss, dass Hochgeschwindigkeits-Internetzugänge irgendwann zum Standard gehören werden, ist auch hier noch einiges an Potential vorhanden.

5 Quellenangaben

1. E.Wilde: World Wide Web – Technische Grundlagen; Springer Verlag, 1999 Berlin, Seiten 191-249
2. www.selfhtml.org Ein sehr ausführliches deutsches Tutorial über HTML, welches auch auf dem eigenen Rechner installiert werden kann.
3. www.htmlhelp.com Ebenfalls eine gutes Tutorial jedoch in Englisch.
4. www.w3.org Homepage des W3C mit sämtlichen Recommendations

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Die Datenstrukturierungssprache XML

Marius Staub
staubm@ee.ethz.ch
11. Januar 2002

1 Die Grundlagen von XML

In diesem Kapitel werden kurz die wichtigsten Fragen rund um XML beantwortet. Die einzelnen Themen werden anschliessend noch genauer behandelt.

1.1 Was ist XML?

Die Abkürzung XML steht für „eXtensible Markup Language“. XML ist eine Datenstrukturierungssprache, die zur Beschreibung der *logischen Struktur* von Informationen dient. Ganz im Gegensatz zu HTML, wo nur etwas über das Layout des Textes mitgeteilt wird. Was das bedeutet, möchte ich kurz in einem Beispiel veranschaulichen. Zuerst die HTML-Version:

```
<p>
<b>Peter </b>
<em>443774</em>
<br>
<b>Rolf </b>
rolf@somewhere.com
</p>
```

Dieser Code bewirkt im Browser eine Darstellung wie

Peter *443774*
Rolf rolf@somewhere.com

Der Browser hat allerdings keine Ahnung, dass es sich bei „Peter“ um einen Namen handelt. Die Tags in den eckigen Klammern sagen nur etwas über die visuelle Darstellung der Textzeilen aus; zum Beispiel, dass die Telefonnummer kursiv gedruckt werden soll.
Das selbe Beispiel nun in XML:

```
<personList>
  <person>
    <name>Peter </name>
    <phone>443774</phone>
  </person>
  <person>
    <name>Rolf </name>
    <email>rolf@somewhere.com</email>
  </person>
</personList>
```

Die Informationen im HTML und im XML – Code sind genau die gleichen. In der XML-Version sind die Daten aber strukturiert und ein geeigneter Browser „versteht“ nun, dass mit „Peter“ ein Namen gemeint ist. Mit einem Style-Sheet könnte man nun den Browser dazu bringen, den XML-Code genau gleich wie den HTML-Code darzustellen, doch dazu später mehr.

1.2 Wozu wurde XML entwickelt?

Das Entwickeln und Einführen einer neuen (Web)Sprache ist kein einfaches Unterfangen. Warum also der ganze Aufwand, wenn man ja schon HTML hat? Das Problem an HTML ist seine Beschränktheit. Mit HTML kann man praktisch nur die *Darstellung* der Daten im Browser beeinflussen, nicht aber deren logische Struktur. Zudem ist HTML ein riesiges Flickwerk geworden, wo jeder Hersteller seine eigenen Standards definierte. Darum entschied man sich, mit XML einen konsequenten Neuanfang zu machen.

XML ist (wie HTML übrigens auch) eine Teilmenge der extrem komplexen Markup-Sprache SGML. Man könnte XML als einen einfachen Dialekt und SGML als die Muttersprache bezeichnen. XML stellt die wichtigsten Eigenschaften von SGML zur Verfügung und man ist dazu übergegangen, HTML mit Hilfe von XML zu beschreiben. Das Ergebins davon ist XHTML, das zu HTML kompatibel ist.

Ein weiterer Vorteil von XML ist seine Verfügbarkeit auf allen Plattformen. Da es sich bei XML eigentlich nur um Text-Files handelt, können sie von praktisch allen (Internet)Usern gelesen werden. So wird der Datenaustausch zwischen verschiedenen Anwendungen erheblich vereinfacht.

2 Aufbau eines XML-Dokuments

Im diesem Abschnitt werden die wichtigsten Grundzüge eines XML-Dokuments beschreiben, sowie die Bestandteile, die nötig sind, um die Daten in einem Browser darzustellen.

2.1 Grundzüge der XML-Syntax

Die XML-Spezifikation des w3-Konsortiums legt genau fest, wie ein XML-Dokument aussehen muss. In XML muss - im Gegensatz zu HTML[1] - zu jedem Start-Tag auch ein End-Tag vorhanden sein, sonst ist das Dokument fehlerhaft. Die XML-Definitionen sind allgemein strenger als die von HTML. Der Grund ist der, dass mit XML Daten weitergegeben werden und fehlerhafte Datenstrukturen führen fast unvermeidlich zu Datenverlust.

2.2 Die DTDs

Die DTD (Document Type Definition) definiert die XML-Tags. Man kann die DTD entweder direkt in das XML-Dokument einbinden oder in einer separaten Datei speichern. In HTML gibt es nur eine einzige[2] DTD, welche nicht verändert werden kann. Dort sind die bekannten HTML-Tags wie `<p>`, `<br>` usw. definiert. Zum besseren Verständnis soll dazu wieder ein Beispiel dienen. Der folgende XML-Code beeinhaltet eine DTD und eine kleine „Anwendung“ dieser Tags:

```
<?xml version="1.0"?>

<!DOCTYPE personList [
      <!ELEMENT personList (person+) >
      <!ELEMENT person (name, (phone | email)*)+ >
      <!ELEMENT name (#PCDATA) >
      <!ELEMENT phone (#PCDATA) >
      <!ELEMENT email (#PCDATA) >
]>

<personList>
<person><name>Peter</name><phone>443774</phone></person>
<person><name>Rolf</name><email>rolf@somewhere.com</email></person>
</personList>
```

[1] Eigentlich wäre auch in HTML zu jedem Tag ein End-Tag nötig. Da aber die allermeisten Web-Seiten nicht sauber geschrieben sind, wurden die Browser so programmiert, dass sie die Bedeutung von falschen oder fehlenden Tags versuchen zu erraten.

[2] In Wirklichkeit sind es deren drei. Man kann sie aber zur Vereinfachung als eine einzige betrachten.

Die erste Zeile enthält die Information über die verwendete XML-Version. In der zweiten Zeile beginnt die DTD. Hier werden die Tags definiert, die nachher im Dokument verwendet werden können. Man könnte die DTD auch als Sammlung von Datentyp-Definitionen betrachten. Zum Vergleich: In C kann man benutzerdefinierte Datentypen mit struct erzeugen (vgl. Klassen in Java), während sie hier in der DTD definiert werden..

Das erste Element, die „Wurzel“, heisst hier `personList`. Dieses Element besteht wiederum aus Elementen mit dem Namen `person` und ein `person`-Element besteht aus einem `name` und entweder einem `phone` oder `email`. Wird die DTD extern, das heisst in einer separaten Datei,, definiert, so muss diese Datei den gleichen Namen haben wie die Wurzel.

Im letzten Block werden zwei Elemente vom „Typ“ `person` deklariert und die Elemente mit Inhalt gefüllt. Diese sind vergleichbar mit der Instanz eines Structs/Klasse. Man beachte, dass zu jedem Start-Tag immer ein entsprechendes End-Tag gesetzt wurde.

2.3 Die Darstellung im Browser

Wenn man den obigen Code in einer Datei speichern würde und dann mit einem XML-fähigen Browser anschauen würde, wäre man wahrscheinlich über die Darstellung erstaunt, denn der Browser zeigt einfach den Quellcode in anderen Farben an.

Um zu erreichen, dass der Browser nur die Daten, nicht aber die Tags anzeigt, muss man noch ein sogenanntes Style Sheet schreiben. Das Style Sheet enthält die nötigen Layout-Angaben für den Browser, damit dieser die Informationen gut lesbar ausgeben kann. Dazu muss die folgende Zeile in den obigen Code eingefügt werden (nach der 1. Zeile):

```
<?xml-stylesheet href="test.css" type="text/css"?>
```

Dieser Code teilt dem Browser mit, dass sich in der Datei test.css ein CSS-Stylesheet befindet, das zur Darstellung der Daten benutzt werden soll. In der Datei test.css könnte dann zum Beilspiel folgendes stehen:

```
person { display : block; margin-botton: 5mm }
name { display : list-item; font-weight: bold }
phone { display : list-item; font-style : italic }
email { display : list-item; text-decoration : underline }
```

Das Ergebnis sieht dann im Browser folgendermassen aus:

Peter *443774*
Rolf rolf@somewhere.com

Anstatt den CSS-Stylesheets könnte man auch ein XSL-Stylesheet verwenden. Die XSL-Stylesheets sind vom W3C dazu eingeführt worden, um den XML-Dokumenten die Layout-Informationen mitzuliefern. Allerdings sind XSL-Stylesheets recht kompliziert und für viele Anwendungen bieten die einfacheren CSS-Stylesheets ausreichende Möglichkeiten.

Die folgende Abbildung verdeutlicht noch einmal, wie ein Browser ein XML-Dokument darstellt.

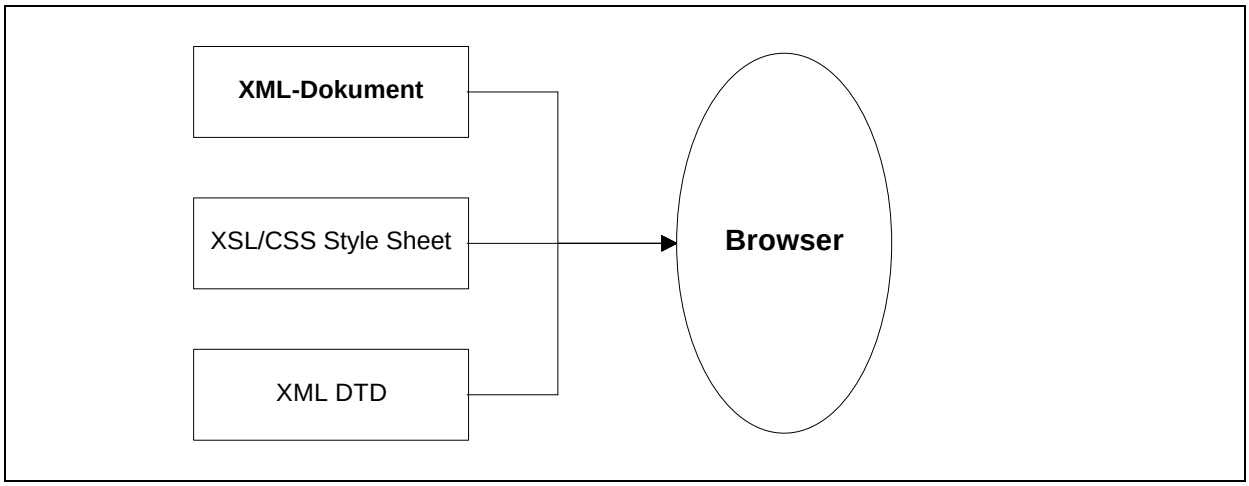


Fig. 1 Darstellung eines XML-Dokumentes im Browser

2.4 Weitere XML-Anwendungen

Bis jetzt betrachteten wir XML vor allem in seiner Rolle als neue Websprache. Für Homepages, die keine Datenbanken beinhalten, ist der Aufwand zu gross, diese mit XML darzustellen. Hier sind die Möglichkeiten von HTML völlig ausreichend. XML wird dort interessant, wo die Informationen wechseln. Das einfachste Beispiel sind News-Seiten im Internet, wo die News-Informationen als XML-Dokumente übermittelt werden. Natürlich gibt es noch viele andere Möglichkeiten, eine News-Seite aufzubauen (z. B. mit ASP)

Die grosse Stärke von XML liegt mehr darin, dass Daten zwischen grossen Datenbanken ausgetauscht werden können. Man kann mit XML auch neue, eigene Markup-Sprachen definieren. So führte ein Verband von E-Commerce Unternehmen einen eigenen Standard ein, um die Abwicklung von Transaktionen und Informationsaustausch zwischen Käufer und Verkäufer zu vereinfachen.

Das W3C führte auch eine Beschreibungssprache für Mathematik ein,die auf XML basiert (MathML). Es gibt auch zahlreiche andere Bestrebungen, zum Beispiel für chemische Formeln, Beschreibungssprachen einzuführen. Bis anhin wurden mathematische Formeln meistens als GIFs in Homepages eingefügt.

3 Konvertieren und Verknüpfen

Bei der Einführung eines neuen Datenformates muss sichergestellt werden, dass der Austausch mit anderen, auch schon bestehenden Datenformaten ohne Informationsverlust funktioniert. In diesem Abschnitt wird auch das Verlinken von XML-Dokumenten mit nicht-XML basierten Ressourcen beleuchtet.

3.1 Konvertieren von und nach XML

3.1.1 XML und HTML

Ein Spezialfall stellt die Konvertierung von HTML nach XML und umgekehrt dar. Die Umwandlung von XML nach HTML ist relativ einfach, weil XML-Dokumente wohlgeformt sind. Es müssen nur einige Regeln definiert werden, nach denen die XML-Tags in HTML-Tags umgewandelt werden. Dazu wurde vom W3C die Entwicklung von XSLT veranlasst. XSLT ist eine mächtige Sprache, mit welcher die Regeln zur Transformation von XML-Dokumenten beschrieben werden können.

Das Umwandeln von HTML in XML ist schwieriger, weil 95% aller HTML-Dateien fehlerhaft sind und nur dank toleranter Browser gelesen werden können. In einem ersten Schritt müssen darum HTML-Dokumente um die fehlenden End-Tags ergänzt werden und falls nötig auch richtig verschachtelt werden. Der nächste Schritt besteht in der Entwicklung einer XML DTD für HTML-Dokumente. Dies entspricht dem Projekt XHTML.

3.1.2 XML und andere Datenfomate

Der wichtigste Punkt bei der Umwandlung von z.B. firmeninternen Datenformaten in XML ist die Erstellung einer geeigneten DTD, welche die Struktur des anderen Datenformats korrekt wiederspiegelt. Die Konvertierung sollte auch möglichst automatisch abgewickelt werden können, ohne dass die Daten nachher noch „von Hand“ korrigiert werden müssen.

Ein Anwendungsbeispiel für die Konvertierung nach XML ist das Publishing. Wenn die Daten im Intranet oder Internet veröffentlicht werden sollen, geschieht dies am einfachsten, wenn die Daten in XML-Format vorliegen. Man kann bei einer solchen Umwandlung auch gleich dafür sorgen, dass nur diejenigen Daten in der XML-Version vorhanden sind, die dann auch für die Veröffentlichung gedacht sind.

Für die Umwandlung von XML in andere Markup-Sprachen wird die weiter oben schon erwähnte Transformationssprache XSLT verwendet. Manchmal ist auch eine Konvertierung von XML nach XML nötig, wieder etwa dann, wenn nur gewisse Daten aus einem XML-Dokument für die Veröffentlichun gedacht sind.

3.2 Links mit XML

In HTML kennt man nur eine sehr einfache Art von Links. HTML-Hyperlinks funktionieren nur in eine Richtung; man kann nachher nicht mehr zurückspringen. Diese Links nennt man „unidirektionale Links“. Für XML wurde ein spezielles Umfeld namens XLink definiert, welches viel mehr Möglichkeiten für Links bietet. So sind in XML auch bidirektionale Links möglich, oder auch Links, bei denen man zwischen verschiedenen Zielen wählen kann. XLink stellt auch Methoden zum Erstellen von Links aus schreibgeschützten Dokumenten heraus zur Verfügung.

4 Zusammenfassung

Seit der Veröffentlichung im Jahr 1998 durch das W3C hat XML immer mehr Beachtung bekommen. Das liegt nicht zuletzt auch daran, dass die aktuellen Browser XML immer besser unterstützen. Zudem haben die grossen Firmen der IT-Industrie XML weitgehend akzeptiert und verwenden diese Sprache in ihren eigenen Projekten.

XML wird aber HTML kaum je ersetzten oder verdrängen, denn für viele Anwendungen bietet HTML vollig ausreichende Möglichkeiten. Wie sich XHTML im alltäglichen Internet-Gebrauch durchsetzen kann, wird die Zukunft zeigen.

Die Plattformunabhängigkeit, Einfachheit und breite Unterstützung der XML-Technologie werden in den kommenden Jahren sicher dazu führen, dass XML noch mehr Anwender findet. Sei es im Web oder sei es in sonstigen Anwendungen.

5 Anhang

5.1 Quellenangaben

1. E. Wilde: „World Wide Web – Technische Grundlagen“; Springer Verlag, Berlin 1999
2. W3C: Deutsche Übersetzung von XML 1.0 Recommendation; http://www.mintert.com/xml/trans/REC-xml-19980210-de.html
3. The XML FAQ; http://www.ucc.ie/xml/
4. XML - Extensible Markup Language; http://www.boku.ac.at/htmleinf/xmlkurz.html

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

IP - TELEPHONIE
VoIP

Reto Felix
rfelix@ee.ethz.ch
11. Januar 2002

1 Einführung

Über das Internet zu telefonieren, ist nach wie vor etwas für Enthusiasten, es eignet sich nur bedingt für die geschäftliche Kommunikation. Es muss zwischen lokalen Netzen und dem Internet unterschieden werden. Damit die IP-Telefonie eine hohe Gesprächsqualität bieten kann, erfordert diese ein Netz, das gewisse Quality of Service Merkmale aufweist. Diese Merkmale können derzeit nur in einem modernen Intranet bereitgestellt werden. Das Internet (WWW) bietet heute weder eine konstante Bandbreite, noch können Prioritäten für bestimmten Datenverkehr vergeben werden.

Der Begriff »Voice over IP« (VoIP) bezeichnet eine Technik, mit der IP-Netzwerke Sprache übertragen – im Allgemeinen schlechter als die herkömmlichen öffentlichen Telefonnetze. Das Web transportiert IP-Pakete über verschiedene Wege mit unterschiedlich langen Laufzeiten. Der Anwender hat keine Steuerungsmöglichkeiten, die Sprachqualität ist deshalb in der Regel mangelhaft.

Im lokalen Netzwerk dagegen hat es der Administrator in der Hand, die Netzkomponenten so zu konfigurieren, dass sie die erforderliche Dienstgüte bereitstellen. Für Unternehmensnetze ist die IP-Telefonie deshalb bereits heute eine Alternative. Richtig implementiert, kann sie sich bei der Sprachqualität durchaus mit ISDN messen.

Ein häufig genannter Vorteil von VoIP ist die Einsparung der Telefongebühren. Dies wird allerdings erst dann richtig zum Tragen kommen, wenn IP-Telefonie auch über das Internet mit hoher Qualität möglich ist. Ein großer Pluspunkt ist, dass mit VoIP nur noch eine IT-Infrastruktur gepflegt werden muss.

1.1 Entwicklung der IP-Telephonie

Als vor mehr als 100 Jahren das Telefon erfunden wurde, war wohl niemandem bewusst welche Bedeutung dieses Gerät für die Kommunikation erlangen würde. Trotz der vermeintlich vielen Mängel. Seitdem wird das Telefon und die dazu gehörende Technik ständig weiterentwickelt.

Zunächst wurde unsere Sprache analog übertragen. Die Frequenz der Sprache (300 - 3000 Hz) wurde kontinuierlich über ein zweiadriges Kupferkabel gesendet. Äußere Störeinflüsse verursachten ein Rauschen oder Knistern bei der Sprachübertragung. In den letzen Jahren wurde die Sprachübertragung digitalisiert. ISDN (Intergrated Service Digital Network) tastet die Schallschwingungen ab und wandelt diese in binäre Datenströme um. Jede Sprachschwingung wird 8000 mal pro Sekunde getastet und mit 8 bit binär codiert. 8000/s x 8 bits = 64 kbit/s.

Parallel zum Telefonnetz entwickelte sich seit dem Ende der 70er Jahre eine andere Technik, die zwar keine Sprache aber dafür Daten in hoher Geschwindigkeit übertragen konnte. Was lag also näher als der Gedanke ein digitales Telefon-Netz und ein paralleles Datennetz miteinander zu verschmelzen. Daraus entstand der Anspruch die »alte« Telefonwelt zu modernisieren und fit für die moderne Datentransportwelt zu machen - die Entwicklung der IP-Telephonie begann.

Die ersten Lösungen für die Internet-Telefonie waren noch sehr primitiv. Die Technik arbeitete zunächst nur halbdulex.(d.h. auf jeder Seite konnte nur entweder gesendet, oder empfangen werden.) Die Verbindung rauschte und die Daten kamen mit grosser Verzögerung beim Empfänger an. Aber schon 1996 sprach man bereits von der Abschaffung des herkömmlichen Telefons und es wurde damals das Ende der Telefongesellschaften vorausgesagt. Wie sich dann aber gezeigt hat waren diese Phantasien etwas voreilig. Telecom Gesellschaften sind heute auch Anbieter von Datenkommunikation. Bis heute hat sich die IP-Telephonie noch nicht hundertprozentig durchzusetzen vermocht. Dies weil die Übertragung von Audiodaten in Echtzeit über Datennetze immer noch gewisse Nachteile gegenüber des Telefons mit sich bringt.

2 Technik

2.1 Normen H.323 / SIP

2.1.1 H.323

Im Jahr 1996 verabschiedete der »Telecommunication Standardization Sector« der ITU (International Telecomunications Union) die Empfehlung H.323. Sie sollte Multimedia-Kommunikation über lokale Netzwerke ermöglichen, vor allem den Transport von Audiodaten in Echtzeit, sowie optional den Transfer von Videos und Daten bei Punkt-zu-Punkt- und Multipoint-Konferenzen. Zwei Jahre später folgte eine zweite Version, die für alle IP-Netze ausgelegt war: vom LAN bis hin zum Internet.

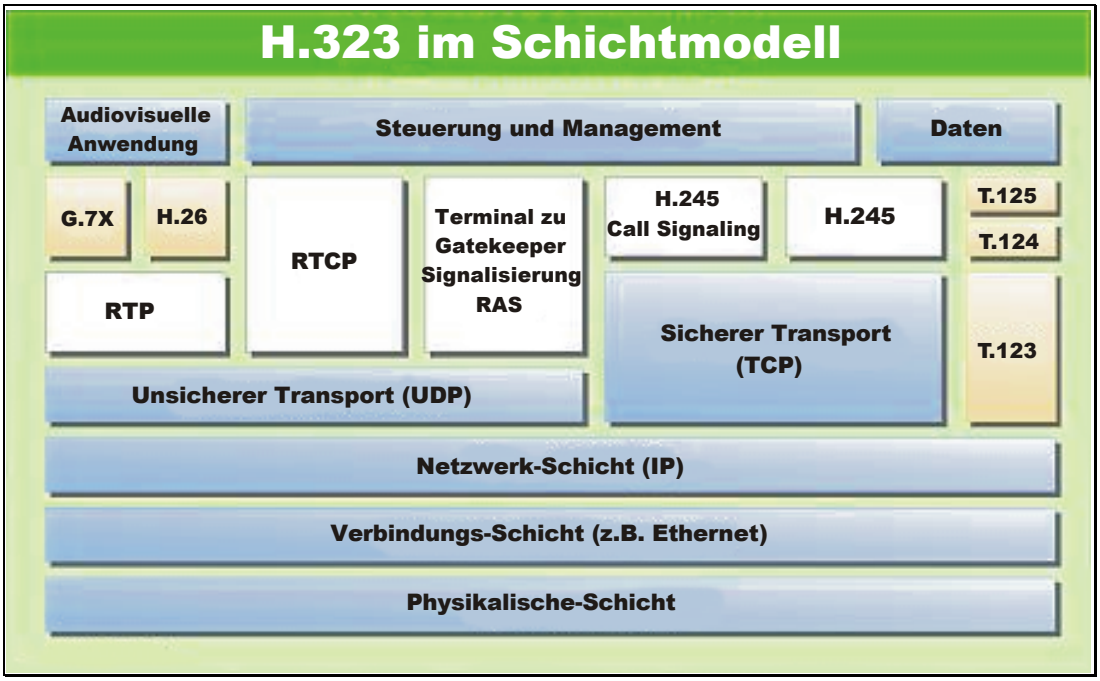


Fig. 1 H.323 Schichtmodell

H.323 beschreibt die Komponenten und Verfahren, die zum Telefonieren über paketgestützte Netze notwendig sind. Zu den Komponenten zählen Terminals, Gateways, Gatekeeper, Multipoint-Controller und -Prozessoren sowie Multipoint Control Units (MCU). Die Mechanismen regeln unter anderem, auf welche Weise Verbindungen aufgebaut werden und sich die Kommunikation überwachen lässt. Weil dabei mehrere Protokolle mitspielen, etwa das Real-Time Transport Protocol oder ITU-Protokolle wie H.225.0 und H.245, wird H.323 auch als Protokollfamilie bezeichnet. Ob sich H.323 langfristig durchsetzen kann, ist fraglich. Denn mit dem Session Initiation Protocol (SIP) wird derzeit eine einfacher zu implementierende Alternative für die Signalisierung in VoIP-Netzen entwickelt.

2.1.2 SIP (Session Initiation Protocol)

Das Session Initiation Protocol SIP ist ein einfaches Management- Protokoll für Internet Konferenzen. Es stellt unter anderem Mechanismen für Einladungen, Rufweiterleitung, Adressauflösung,

Parameterverhandlung und Authentifizierung zur Verfügung. Das Programm ist Teil einer umfassenden Multimedia-Architektur, welche als Alternative zu H.323 entwickelt wurde.

SIP ist dabei sowohl vom Transportmedium als auch vom verwendeten Konferenz-Kontroll-Protokoll unabhängig. Es implementiert selbst Strategien um mit verlorenen Nachrichten umzugehen. SIP ist damit nicht auf die Zuverlässigkeit des Transportmediums angewiesen. Ebenso ist das SIP-Protokoll nicht auf verbindungslose bzw. -orienterte Protokolle angewiesen. Es arbeitet gleichermaßen über TCP wie auch über UDP. Seine Modulstruktur erlaubt die Zusammenarbeit mit H.323.

2.2 Aufbau eines H.323 Systems

H.323-Systeme bestehen im Wesentlichen aus Gatekeeper, Gateway und den Endgeräten. Der Gatekeeper übernimmt die Zuordnung von IP-Adressen zu Telefonnummern sowie die Zugriffskontrolle und verwaltet die im LAN verfügbare Bandbreite. Das Gateway stellt die Verbindung her zwischen dem VoIP-Netz und dem öffentlichen Telefonnetz. Es baut Gespräche auf und wieder ab, komprimiert die Sprache und packt sie in IP-Pakete ein. Als Endgeräte sind IP-Telefone mit Ethernet-Anschluss und Software-Clients erhältlich. Sie erfüllen die gleichen Aufgaben wie das Gateway. Sobald eine Verbindung aufgebaut ist, tauschen die Endgeräte die Sprachdaten direkt miteinander aus.

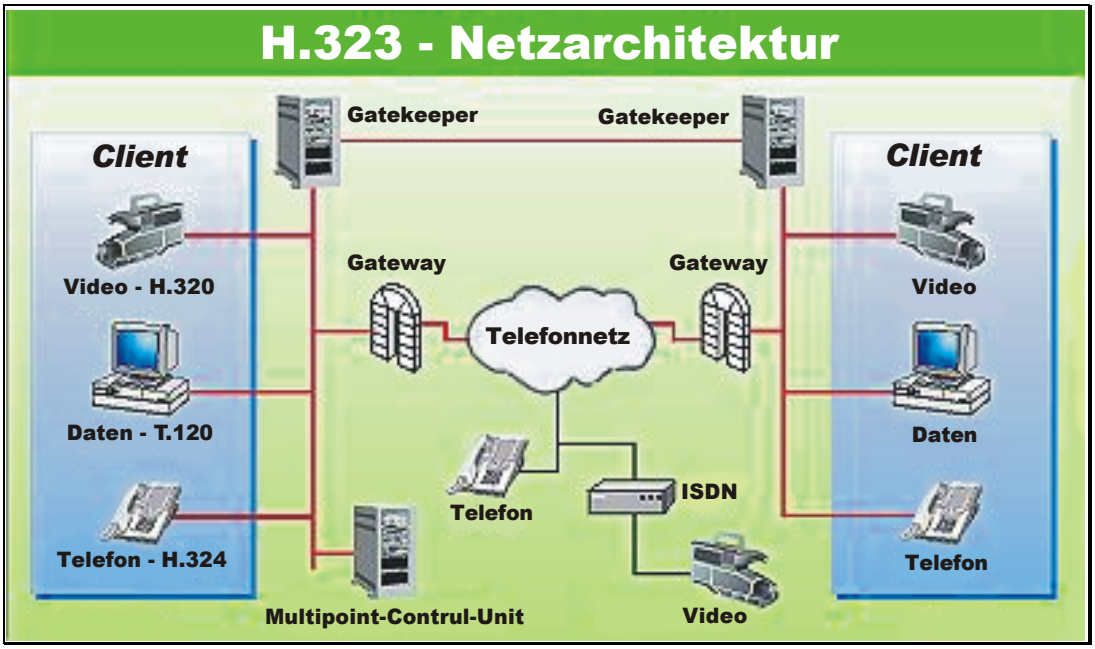


Fig. 2 H.323-Netzwerkmodul der ITU

2.2.1 Gatekeeper

Call-Control-Funktionen spielen für jedes Netzwerk mit Sprachübertragung eine wichtige Rolle. Viele dieser Funktionen werden von komplexen Datenbank-Managementsystemen übernommen. Hierzu gehört unter anderem die Gesprächsabrechnung, die Adressumsetzung, das Routing und das Bandbreiten-Management. Der Gatekeeper ist ein Hochleistungsrechner mit High-Availability-Merkmalen.
Der Gatekeeper hat folgende Funktionen:

- Die Adressumsetzung ordnet an jedem Endpunkt einer Alias-Adresse eine Netzwerkadresse zu und umgekehrt.

- Die Zugangskontrolle begrenzt den Zugang je nach verfügbarer Netzwerkbandbreite und den Zugriffsrechten der Anwender.
- Die Verwaltung der Bandbreite im Netz sorgt für eine optimale Kommunikationsqualität zwischen den Endpunkten.
- Das Zonenmanagement legt fest, welche H.323-Endpunkte von einem Gatekeeper jeweils kontrolliert werden. Jeder Endpunkt beantragt eine Registrierung bei diesem Modul und profitiert damit von allen Gatekeeper-Funktionen.

2.2.2 Gateway

Gateways stellen die Verbindung zwischen der traditionellen Technik und der digitalen Welt der Internet-Telefonie her, so dass die Benutzer beider Techniken miteinander kommunizieren können. Die Hauptfunktion des Gateways besteht darin, Übersetzerdienste für das »virtuelle« Endgerät und für die verschiedenen Übertragungsformate, Kommunikationsprozeduren und Audio-Codecs zu leisten.

Das Gateway ist eine Zweiwege-Schnittstelle zwischen dem Telefonnetz und dem IP-Netzwerk. Es kann daher entfallen, wenn keine Verbindung zum regulären Telefonnetz erforderlich ist; zum Beispiel in einem unternehmenseigenen LAN.

2.2.3 Mutlipoint-Control-Unit (MCU)

Die Multipoint-Control-Unit (MCU) ist diejenige Komponente der VoIP-Architektur, die den Benutzern erweiterte Möglichkeiten der Zusammenarbeit durch Telefon- oder Videokonferenzen bietet. Die MCU wirkt als Endpunkt im Netzwerk und ermöglicht Konferenzschaltungen mit drei oder mehr H.323-Endgeräten. Sie besteht aus einem Multipoint-Controller und enthält auf Wunsch einen Multipoint-Prozessor. Der Gatekeeper kann die MCU explizit aktivieren, wenn zwei oder mehr Endpunkte an der selben Konferenzschaltung teilnehmen.

Der Multipoint-Prozessor kombiniert und vermittelt alle Audio/Video- und Datensignale zwischen den H.323-Endpunkten.

2.2.4 Endgeräte

Ein VoIP-Endgerät oder »Client« ist das Element, von dem die Initiative zum Herstellen von Live-Kommunikationsverbindungen ausgeht. Der heute gebräuchlichste Endgerätetyp ist ein Softwarepaket wie »Netmeeting« von Microsoft, das auf einem PC läuft. Mit Hilfe dieser Schnittstelle können die Anwender Telefongespräche über das Internet führen. Im Hintergrund sorgt die Software für den Aufbau und das Trennen der Verbindungen sowie für das Kodieren und Dekodieren der übertragenen Sprache.

Das Endgerät ermöglicht in Echtzeit eine bidirektionale Sprach-, Video- oder Datenkommunikation mit anderen VoIP-Endgeräten. Es kommuniziert mit den VoIP-Gateways über das ITU-Protokoll H.245 (International Telecommunications Union) für die Verbindungssteuerung und über die ETSI-Standards Q.931 (European Telecommunications Standards Institute) für den Verbindungsaufbau und RAS (Remote Access Service) für die Registrierung und Administration im lokalen Gatekeeper. Dabei arbeitet es mit verschiedenen Elementen zusammen, beispielsweise mit einem regulären Telefonapparat oder dem Mikrofon/Lautsprecher auf der Audioseite beziehungsweise einer Kamera/Monitor-Kombination für die Videoübertragung.

3 Anwendung von VoIP

3.1 Vergleich mit konventioneller Telephonie

Die Kostenersparnis durch Einsparen von Telefongebühren spielt heute für Untenehmen nur noch eine untergeordnete Rolle bei der Entscheidung auf IP-Telefonie umzustellen. Hier zeigt die Liberalisierung des Telecom-Marktes und der damit verbundene Preiszerfall Wirkung. Heute geht es beim Umstieg auf IP-Telefonie vor allem um folgendes.

Aus wirtschaftlicher Sicht sind getrennte Infrastrukturen für Datenkommunikation und Telefonie teurer Luxus. Es gibt mittlerweile Lösungen, die Sprachübertragungen in IP-Netze integrieren. Sprach- und Dateninfrastrukturen unterscheiden sich in erster Linie in den verschiedenen Anschlussschnittstellen der Telefone und Netzwerkkarten sowie der Netzeigenschaften. Das Telefonnetz arbeitet verbindungsorientiert und stellt für jede Verbindung einen leitungsvermittelten Sprachkanal zur Verfügung. Der Vorteil liegt in der garantierten Bandbreite, die bei einer ISDN-Verbindung konstant 64 kBit/s pro Kanal beträgt. Nachteilig ist, dass diese Bandbreite für eine reine Sprachverbindung gar nicht ausgenutzt wird.

Diese Probleme bestehen in Datennetzwerken nicht. Sie transportieren Pakete verschiedener Rechner über einen gemeinsamen Leitungsweg und teilen sich die verfügbaren Kapazitäten. Es handelt sich hierbei um eine paketvermittelte Kommunikationslösung, die verbindungslos arbeitet. Während dies für die in der Regel zeitunkritische Datenkommunikation eine optimale Lösung darstellt, birgt das Übermittlungsverfahren für Sprachdaten Probleme. Der Transport der Daten von verschiedenen Rechnern über ein gemeinsames Medium hat zwangsläufig zur Folge, dass nicht alle Computer gleichzeitig senden können. Es entstehen Wartezeiten, deren Dauer zufällig ist. Der Transport der Pakete innerhalb des Datennetzes erfolgt somit mit unterschiedlichen Laufzeiten.

Paketlaufzeit und Verlustrate sind für VoIP so wichtig, weil zu spät eintreffende Sprachpakete verloren gehen. Subjektiv störend können bereits Laufzeitschwankungen ab 25 ms sein. Negativ wirken sich auch unterschiedlich lange Paketlaufzeiten aus.

3.1.1 Vorteile der IP-Telephonie

- Nur noch ein gemeinsames Datennetz.
- Bessere Auslastung des Transportnetzes.
- Das vorhandene Datenkabel zum Anschluss des Arbeitsplatzrechners wird auch für ein zusätzliches IP-Telefon mitgenutzt.
- Der IP-Telefonie-Server ist frei skalierbar. Ungeachtet wieviele IP-Telefon-Appartate verwaltet werden, es ist keine teure Aufrüstung von Baugruppen erforderlich.
- Geringerer Administrationsaufwand.
- In lokalen Netzen bereits heute ISDN Qualität.

3.1.2 Nachteile der IP-Telephonie

- Heutige Internet-Router arbeiten in der Regel langsamer als Telefon-Switches. Zudem routen sie Datenpakete häufig auf grossem Umweg ans Ziel. Dies beeinträchtigt Echtzeitanwendungen.
- Die Tonqualität hängt von der momentanen Netzbelastung ab. Generell erreicht sie jedoch nicht die gewohnte Qualität konventioneller Telefonie
- Firewalls erschweren die IP-Telefonie oder machen sie in Einzelfällen sogar unmöglich.
- Bisher ist die IP-Telephonie immer noch unzureichend standardisiert, was die Kommunikations-möglichkeiten entscheidend einschränkt. Sofern bei der Variante PC zu PC der Angerufene nicht die selbe Telefonsoftware wie der Anrufer benutzt, ist er i.a. auch nicht erreichbar.
- Durch das Reservieren von Bandbreiten und durch bevorzugtes Routen von Sprachpaketen anstelle von Datenpaketen wird zwar die Qualität des Dienstes selbst zu Zeiten höchster Netzbelastung sichergestellt, gleichzeitig jedoch auch die eigentlich angestrebte Kostensenkung in Frage gestellt.
- Zuweilen werden Sicherheitsbedenken geäußert, dass Telefonate über das öffentliche Internet möglicherweise leichter abgehört werden können.

3.1.3 Zukunft der IP-Telephonie

Die Internet-Telephonie erhält gerade einen neuen Schub durch die nächste Generation des IP-Protokolls IPv6. Mit der von Ipv6 gebotenen Quality of Service-Unterstützung lassen sich einige der oben genannten Nachteile beheben. Im Vergleich zur Vorgängerversion beinhaltet die Fortschreibung einen nahezu unendlich erweiterten Adressraum, eine Bandbreitenreservierung und durch eine eingebettete Verschlüsselung erstmals das Potential für gesicherte Gespräche. Auch wenn viele Anbieter schon angetreten sind, den großen Tele-kommunikationsfirmen mit sehr niedrigen Gebühren die Kunden abzujagen, so stecken sowohl VoIP als auch IPv6 noch am Anfang ihrer Implementierung. Beide Technologien werden wahrscheinlich zusammen auf Erfolgskurs gehen.

3.2 Fazit

Ein optimal konfiguriertes Netzwerk ist die wichtigste Voraussetzung, um mit VoIP eine hohe Sprachqualität zu erreichen. Für eine wirksame Priorisierung des Sprachverkehrs müssen die einzelnen Netzkomponenten sorgfältig aufeinander abgestimmt werden. Dann lässt sich mit den heute verfügbaren VoIP-Lösungen innerhalb von lokalen Netzen sogar ISDN-Qualität erreichen.

Konkurrenzfähig gegenüber dem ISDN- beziehungsweise Telefonnetz ist das Datennetzwerk erst durch internationale Standardisierungen der ITU sowie der Internet Engineering Task Force (IETF) geworden. Die Standards stellen die Basis für moderne IP-Telefone zum direkten Anschluss an das lokale Netzwerk (LAN) sowie für VoIP-Gateways dar.

In aktuellen Projekten arbeiten Experten bereits an Weiterentwicklungen, die vor allem ein besseres Zusammenspiel von paket- und leitungsvermittelten Netzen zum Ziel haben. Dazu sind unter anderem einheitliche Implementierungen und Gateways zu den vorhandenen Telekommunikations-netzen erforderlich, außerdem Mechanismen für die Adressierung, das Reservieren von Ressourcen und die Sicherheit. Darüber hinaus ist zu klären, ob das Internet in seiner jetzigen Form überhaupt für Echtzeitdienste wie Voice over IP tauglich ist. Denn bereits Standard-Anwendungen wie E-Mail, FTP, Telnet oder das World Wide Web bringen das Netz in Spitzenzeiten an den Rand seiner Kapazität.

Quellen-Angaben

1. Jürgen Kuri: Sprache in Päckchen; c't'Heft 10, 1999
2. Alex Kossel: Netzgespräche – Internet Telefonie in der Praxis; c't Heft 10, 1999
3. D. Rizzetto, C. Catania: A VoIP Service Architecture; Internet Computing, June 1999
4. http://www.networkworld.de/artikel/index.cfm?pageid=154
5. http://www.iid.de/informationen/IP-Telefonie
6. http://www.e-online.de/sites/kom/0503131.htm
7. http://www.rznet.de/iptelHistory.html?opendocument&expandview&count=99999

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

SSL, SHTTP
sichere Kommunikation

Robin Elsasser
erobinethz.ee.ch
11. Januar 2001

1 Einleitung

Auf dem Internet werden immer mehr und neue Anwendungen angeboten. Dabei reicht es bei vielen von diesen Produkten nicht, dass die Daten schnell und fehlerfrei an die richtige Adresse geschickt werden können. Die Übertragung der Daten soll auch sicher sein. Um diese Sicherheit gewährleisten zu können, sind folgende Voraussetzungen notwendig:

- Vertraulichkeit
  Nur ein bestimmter Personenkreis darf den Inhalt der Daten erfahren
- Integrität
  Die Daten müssen vor Veränderung geschützt sein
- Authentizität
  Die Herkunft von Daten muss vom Empfänger zweifelsfrei bestimmt werden können
- Nichtabstreitbarkeit
  Die Herkunft von Daten muss auch gegenüber Drittpersonen nachweisbar sein

Natürlich werden zum Teil nur eine oder ein Teil der aufgeführten Punkte verlangt oder erwünscht.

Die herkömmlich auf dem IP basierenden Dienste weisen zwei Schwachstellen auf, welche ein Angreifer leicht ausbeuten kann:

- Die Kommunikation im Netz erfolgt im Klartext. So kann man mit geeigneten Programmen die Datenpakete aufzeichnen und problemlos auswerten. Dazu kommt, dass Benutzername und Passwort zum Beispiel beim Abholen der E-Mail vom Server mittels POP3 unverschlüsselt übertragen werden und abgehört werden können.
- Beim Übertragen von Daten ist es möglich, dass diese auf dem Weg von Sender zu Empfänger von einer Mittelperson abgefangen und manipuliert werden können.

Um Daten nun vor diesen Angriffsmöglichkeiten zu schützen, gibt es grundsätzlich zwei verschiedene Möglichkeiten:

- Verwenden einer sicheren Transportarchitektur
  Bei diesem Szenario wird das Anwendungsprotokoll nicht verändert. Es wird davon ausgegangen, dass die Transportinfrastruktur selbst die Sicherheit bereitstellt, so dass als einzige zu entscheidende Frage übrigbleibt, ob die normale (unsichere) oder die sichere Transportinfrastruktur verwendet werden sollte und wie sich eine die sichere Transportinfrastruktur einsetzende Verbindung herstellen lässt. Diese Art der Sicherheit kann von allen Anwendungen genutzt werden, welche die sichere Transportinfrastruktur erkennen.
- Verwenden eines sicheren Protokolls auf Anwendungsebene
  Im zweiten Fall wird davon ausgegangen, dass die Transportinfrastruktur unsicher ist und aus diesem Grund das Anwendungsprotokoll dahingehend abgeändert wird, dass es selbst über Sicherheitsmerkmale verfügt. Bei diesem Ansatz sind die Anforderungen bezüglich der Transportinfrastruktur wesentlich niedriger, aber die innerhalb der Anwendungen zu erledigenden Arbeiten (das Hinzufügen von Sicherheitsmerkmalen zum Protokoll der Anwendungsebene) um so schwieriger. Darüber hinaus lassen sich die Sicherheitsmerkmale lediglich für eine bestimmte Anwendung verwenden.

2 Kryptographische Verfahren

Die Sicherheitsmassnahmen, die getroffen werden damit Daten in unsicheren Netzen sicher ans Ziel kommen verwenden meist irgendwann eine Art von kryptographischem Verfahren. Deshalb wird an dieser Stelle kurz das Wichtigste erklärt.

2.1 Klassifizierung von Verfahren

Ein schwaches Verfahren erkennt man daran, dass es mit geringer bis mittlerer Rechenkapazität gebrochen werden kann. Der Verdacht auf ein solches Verfahren liegt nahe, wenn der zugrundeliegende Algorithmus geheimgehalten wird. Die Sicherheit eines guten Verfahrens liegt nur in der Geheimhaltung des Schlüssels, der Algorithmus dagegen wird als bekannt vorausgesetzt. Bei einem guten Verfahren kann man deshalb die Schlüssellänge als Mass für die Sicherheit benützen. Hier muss man jedoch sicher zwischen symmetrischen und asymmetrischen Verfahren unterscheiden, da die Prinzipien der Verfahren ganz unterschiedlich sind und gegen asymmetrische Verfahren mit ihren deutlich grösseren Schlssellängen bessere Angriffe als erschöpfendes Durchsuchen existieren.

2.2 Symmetrische Verschlüsselungsverfahren

Dieses Verfahren wird auch als Secret-Key- oder Private-Key-Verfahren bezeichnet und ist dadurch gekennzeichnet, dass sich Sender und Empfänger einen gemeinsamen, notwendigerweise geheim zu haltender Schlüssel teilen. Symmetrische Verfahren unterteilen sich in Blockchiffren und Stromchiffren.

2.2.1 Blockchiffren

Für dieses Prinzip wird der Text in einzelne Blöcke aufgeteilt, welche unabhängig voneinander verschlüsselt werden. Der innere Aufbau ist bei allen Blockchiffren ähnlich und sieht folgendermassen aus: nach einer möglichen initialen Eingangsbehandlung eines Klartextblocks wird eine interne Rundenfunktion mehrfach durchlaufen sowie eine abschliessende Endebehandlung durchgeführt. Die Rundenfunktion zerwürfelt die Bits des Blocks und ersetzt Teilblöcke durch andere Bitfolgen. Zur Entschlüsselung muss die Rundenfunktion in umgekehrter Richtung durchlaufen werden.

2.2.2 Stromchiffren

Stromchiffren arbeiten pro Grundoperation nicht mit einem Textblock, sondern mit einzelnen Bits oder einem Byte. Der entscheidendere Unterschied zu den Blockchiffren besteht aber vielmehr darin, dass die benutzte Transformationsfunktion nicht konstant ist. Der resultierende Chiffretext für zwei gleiche Klartextphrasen hängt (auch) von der Position des Klartextstücks ab und ist für beide unterschiedlich. Dieses Verhalten resultiert aus dem prinzipiellen Aufbau einer Sromchiffre.

2.3 Asymmetrische Verschlüsselungsverfahren

Dieses Verfahren wird auch als Public-Key-Verfahren bezeichnet, was daher rührt, dass es im Ver- und Entschlüsselungsverfahren zwei unterschiedliche Schlüssel benutzt und einer der beiden öffentlich bekannt gegeben wird. Da die beiden Schlüssel mit einem schwierigen mathematischen Problem miteinander verknüpft sind und die Schlüssellängen viel länger als bei symmetrischen Verfahren sind, sind sie deutlich schwieriger zu brechen.

Mit diesen Voraussetzungen kann ein Empfänger einer geheimen Nachricht dem Sender eben dieser den öffentlichen Schlüssel auf ungesichertem Wege zustellen. Da der private Schlüssel nur im Besitz des Empfängers ist, kann der Sender seine Daten mit dem öffentlichen Schlüssel verschlüsseln und sicher verschicken.

Auch bei diesem Prinzip ist jedoch Vorsicht geboten, da sie keineswegs vor einer Mittelperson schützt. Welche die Daten abfängt und manipuliert um später auszutauschende Daten auch entschlüsseln zu können.

2.4 Hashfunktionen

Hashfunktionen werden auch als Fingerabdruck einer Eingabe bezeichnet, denn sie formen eine Eingabe beliebiger Länge in eine Ausgabe, genannt Hashwert, fester Länge von üblicherweise 128 oder 160 bit um. Die Berechnung des Hashwerts ist nicht umkehrbar, und es wird vielmehr gefordert, dass es nicht zwei verschiedene Eingaben gibt, die den selben Hashwert erhalten. Dies ist natürlich wegen der festen Länge nicht möglich, aber eine solche Doppelbesetzung darf nicht berechenbar sein.

Die Hashfunktion zerteilt einen Text wie bei den Blockchiffren und verfährt dann ähnlich wie bei den Stromchiffren mit einer Art Gedächtnis. Das heisst, dass das Behandeln der Textblöcke von den vorherigen Textblöcken abhängt.

3 Secure Sockets Layer (SSL)

SSL ist ein von Netscape entwickeltes System zur Public-Key-basierenden Absicherung von HTTP-Kommunikationskanälen. SSL bildet eine zusätzliche Schicht zwischen der normalerweise verwendeten TCP/IP-Schicht und HTTP als dem Anwendungsprotokoll liegt. Dieses Einsetzen von SSL lässt sich auch problemlos zum Hinzufügen von zusätzlicher Sicherheit zu normalerweise auf der TCP/IP-Schicht aufsitzenden Anwendungen verwenden. Wie SSL funktioniert und wie SSL mit HTTP zusammen angewandt wird, behandeln die folgenden Kapitel.

3.1 Ziele

Mit dem Ziel vor Augen, eine sichere Kommunikation über ein unsicheres Medium zu ermöglichen, definiert SSL ein Protokoll, das eine Verbindungssicherheit bereitstellt, die drei grundlegende Eigenschaften besitzt:

- Verbindungssicherheit
  Nach einem anfänglichen Handshake wird mit Hilfe eines Verschlüsselungsverfahrens ein geheimer Schlüssel definiert. Zur Datenverschlüsselung wird eine symmetrische Verschlüsselungsmethode verwendet.
- Optionale Authentifizierung
  Die Identität des Kommunikationspartners kann mit Hilfe eines asymmetrischen Verschlüsselungsverfahrens (d.h. mit Hilfe eines öffentlichen Schlüssels) authentifiziert werden.

- Zuverlässigkeit einer Verbindung
  Die Verbindung ist zuverlässig. Die Nachrichtenübertragung schliesst eine mit Hilfe eines verschlüsselten Message Authentication Code (MAC) vorgenommene Integritätsüberprüfung der Nachrichten ein. Die MAC-Berechnung wird unter Verwendung sicherer Hash-Funktionen vorgenommen.

Im folgenden werden die Ziele des SSL-Protokolls nach ihrer Wichtigkeit geordnet aufgeführt.

- -Kryptografische Sicherheit
  SSL sollte dazu verwendet werden, eine sichere Verbindung zwischen zwei Stellen aufzubauen.
- Interoperabilität
  Unabhängige Programmierer sollten SSL einsetzende Anwendungen entwickeln können, die in der Lage sind, Verschlüsselungsparameter auszutauschen, ohne jeweils den Code der anderen zu kennen.
- Erweiterbarkeit
  SSL versucht, einen Rahmen bereitzustellen, innerhalb dessen sich neue Verfahren sowohl zur Erstellung von öffentlichen Schlüsseln als auch zur Verschlüsselung grosser Datenmengen dem Erfordernissen entsprechend miteinander verbinden lassen. Auf diese Weise werden auch zwei Sekundärziele erreicht: es entfällt die Notwendigkeit der Erstellung eines neuen Protokolls (und damit die Möglichkeit des Auftretens neuer Schwächen), und es wird umgangen, eine vollkommen neue Sicherheitsbibliothek implementieren zu müssen.
- Relative Wirksamkeit
  Kryptografische Verfahren, insbesondere Operationen mit öffentlichen Schüsseln, sind häufig sehr rechenintensiv. Aus diesem Grund enthält SSL ein Schema zum Caching von Sitzungen, um so die Anzahl der von Grund auf neu aufzubauenden Verbindungen zu reduzieren. Ausserdem wurde darauf geachtet, die Netzwerkbelastung gering zu halten.

Diese Vorgehensweise bietet sehr wohl Schutz vor Belauschung, jedoch nicht von Angriffen einer Mittelperson.

3.2 Funktionsweise

Im allgemeinen lassen sich mit Hilfe von SSL drei unterschiedliche Arten von Verbindungen zwischen Client und Server aufbauen, die sich bezüglich des jeweils eingesetzten Authentifizierungsverfahrens unterscheiden. Zum Zwecke der Authentifizierung wird ein von einer akzeptablen Authentifizierungsstelle ausgegebenes Zertifikat benötigt.

- Anonymität
  Bei diesem Szenario werden weder der Client noch der Server authentifiziert
- Server-Authentifizierung
  Bei der Server-Authentifizierung muss der Server ein vom Client akzeptiertes Zertifikat vorweisen. Obwohl dem Server die Identität des Clients nicht bekannt ist, kann sich der Client über die Identität des Servers sicher sein.
- Authentifizierung beider Parteien
  Bei diesem Szenario werden sowohl der Client als auch der Server durch Zertifikate authentifiziert, d.h. es kennt jeder die Identität des anderen.
-

SSL besteht aus zwei Phasen. Während der ersten Phase findet ein Handshake statt, bei dem die jeweiligen Fähigkeiten beider Seiten festgestellt werden und eine optionale Authentifizierung vorgenommen sowie das bei dieser Sitzung verwendete Verschlüsselungsverfahren ausgewählt wird. SSL basiert auf dem Sitzungskonzept. Unter Verwendung leistungsfähiger Verschlüsselungsverfahren wird ein Sitzungsschlüssel ausgetauscht, der zum Verschlüsseln der zwischen Client und Server ausgetauschten Daten dient. Dieser Sitzungsschlüssel verwendet ein schwächeres (aber effizienteres) Verschlüsselungsverfahren als das zum Austauschen der Schlüssel eingesetzte. Dies ist vertretbar, da der Schlüssel lediglich für die Dauer einer Sitzung eingesetzt wird. Falls eine der beiden Seiten davon ausgeht, dass der Sitzungsschlüssel nicht mehr sicher ist, wird ein neuer Handshake inklusive der Erzeugung eines neuen Sitzungsschlüssels eingeleitet. SSL definiert eine Reihe unterschiedlicher Algorithmen sowohl für den Schlüsselaustausch als auch für den Sitzungsschlüssel unterstützten Algorithmen.

3.3 Kombinieren von HTTP und SSL

Da ein grosser Unterschied zwischen einer normalen (unsicheren) TCP-Verbindung und einer SSL-Verbindung besteht, muss der Client zuerst wissen, welche Art der Verbindung er aufbauen soll. Dies geschieht indem der URL nicht den Präphix "http" sondern "https" verwendet. Eigentlich gibt es die Möglichkeit SSL als Transportprotokollschicht im Betriebssystem zu implementieren. Da dies aber noch nicht sehr verbreitet ist, muss jedes Anwendungsprogramm SSL selber implementieren.

4 SHTTP

Die Alternative zu HTTPS stellt SHTTP dar. Obwohl die Verschlüsselungsfähigkeit beider Protokolle ähnlich ist, ist SHTTP weit weniger verbreitet.

SHTTP definiert ein auf HTTP basierendes Nachrichtenformat. Dieses erweitert HTTP durch Sicherheitsmerkmale, die eine Authentifizierung, sichere Datenübertragung sowie Verhandlungsoptionen zwischen Client und Server ermöglichen. Der grundleende Gedanke von SHTTP liegt in der sicheren Kapselung von HTTP-Nachrichten. SHTTP ist nicht von einer bestimmten HTTP-Version abhängig, da die Kapselung von HTTP-Nachrichten nicht auf der exakten HTTP-Syntax beruht.

5 SSH – Secure Shell

SSH bezeichnet ein Protokoll und eine Software-Suite zur kryptographischen Absicherung unterschiedlicher Kommunikationskanäle über potentiell unsichere Netzwerke. Die Verschlüsselung basiert auf dem asymmetrischen Public-Key-Prinzip in Kombination mit symmetrischer Verschlüsselung. Dabei kann nur ein passender privater Schlüssel Daten entschlüsseln, die mit dem öffentlichen Schlüssel kodiert wurden und umgekehrt. In der Praxis schickt ein Server "Challenges" (Herausforderungen) an einen Client. Ist dieser in der Lage, eine Challenge korrekt zu entschlüsseln oder kann der Server eine vom Client signierte Challenge verifizieren, so gilt der Test als bestanden.

Beim Aufbau der Verbindung erhält zunächst der Client den öffentlichen Schlüssel des Servers. Er generiert einen für jede Verbindung neu zu schaffenden symmetrischen Schlüssel, verschlüsselt ihn mit dem öffentlichen Key des Servers und schickt ihn diesem. Von nun an verläuft die Übertragung sämtlicher Daten zwischen beiden Rechnern verschlüsselt.

Bei der Ermittlung der Zugangserlaubnis lassen sich hauptsächlich vier Verfahren unterscheiden, wovon die ersten beiden als rechner-, die letzten beiden als anwenderbasiert gelten:

- Reine ~/.rhosts- und hosts.equiv-basierte Authentifizierung:
  Gilt wegen der leicht vortäuschbaren falschen Rechneridentität als hochgradig unsicher und wird von SSH2-Servern nicht mehr unterstützt
- Rechnerbasierte Public-Key-Authentifizierung:
  Ein Benutzer erhält wie beim ersten Verfahren Zugang zum System, jedoch prüft die Software zusätzlich die Identität des Client-Rechners über das Public-Key-Prinzip.
- Benutzerbasierte Public-Key-Authentifizierung:
  Der Benutzer bestätigt seine Identität mit seinen eigenen Schlüsseln.
- Passwortauthentifizierung:
  Schlagen die ersten Verfahren fehl bzw. hat der Benutzer oder der Client-Rechner gar keine Schlüssel definiert, fällt der Server auf die herkömmliche Passwortauthentifizierung zurück. Da seit Beginn der Verbindung die Verschlüsselung aktiv ist, geht nie ein Passwort unverschlüsselt zum Server.
-

Mit SSH lassen sich herkömmliche r-Tools und Telnet komplett ersetzen. Einsatzmöglichkeiten sind Remote-Logins, Remote-Ausführung textbasierter und grafischer Programme sowie Dateiübertragung mit Verschlüsselung und Kompression.

Weitere Einsatzgebiete eröffnen sich durch Port-Forwarding. Dabei tunnelt SSH während einer Verbindung alle auf einem lokalen Port eintreffenden Pakete und leitet sie verschlüsselt und eventuell komprimiert an einen Port auf dem Zielrechner. Damit lassen sich zahlreiche TCP/IP-basierte Kommunikationsprotokolle wie POP und IMAP ohne eine Erweiterung der ursprünglichen Programme verschlüsseln.

6 Schlusswort

Das Verlangen und die Technik von sicheren Kommunikationswegen ist sicher nicht mehr in der ersten Phase, aber doch immer noch am Anfang. Die Grundlagen und zum Teil auch schon ganze Strukturen für sichere Übertragungsarten sind schon jetzt Vorhanden, werden aber hauptsächlich wegen Mangel an Vertrauen noch nicht sehr oft genutzt. Deshalb ist es auch schwierig die Standardisierung voranzutreiben. Dazu kommt noch, dass die Angreifer auch dazulernen und Verschlüsslungsverfahren müssen ständig verbessert werden.

Die Voraussetzungen, die an einen sicheren Kommunikationsweg gestellt werden sind Vertraulichkeit, Authenzität, Integrität und Nichtabstreitbarkeit. Um dies zu erreichen können verschiedene Methoden und Verfahren angewandt werden. Es ist sicher wichtig, dass man die Verschiedenen Verschlüsselungsangebote studiert und dann das für die Anwendung geeignetste wählt. Einige Beispiele wurden ein wenig unter die Lupe genommen. Das waren: SSH, SSL, SHTTP.

Referenzen

1. S. Leich: Doppelt genäht; iX, Heft 1, Seiten 146-149
2. E.Wilde: World Wide Web – Technische Grundlagen; Springer Verlag, 1999 Berlin, Deutschland, Seiten 127-135
3. M. Thorbrügge: Lückenfüller; c’t, Heft 16, Seiten 176-179

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Elektronische Post im Internet

Zimmermann Luca
zluca@student.ethz.ch
19. Dezember 2001

1 Wissenswertes über die E-Mail

Was vor geraumer Zeit mühsam zu Fuss oder hoch zu Ross über Stock und Stein, später dann auf den ersten Strassen und schliesslich auch noch durch die Lüfte transportiert wurde, wird heutzutage innerhalb weniger Sekunden von einem Teil der Erde zu seinem Bestimmungsort irgendwo an einem Ort ausgeliefert.

Ich spreche natürlich nicht nur über irgendwelche Geschenke, die gebeamt wurden, Geldeinzieher oder dessen Rechnungen, nein, sondern über die sich stetig weiterentwickelnde Nachrichtenübermittlung. Ich habe nun einige Informationen zum Thema E-Mail Nachrichten gesammelt und in diesem Dokument zusammengestellt.

Als sich langsam die ersten Netzwerke bildeten, wurde der Austausch elektronischer Nachrichten enorm wichtig, da die Netzwerke immer grösser wurden und mit der Erweiterung zum Internet auch die Distanzen zunahmen. Vorerst war die Nutzung dieser Netze ausschliesslich für Wissenschaftler und für das Militär bestimmt. Nachdem auch erst einmal die Behörden in den Genuss von gemeinsam verwalteten Ressourcen kamen, konnte man die Entwicklung trotz aller Skepsis nicht mehr aufhalten.

Der E-Mail Dienst basiert auch auf Übermittlungs-Protokollen, wie die meisten Netzwerkanwendungen. Weil aber die Entwicklung nicht stornierte, die Verbreitung und die Anforderungen stark anstiegen, durften diese Dinge nicht in Vergessenheit geraten. So kamen unzählige Erweiterungen und Neuversionen dazu, dass ein Ordnungsdienst dringend nötig wurde. Damit sich die heutige Form der Nachrichtenkommunikation überhaupt soweit entwickeln konnte, hat sich eine internationale Organisation die Aufgabe gestellt, diese Herausforderung zu meistern. Das IMC (Internet Mail Consortium) strukturiert die bereits vorhandenen Protokolle und prüft neue Vorschläge auf ihre Tauglichkeit in Bezug auf weltweite Nutzung. Wenn ein neues Protokoll aufgenommen werden soll, so wird es noch nicht von Anfang an als Standard definiert, sondern erst einmal als I'D (Internet-Draft) eingeführt. So können diese von allen genutzt und einer Prüfung auf Kompatibilität unterzogen werden. Je nachdem wie sich die I'D verhält, wird sie schnell als RFC (Request For Comments) eingestuft oder noch solange als I'D behandelt, bis sie verbessert oder endgültig verworfen wird. Eine Liste aller RFCs für E-Mail-Dienste findet man unter www.imc.org/rfcs.html. Die Standards sind deshalb so schwierig festzulegen, weil sich die ganze Umgebung stetig wandelt und jeder Nutzer eine bessere Implementierung verlangt.

1.1 Allgemeines

Die elektronische Nachricht verdankt ihre vermehrte Nutzung mehreren Umständen. Alles in Allem ist eine E-Mail eine reine Textdatei, die von jedem Editor gelesen werden kann, falls die Nachricht entschlüsselt wurde. Früher erledigten separate Programme die Codierung aller ASCII-Zeichen der Nachricht in Zahlen, die dann gesendet wurden. Mit der entsprechenden Software liess sich das Ganze nach erfolgreicher Übermittlung wieder decodieren. Aber die ASCII-Zeichen konnte trotzdem jeder auf seinem eigenen System lesen.

1.1.1 Aufbau

Der Aufbau eines elektronischen Briefes kann folgendermassen beschrieben werden. Zuerst wird der Header geschickt, dann eine Leerzeile, gefolgt vom Body. Diese Anordnung veranlasst den aufgerufenen Mail-Server, die E-Mail an den gewünschten Ort zu verschicken.

- Die wichtigsten Header beinhalten die Informationen wie Absender, Empfänger und Absenderdatum. Sie waren am Anfang ausreichend, wurden dann schliesslich aber doch noch ergänzt.

Beschreibt:	**Benötigter Header:**
Absendername	`From:`
Empfängeradresse	`To:`
Absenddatum	`Date:`
Ein Durchschlag dieser Nachricht wird an die Adresse geschickt, die unter dieser Rubrik eingetragen wird. Steht für „Carbon Copy“	`Cc:`
Der eigentliche Empfänger unter ´`To:`´ kriegt nicht mit, dass eine Kopie der Mail zusätzlich an Dritt-Personen weitergeleitet wurde. Steht für “Blind Carbon Copy”.	`Bcc:`
E-Mail-Adresse des Absenders	`Sender:`
Den Titel oder den Betreff der Nachricht	`Subject:`
E-Mail-Adressen derjenigen, die zum Beispiel eine Lesebestätigung angefordert haben.	`Reply to:`
Einzigartige Kennnummer der Nachricht, mit der sie jederzeit wiedererkennt werden kann.	`Message-ID`
Den Weg, den die Mail gemacht hat und dabei Daten gespeichert wurden. Darunter fallen verwendete Protokolle und angelaufene Adressen, wie auch Datum und ID.	`Received`
Die zur Verschlüsselung verwendete Software an	`Encrypted`

Tab. 1 Einige Header mit dessen Funktionen

- Der Body enthält die eigentlichen Informationen, die es mitzuteilen gilt. Die E-Mail ist ein Textdokument, welches also nur ASCII-Zeichen verwendet. Nach der Eingabe der Header folgt eine Leerzeile, gefolgt vom eigentlichen Body. Die Nachricht gilt erst als abgeschlossen, wenn auf einer einzelnen Zeile ein ´.´ geschrieben wird.

1.1.2 Accounts

Die E-Mail Adresse Sie bestehen aus einem Benutzernamen, gefolgt von einem ’@’ und schliesslich dem Domain Name, was dem Rechnernamen entspricht.

In der Anfangszeit war der Domain Name tatsächlich auch der Rechnername des Empfängers, was sich schnell änderte, da das Internet mit seinen Möglichkeiten einem immer breiteren Publikum zugänglich gemacht wurde. Es begann die grosse Geschäftemacherei mit den E-Mail-Accounts, worauf nach der grossen Euphorie schliesslich auch noch kostenlose Anbieter wie Pilze aus dem Boden schossen. Diese FreeMail Accounts sind jedoch sehr begrenzt in ihrer Mail-Box-Grösse, was etlicher Pflege bedarf, wenn grössere Nachrichten umherverschickt werden. Nützlich hingegen ist eine solche gratis Internet-Adresse auf jeden Fall, sei es nur um als Zweit-Adresse angeben zu können.

1.1.3 Attachments

Zu Beginn des Internet-Zeitalters dachte man noch nicht einmal an das Verschicken von irgendwelchen Anhängseln. Wenig später machte sich jedoch ein Bedürfnis nach diesem Service bemerkbar. Sei es für den Versand von Bildern, Bibliotheken oder sogar ganzen Programmen. Seit 1996 hat sich nun auch für verschiedene Dateiformate, die zusammen mit einem Mail verschickt werden, ein Standard durchgesetzt. Die Anhängsel (Attachments) einer Nachricht werden ähnlich einer Mail strukturiert. Damit lässt sich die Nachricht in Teile gliedern, denen immer zuerst eine Art Header vorangeht, der angibt, um welche Art Informationen es sich handelt und wie sie codiert ist. Damit lässt sich auch eine Verschachtelung vollführen, was für Filterzwecke enorm optimierend sein kann. Der Multipurpose Internet Mail Extensions Standard, kurz MIME, ist eine der unzähligen Erweiterungen des Sendeprotokolls, welches diese Aufgabe übernimmt. Das MIME-Protokoll sorgt dafür, dass der Mail noch zusätzliche Header angehängt werden, unter Anderen sind diese zwei besonders wichtig; MIME-Version `<version number>` und Content-Type `<type>/<subtype>`. Immer wenn im Header ´`multipart/alternative`´ steht, bedeutet dies, dass der folgende Teil dieselben Informationen in verschiedenen Darstellungsformaten beinhaltet. Der Client des Empfängers kann nun entscheiden, welche Art der Informationsanzeige haben möchte. Heutzutage wird aber bei den meisten Clients eine 8-Bit-durchlässige Mail-Verbindungen eingesetzt. Die Codierung, die für Texte angewandt wurde, nennt sich ’quoted-printeble’, wobei jedes Nicht-ASCII-Zeichen durch ein Gleichheitszeichen und eine Hexadezimalzahl dargestellt wird. Für Bilder, Audiofiles, Videos, oder Publikationen, die angehängt werden, kommen verschiedene Kodes zum Einsatz, was leider häufig zu Kommunikations-problemen kommt, da Server und Client nicht dieselbe Implementierung benutzen.

2 Versand elektronischer Nachrichten

Da der heutige PC-Benutzer meistens eine Maschine am Arbeitsplatz vorfindet und auch zu Hause einen Rechner für private Anwendungen und Arbeiten stehen hat, ist die E-Mail auch von ihm geschätzt und wird rege benutzt. Es erleichtert natürlich schon erheblich die Motivation, wenn man vom Arbeitsplatz aus die noch nicht erledigte Arbeit abschicken lassen kann, und später von zu Hause aus zugreifen kann. Doch wie soll das funktionieren, wenn doch diese beiden Rechner nicht im selben Netzwerk integriert sind? Früher baute der Absender zu diesem Zweck eine Punkt zu Punkt Verbindung zum Empfänger auf und verschickte die Nachricht. Das funktionierte dann später auch mit mehreren Benutzern eines hauseigenen Netzes. Das war damals kein Problem, da ja die Daten auf einem zentralen Rechner lagen, auf den alle Zugriffs-Rechte besaßen, die sich identifizieren (Name, Passwort) konnten. Über größere Distanzen hingegen wird heutzutage über die Telefonleitung eine Verbindung zum Provider hergestellt, so dass abgeschickte Nachrichten zuerst einmal auf ihm gespeichert werden. Diese Rechner, die auch Mail-Server genannt werden, können aber auch Rechner eines anderer Anbieters von E-Mail-Diensten sein. Sobald er Rechenzeit freigibt, wird die E-Mail an den Empfänger geschickt, der nur noch sein Postfach zu leeren braucht. Voraussetzungen dafür sind nur, dass sich der Empfänger als diesen identifizieren kann, und dass beide Systeme dieselben Protokolle benutzen. Vielleicht sei zur Erinnerung nochmals gesagt, dass, wie bei allen gängigen Server-Client-Verfahren so üblich, von beiden Seiten die identischen Protokolle verwendet werden müssen.

2.1 Verbindung

Weil aber die normalen Benutzer ihre Rechner nicht andauernd online schalten wollen, wird die Verbindung vom Mail-Client bei Bedarf aufgerufen. Dazu stehen dem Benutzer heute drei Auswahlmöglichkeiten zur Verfügung, wie sie ihren Mail-Client konfigurieren:

- Getrennt: Sobald der Benutzer eine Verbindung zum Mail-Server herstellt, werden die eingegangenen Nachrichten zum Mail-Client nach Hause geschickt. Die Daten der elektronischen Nachrichten liegen nun auf beiden Rechnern vor.
- Offline: Der Mail-Client öffnet in gewissen Zeitabständen eine Verbindung zum Mail-Server und fragt dort nach neu eingegangenen Nachrichten. Fall eine oder mehrere Nachrichten ungelesen sind, werden diese automatisch an den Mail-Client geschickt und anschliessend auf dem Server gelöscht.
- Online Bei dieser Variante nimmt der Mail-Client mit dem Mail-Server Kontakt auf und führt alle Kommandos auf dem Server durch. Das heisst, der Rechner zu Hause hatte diese Daten nie gespeichert, sie liegen nur beim Server vor.

Ich möchte nicht näher eingehen auf alle verschiedenen Variationen der Protokolle, die für das Versenden der elektronischen Post verantwortlich sind. Ich habe nur die zwei bzw. drei meist verbreiteten Protokolle hier aufgelistet und näher miteinander verglichen. Hier sei auch wieder angemerkt, dass unzählige Varianten zu finden sind.

Wichtig jedoch ist zu wissen, dass für den Transport unserer Nachrichten im Internet zwei grundverschiedene Protokolle eingesetzt werden. Das Verschicken eines elektronischen Briefes ist eine Abfolge von Protokollen.

- Einerseits wurde ein Dienst entwickelt, um den Host Computer mit dem Server zu verbinden und Post zu verschicken. Das hierfür meistverbreitete Protokoll ist sicherlich das SMTP (Simple Mail Transfer Protocol) und das ESTMP (Extended SMTP).
- Demzufolge musste man noch ein Komplementärprotokoll implementieren, welches für den Abholdienst zuständig ist. Die zwei meistgenutzten Protokolle für diesem Zweck sind wohl das POP (Post Office Protocol) bzw. heute POP3 und das IMAP (Internet Message Access Protocol).

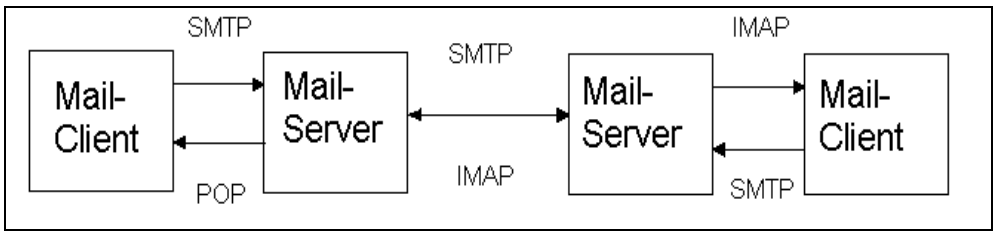


Fig.1 möglicher Verlauf und die dafür verwendeten Protokolle

2.2 Senden

Das Senden von E-Mails wird, wie bereits gesagt wurde, meistens mit dem standardisierten SMTP ausgeführt. Wobei nebst diesem Protokoll auch noch die in Europa und Kanada häufig genutzte X.400-Empfehlung und später das erweiterte SMTP aufkamen.

Als man jedoch mit der Zeit merkte, dass dieser Standard den Anforderungen an die elektronische Post nicht mehr gerecht wurde, definierte man den RCF 1869, welcher die Möglichkeit bereitstellte, Erweiterungen für den ursprünglichen RFC 821 [211] (das ursprüngliche SMTP-Protokoll, welches eine Mail als reines ASCII-Textdokument definiert) zu protokollieren.

Heute wird aber schon fast überall mit Programmen gearbeitet, die eine Arbeitsteilung vornehmen. Das heisst, der Mail-Client (meistens ein Programm mit graphischer Benutzeroberfläche, Bsp.: Outlook, ...) kümmert sich um den Versand der Mails zum Mail Transport Agent (MTA) des Providers. Dieser hat dann die Aufgabe, die zugehörigen Daten für den weiteren Versand zu suchen und schliesslich die Nachricht beim Empfänger abzuliefern. Unter Umständen kann es vorkommen, dass eine Nachricht bei mehreren MTAs vorbeikommt und schliesslich dort auch liegen bleibt, bis der Empfänger seinen Mail-Client anwirft und seine Post abholt.

Wer dennoch einmal keinen passenden Browser zur Verfügung hat und eine Mail verschicken muss, kann sich einfach via Telnet eine TCP/IP-Verbindung zum Port 25 des Zielrechners aufbauen und so kommunizieren.

2.2.1 SMTP

Das erste SMTP-Protokoll besass ein sehr einfaches Ablaufschema. Der Mail-Client öffnete eine Punkt zu Punkt-Verbindung zum Port 25 des Zielrechners der Nachricht. Dazu benötigte der Absender aber erst einmal die IP-Adresse des Empfängers. Mittels der DNS-Server, deren Funktionalität wir bereits kennen, ermittelte man früher diese benötigten Informationen. Heute öffnet das SMTP-Protokoll eine Verbindung und meldet sich beim passiv wartenden MTA an. Nach einem gewissen Anmeldeschema, dass bei allen TCP/IP-Verbindungen ähnlich ist, ist der Client nun in der Lage, die Nachricht zu verschicken. Der MTA bestätigt nach erfolgreichem Datentransfer, dass die Daten nun auf dem Server gespeichert sind, und später dann weitertransportiert werden.

Client	**Server**
<Hat Post, die er verschicken will, weshalb er eine Verbindung zum Port 25 will>	`220`
HELO <irgendeine.domain>	`250 ok`
MAIL FROM:<sender@gratis.ch>	`250 ok`
RCTP TO:<empfänger@billig.com>	`250 ok` oder `user unknown`
DATA	
Header der Mail	
Body	
.	`250 ok`
QUIT	`221`

Tab. 2 Möglicher Ablauf einer Anmeldung beim Server und Versand der Daten nach RCF 0821.

2.2.2 ESMTP

Da das SMTP auch relativ einfach zu implementieren ist, ist es verständlich, dass damit nicht alle Optionen ausgenutzt werden können. Deshalb hat man für die Erweiterung des SMTP-Protokoll einen Platz freigelassen bei den übrigen Standards. Das ESMTP unterstützt folgende Eigenschaften:

- **Ankündigung der Nachrichtengrösse (RFC 1078):**

  Dies kann besonders bei den erwähnten grossen Mails nützlich sein, da man entscheiden kann, ob man eine Mail mit gewisser Grösse annehmen will.

- **Anzeigemöglichkeit für 8 Bit-Zeichensätze (RFC 2047):**

Da das ursprüngliche Protokoll ausschliesslich für Textdateien, welche mit 7 Bit codiert werden, gedacht war, erweiterte man es noch um die Möglichkeit, auch nicht ASCII-Zeichen, wie Umlaute, Buchstaben mit Akzenten usw., sicher zu übertragen und anzuzeigen.

- **Zerlegung einer Mail in Teilstücke (RFC1830):**

  Weil bei grösseren und binären Nachrichten auch längere Versandzeiten oder Verbindungsabbrüche in Kauf zu nehmen sind, wurde eine Funktion hinzugefügt, die es dem Client ermöglicht, eine Nachricht in kleine Teilstücke zu unterteilen und diese nacheinander abzuschicken. Bei erfolgreicher Übertragung des ersten Paketes wird eine Bestätigung zurückgeschickt, worauf erst der nächste Teil versendet wird (RFC 1845).

2.3 Empfangen

Das Verschicken einer Nachricht in die Weiten der virtuellen Welt hat schon einmal funktioniert. Doch wie kommt man nun an die gesendeten Daten wieder heran? Natürlich wurde daran gedacht und ein ebenbürtiges Protokoll geschrieben, dessen Funktion dem Abholen der Post aus dem Schliessfach glich. Da die vom Absender geschickten Daten nun auf dem Server-Rechner (meistens beim Provider) gespeichert sind, hat der Empfänger jetzt die Möglichkeiten, mittels eines Mail-Client von zu Hause aus auf sein Postfach zuzugreifen und seine Nachrichten zu bearbeiten. Für die Übertragung der Nachrichten vom MTA des Empfängers zum Client werden wie besagt wieder spezielle Protokolle benötigt.

2.3.1 POP/POP3

Die ersten POP-Protokolle umfassen im Wesentlichen drei Befehle. Man kann abfragen wie viele Mails neu angekommen sind, sich die gewünschte Mail ausliefern lassen und eine beliebige Mail löschen. Ansonsten läuft der Vorgang analog dem Senden. Der Mail-Client öffnet eine TCP/IP-Verbindung zum Port 110 des MTA und muss sich nun aber zuerst zu erkennen geben. Der MTA fragt nach Username und Passwort (beides wird bei einer POP-Verbindung unverschlüsselt übertragen) und gibt bei erfolgreicher Authentifikation die Einsicht frei. Eine POP-Sitzung findet Offline statt, da die einzigen richtigen Befehle sowieso nur Abholen oder Löschen sind.

2.3.2 IMAP

Mit der stetigen Entwicklung musste auch noch ein Protokoll entwickelt werden, welches den Online-Modus unterstützt. Zu diesem Zweck schrieb man das ´Interactive Mail Access Protocol´, welches erst später zu seinem heutigen Namen ´Internet Message Access Protocol´ gefunden hat. Es ist auf jeden Fall eine funktionelle, nicht syntaktische Obermenge von POP. Die wichtigsten Erweiterungen gegenüber dem POP sind:

- Es ist möglich, weitere Ordner nebst dem Posteingang zu erstellen, zu bearbeiten und zu löschen. Es unterstützt sogar Ordnerhierarchien, so dass man auf dem IMAP-Server eine geordnete Verwaltung seiner Mails einrichten kann.
- Ordnerbearbeitung über das Netzwerk wurde auch erst mit dem IMAP ermöglicht. Die Benutzer an verschiedenen Rechnern werden über Aktualisierungen der gemeinsamen Ordner informiert, Ordner können verschoben und auch mit Nachrichten-Flags (gelesen, ungelesen oder markiert usw.) versetzt werden.
- IMAP ermöglicht die Erkennung einzelner MIME- und anderer Teilstücke der Mail, die dann separat heruntergeladen werden können. Die Struktur der Mail wird vorerst auf dem Client gespeichert, so dass anhand dieser entschieden wird, welche Stücke man sehen will.

Leider wird anhand der Komplexität von IMAP viel häufiger das leichter zu implementierende POP3 verwendet, obwohl die Nachteile des IMAP nicht bei allen Rechnern gleich sind; Die Funktionen des IMAP-Protokolls werden noch nicht in allen Anwendung erfordert, so dass die

Ressourcen stark in Anspruch genommen werden. Mit den immer neueren Versionen steigt aber auch die Anzahl Nutzer dieses Dienstes.

Empfänger-Client	Server
<Will eine Verbindung zu Port 110 öffnen>	`+ ok`
USER <student>	`+ ok, please send password`
PASS <offenesgeheimnis>	`+ ok, 2 messages ready for student`
RETR 1	`+ ok  message 1 (644 octets)`
	Sendet Mailtext mitsamt Infos über Verlauf
	.
DELE 1	`+ ok`
QUIT	

Tab. 3 Möglicher Ablauf eines Verbindungsaufbaus nach RCF 1939 (POP3), worauf eine von zwei neuen Nachrichten abgerufen wird. Anschliessend wird sie auf dem Server gelöscht.

3 Schlusswort

Die elektronische Nachricht besteht im Wesentlichen aus einem Header und einem Body, was die eigentlich Nachricht ist. Ursprünglich war eine E-Mail nur zum Versand von Texten gedacht, was aber den heutigen Anforderungen widerspricht. So ist es heutzutage auch möglich Audio-, Programm- und sogar Video-Daten als Attachments vollständig zu übertragen.

Wir wir gesehen haben, sind für den Transport einer Mail im wesentlichen die zwei Protokolle SMTP und POP zuständig und die dazu passende Software. Die Protokolle sind lediglich für den Versand zuständig, legen aber keinen Weg fest. Sie bestimmen nur die Art der zu übermittelnden Daten und wohin sie schliesslich gelangen sollten, aber den zu durchlaufenden Weg lassen sie offen. Für die Attachments ist das erweiterte MIME-Protokoll zuständig.

Die Mails lassen sich mit den heutigen Standards auch in einem Netzwerk gut verwalten, was es für viele Firmen attraktiv erscheinen lässt, ihr eigenes LAN (Local Area Network) in Betrieb zu nehmen, was wiederum für eine Weiterverbreitung der Benutzer sorgt.

Es ist jedoch auch noch zu erwähnen, dass bis jetzt eigentlich nur über die positiven Seiten der elektronischen Post geschrieben wurde, worauf hier noch einige Gedankenanstösse folgen:

- Die herkömmlichen Übertragungsmedien, wie Briefe, Fax, Telegramm oder auch Telefone werden noch und noch verdrängt aus unserem täglichen Leben.
- Die Sprache, die in einer E-Mail oder einem SMS mittlerweile benutzt wird, gleicht einem Wirrwarr aus Symbolik und Neu-Englisch.
- Die Herkunft und der Inhalt der eingegangen Mail könnte während des gesamten Versands von Dritt-Personen manipuliert worden sein, da die herkömmlichen Standards keine Verschlüsselung und Kennzeichnung vorsehen. Es sind jedoch Protokolle unterwegs, die diese Probleme behandeln. Wie das S/MIME-Protokoll (Secure MIME), welches auch immer weitere Verwendung findet.
- Aufbau einer virtuellen Freundschaft ist ja schön und gut, aber im Extremfall wird man sich irgendwann vom realen Freundeskreis isolieren und man kennt nur noch irgendwelche Personen, die man noch nie gesehen hat.

- Möglichkeiten zur Lahmlegung eines gesamten Netzwerkes sind für Jedermann gegeben. Da mittels geeigneter Adressen die rasche und ortsunabhängiger Verteilung gewährleistet ist.

Zu guter Letzt möchte ich noch anmerken, dass der E-Mail-Dienst mit seinen Möglichkeiten eine weitere Entwicklung der Menschheit ist. Die Menschen haben sich die Grundlagen dazu selber konstruiert und werden auch immer weiter daran forschen. Wie in allen Bereichen muss man einfach an den gesunden Menschenverstand glauben, damit diese Errungenschaften der Technik nicht missbraucht werden. Wir werden uns immer weitere Dinge einfallen lassen, um schneller und ökonomischer zu kommunizieren, und weiterforschen. Doch wann das Ganze ein Ende haben wird, ist fraglich. Da die heutige Forschung schliesslich auch immer mehr an ihre Grenzen gelangt, ist vorweggenommen, dass die Nachrichtenübermittlungstechnik auch begrenzt ist, und auch bleiben wird.

Bibliographie:

- c't 1999 Heft 8, S. 152-154.
- Jürgen Plate, Internet-Möglichkeiten und Dienste, 2.1 Die elektronische Post, http://www.fs.ei.tum.de/admin/howto/internet/inetein2.html#2.1.
- Internet Mail Consortium Homepage: http://www.imc.org.

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

WAP und WML

Tobias Rein
reint@ee.ethz.ch
1. Februar 2002

1 Wireless Application Protocol (WAP)

Neben dem Internet-Zugriff via PC kommen in letzter Zeit immer häufiger mobile Alternativen ins Gespräch. Als Schlüsselfiguren fungieren dabei die so genannten WAP-Devices. WAP ist im allgemeinen Sprachgebrauch das mobile Internet. Wireless Application Protocol signalisiert auch, dass es sich dabei um ein einziges Protokoll handelt für die mobile Datenübertragung. In Wirklichkeit ist WAP aber eine ganze Protokollfamilie, mittels derer die Übertragung von Daten zu mobilen Endgeräten geregelt wird und die auch weitere Spezifikationen wie die Seitenbeschreibungssprache WML enthält.

Obwohl ein Handy heutzutage schon fast zur Grundausrüstung gehört, zeigen sich schon beim Abrufen der persönlichen E-Mail die Grenzen dieser Systeme:

- kleines Display mit niedriger Auflösung und meistens nur einem Farbton
- schmale Bandbreite
- stark begrenzter Speicher

1.1 Das WAP Forum

Gegründet 1997 von den Firmen Phone.com, Ericsson, Motorola und Nokia soll das WAP Forum einen internationalen Standart für den Datenverkehr in Mobilfunknetzen erschaffen. Heute sind es wohl über 400 [4] der weltgrössten Telekommunikationsunternehmen, die dem WAP Forum angehören. Eine möglichst heterogene Zusammensetzung soll dazu beitragen, dass weder Soft- noch Hardwarefirmen die Oberhand dieser Entwicklung erhält.

Das WAP Forum ist eine Non-Profit-Organisation, die Protokolle entwickelt, welche für jedermann offen zugänglich und unabhängig von den Netzwerkstandards sind. Die Philosophie, die bei der Entwicklung von WAP im Vordergrund stand, ist einerseits so wenig Ressourcen des mobilen Gerätes zu verwenden wie nur möglich, und andererseits die Einschränkungen der Geräte durch Bereicherung der Funktionalität des Netzwerkes zu kompensieren. Mit anderen Worten: Den auf dem mobilen Endgerät eingebauten Browser so einfach wie möglich belassen, die erforderliche Intelligenz dafür aufs Netzwerk verlagern.

Im April 1998 wurde eine erste Version der WAP Spezifikationen 1.0 vorgelegt. Als Grundlage diese Spezifikation dienten – soweit möglich – bestehende Standards der Internettechnologien. Doch erst mit WAP 1.1 und WAP 1.2 von Ende 1999 wurde der Standard so stabil und leistungsfähig, dass fast überall auf der Welt WAP-Dienste entstanden.

1.2 Die WAP Architektur

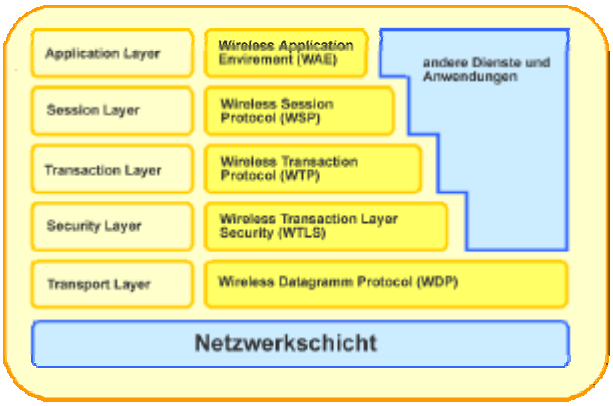


Fig. 1 Der WAP Stack

Wie auch andere Protokollfamilien, basiert die WAP Architektur auf einem Schichtmodel. Der so genannte WAP Stack besteht aus 5 Schichten und überdeckt den gesamten Prozess der schnurlosen Übertragung. Die Abbildung 1 zeigt die Anordnung der verschiedenen Schichten und deren Protokolle.

- Anwendungsschicht (Application Layer)
  Den Application Layer stellt das Wireless Application Environment (WAE) dar. Dessen Aufgabe ist die Bereitstellung von Mitteln zur Entwicklung von Anwendungen und Diensten, die über alle Datenstandards der mobilen Kommunikation hinweg portierbar sind. Primärer Nutzniesser dieser Schicht ist der Micro-Browser, die im Web-basierten Material navigiert.
- Sitzungsschicht (Session Layer)
  Mit Hilfe des Wireless Session Protocol (WSP) regelt die Sitzungsschicht den Ablauf einer Sitzung. Dies sind im wesentlichen folgende Phasen:
    - eine Sitzung starten
    - Inhalte austauschen
    - Sitzung beenden
- Transaktionsschicht (Transaction Layer)
  Die Spezifikationen für die Transaktionsschicht enthält das Wireless Transaction Protocol (WTP). Diese Schicht ist für die Art und Weise des Transportes der Daten verantwortlich und bietet drei Klassen von Übertragungsdienste an:
    - Unzuverlässige Einweg-Anforderung
      Das Mobilgerät sendet eine Anfrage, es wird weder garantiert, dass diese beim Server ankommen, noch dass sie beantwortet wird.
    - Zuverlässige Einweg-Anforderung
      Das Protokoll garantiert den Empfang der Nachricht beim Server.
    - Zuverlässige Zweiweg-Anforderung
      Hier gilt die Transaktion erst dann als erfolgreich abgeschlossen, wenn am Mobilgerät die Antwort des Servers eingetroffen ist.
- Sicherheitsschicht (Security Layer)
  Wireless Transport Layer Security (WTLS) ist eine optionale Schicht, die Verschlüsselungs-einrichtungen beinhaltet. Ebenfalls enthält es Spezifikationen über Datenintegrität, Abhörsicherheit und Benutzer Authentifizierung.
- Transportschicht (Transport Layer)
  Das Wireless Datagram Protocol (WDP) repräsentiert die Transportschicht und bildet die Schnittstelle zum physikalischen Netzwerk. Sie kann an die Vorgaben des Netzwerkanbieters angepasst werden und macht WAP somit völlig unabhängig von der Art der Netzwerkübertragung.

1.3 WAP Technik

Um mit einem mobilen Endgerät eine WML Seite abzurufen, muss dieses mit einem Modem und einem Micro-Browser ausgestattet sein. Die WML Seiten liegen im Internet auf gewöhnlichen Web-Servern. Bevor eine bestimmte WML Seite angezeigt wird, muss diese zunächst angefordert werden. Der so genannte Request wird nicht direkt, wie es bei herkömmlichen HTML Seiten der Fall ist, zum Web-Server geleitet, sondern wird über ein WAP Gateway umgeleitet. Dieses hat die Aufgabe, die binär codierte Anfrage des mobilen Endgerätes in eine HTTP-Anfrage (http request) umzuwandeln und an den entsprechenden Web-Server weiter zu leiten. Daher ist beim WAP nicht nur die IP der entsprechenden Seite erforderlich, sondern auch die Adresse des Gateways, das zwischen dem Mobilfunknetz und dem Internet vermittelt.

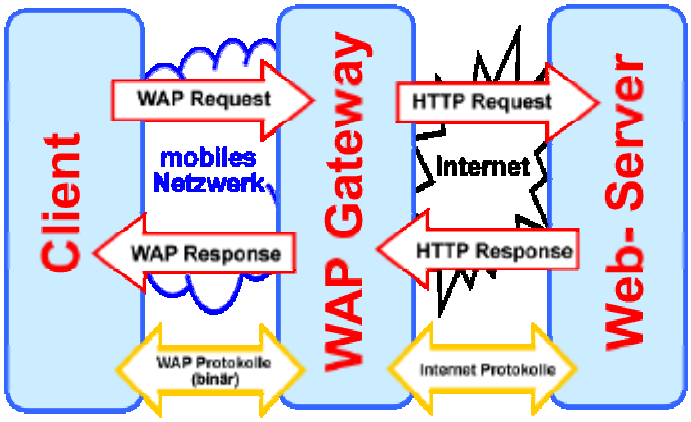


Fig. 2 Der WAP Request

Der Web-Server liefert wie gewöhnlich die Inhalte der betreffenden Seite ans Gateway zurück. Dieses wandelt die Daten in eine Binärverschlüsselung um und sendet es dem mobilen Endgerät. Da es sich bei einer WML Datei um eine reine Textdatei handelt, stellt die Umwandlung eine starke Kompression dar. Die zu übertragene Datenmenge schrumpft dabei bis auf ein Viertel [4] seiner Ursprünglichen Grösse.

2 Wireless Markup Language (WML)

Wireless Markup Language ist die für WAP entwickelte Sprache. WML kann als Zwischenstufe von HTML und XML verstanden werden, die aber auf kleine Displays und geringe Speicher ausgelegt ist. Momentan sind noch alle WML Seiten schwarz-weiss, da Farben zu viele Speicherressourcen benötigen. Das einzige bisherige Grafikformat für WAP ist das WBMP (Wireless Bitmap), das vom bekanten BMP Format herkommt.

2.1 Aufbau

Basiselement innerhalb der WML Syntax ist die so genannte Card. Mehrere dieser Cards lassen sich wiederum in einem Deck (Stapel) zusammenfassen, wobei das Deck als eine logische Einheit, als WML Seite, angesehen werden kann.

Fig. 3 Cards und Decks in WML

Das Deck stellt das oberste Element eines WML Dokument dar. Beim Empfangen einer WML Seite, wird das ganze Deck übertragen und per Default die erste Cars angezeigt. Durch geeignete Navigation kann dann von einer Card zur anderen gewechselt werden.

2.2 Syntax

Die Sprache WML baut auf der Syntax von XML auf. Die Markup-Befehle (Tags) werden wie auch in HTML in eckigen Klammern geschrieben (<…>). Die meisten dieser Tags treten paarweise auf, mit einem Start-Tag der Form <xxx> und einem End-Tag der Form </xxx>. Im Unterschied zu HTML darf in WML (wie auch in XML) der End-Tag niemals weggelassen werden.

WML ist eine case-sensitive Sprache, das heisst, es muss auf die Gross- und Kleinschreibung dringend geachtet werden.

2.2.1 Beispiel

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//DE">
<wml>
  <card>
    <p align="center">
    Hello World
    </p>
  </card>
</wml>
```

Fig. 4 WML-Beispielseite

Der erste Tag *<?xml version="1.0"?>* ist die übliche Deklaration, welche mit den XML Normen zu tun hat. Er gehört zu jedem gültigen WML Dokument. Zusammen mit dem zweiten Tag, indem mit Hilft des Document Type Definition (DTD) die formale Grafik definiert wird, bildet er den so genannten Prolog.

Danach beginnt das eigentliche WML Deck. Es wird mit *<wml>* begonnen und mit *</wml>* geschlossen, die Cards entsprechend mit *<card>* und *</card>*. Meistens wird mit dem card-Tag noch einen Titel und eine card id mitgegeben: z.B. *<card id = „CardOne" title = „example">*.

Zwischen den card-Tags steht die eigentliche Formatierung der Card die beliebig kompliziert sein kann. In unserem Beispiel wird mit *<p align="center">* eine Absatz gemacht und die bekannten Worte „Hello World" erscheinen eingemittelt auf dem Display.

3 Fazit

WAP hatte von Anfang an mit grossen Problemen zu kämpfen. Dadurch wurde es der Technologie nicht einfach gemacht, eine grössere Akzeptanz beim Kunden zu finden. WAP-User klagen ständig über ein extremes Ungleichgewicht zwischen hohen Kosten und niedrigen Geschwindigkeiten, komplizierten Bedienungen sowie instabilen Netzen.

Kleine Zeichen auf einem Display von der Grösse einer Briefmarke und eine für Mobiltelefone mehrfachbelegte Tastatur schränken Komfort, Geschwindigkeit und Spass bei der Nutzung massiv ein [3].

Eines der grössten Probleme von WAP – aus Sicht der Entwickler von WAP-Angeboten – stellt die nicht-standardisierte Benutzerschnittstelle dar. Existieren für das Internet im Wesentlichen zwei verschiedene Browser (Netscape Navigator und Internet Explorer), so sind die Micro-Browser nicht nur unter den verschiedenen Hersteller in der Darstellungsform unterschiedlich, sie unterscheiden sich auch noch in der Modellreihe.

3.1 Aussichten der Zukunft

WAP stellt mit Sicherheit eine sehr bedeutende Technologie für den mobilen Datenzugriff dar. Aufgrund der oben beschriebenen Problemen wird es doch noch einige Zeit und grösseren Entwicklungsaufwandes bedauern, bis es sich durchsetzten wird.

Vor allem auf Seiten der mobilen Infrastruktur wurden in letzter Zeit richtungweisende Entscheidungen getroffen. Mit GPRS (General Packet Radio Service) und UMTS (Universal Mobile Telecommunications System) wird das Problem der geringen Bandbreite aufgehoben. Das derzeitige GSM-Netz (Global System for Mobile Communication) hat eine Übertragungsrate von 9.6 kbps (Kilobits pro Sekunde). Die oben erwähnten Technologien stellen mit 115 kpbs [4] und 2Mbps [3] (Megabits pro Sekunde) ein wesentlich schnelleres Netz zur Verfügung und öffnen somit neue Dimensionen für die mobile Kommunikation. Diese Techniken unterstützen auch ein völlig neues Abrechnungsverfahren, das nach Datenmengen und nicht nach Zeit abrechnet. Somit wäre ein ständiges Online-Sein durchwegs denkbar.

Auch der WAP Standard an sich wird ständig weiterentwickelt und verbessert. Mit der Veröffentlichung von WAP 2.0 im Sommer 2001 ging WAP in eine neue Phase: Eine fast nicht mehr zu überblickende Menge an Standards zu allen Aspekten des "mobilen Internets" zusammen mit einer kompletten Neuformulierung der Sprache WML als XHTML verlangen einiges an Einarbeitung und Umdenken. Es wird noch mehr auf den bekannten und bewährten Internettechnologien aufgebaut.

3.2 Quellenverzeichnis

1. Andreas Hitzig — Drahtlos surfen mit Volldampf
2. Lars Röwekamp — Handy HTML
3. Univ. –Prof. Dr. Gerhard Friedrich — Gegenwart und Zukunft des mobilen Internets
4. http://www.ccwap.de
5. http://www.wap-wissen.de
6. http://www.wapforum.org

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Übertragungs-technologien

Michael Reiterer
michaelr@ee.ethz.ch
2. Jänner 2002

1 Einführung

Der Aufbau des Internets wird häufig mit einem Schichtenmodell beschrieben, in dem jede Schicht (*Layer*) auf den Dienst der darunterliegenden Schicht aufbaut. Der Zugriff auf die physikalischen Übertragungsmedien erfolgt im untersten Layer. In diesen Übertragungsmedien findet die tatsächliche Datenübertragung statt. Somit bauen alle Layer auf den Dienst dieser Medien auf. Deshalb stellt man an sie i.A. hohe Anforderungen:

- hohe Übertragungsraten
- kontinuierliche Betriebsbereitschaft
- geringe Kosten
- gute mechanische Eigenschaften

Die Wahl eines Mediums stellt in jedem Falle eine Kompromisslösung dar, denn das ideale Medium gibt es unglücklicherweise nicht. Das Signal am Ausgang eines Mediums wird nie identisch sein mit dem Signal am Eingang. Tatsächlich erfährt ein Signal im Medium...

- Dämpfung (*Attenuation*): das Signal wird abgeschwächt
- Verzerrung (*Distortion*): Eingangs- und Ausgangssignal haben nicht dieselbe Form
- Störungseinflüsse (*Noise*)

Störungen in Form von elektromagnetische Strahlen sind überall vorhanden. Jeder Leiter sendet und empfängt elektromagnetische Wellen (z.B. bei Fernsehsendern, Energieversorgungsleitungen, Motoren, Blitzen, usw..). Die Aufnahme von Störungen muss minimiert werden (*noise-immunity*), ebenso das eigene Aussenden von elektromagnetischen Wellen.

Durch unerwünschte Koppelungseffekte zwischen parallel liegenden Kabeln kann es auch Störungen geben, man spricht dann von Nebensprechen (*crosstalk*). Sehr schlimm ist verständliches Nebensprechen.

In diesem kurzen Bericht werden einige Übertragungsmedien vorgestellt und auf die oben genannten Eigenschaften hin untersucht. Man unterscheidet leitungsgebundene und leitungsungebundene Übertragungsmedien. Funk und Infrarot sind Beispiele für leitungsungebundene Medien. Dieser Bericht befasst sich mit folgenden leitungsgebundenen Medien:

- Metallische Leiter: UTP und Koaxialkabel
- Nichtmetallische Leiter: Lichtwellenleiter

2 UTP und Koaxialkabel

Ein *Unshielded Twisted Pair* (UTP) besteht aus zwei isolierten Drähten, die verdrillt werden. Sie werden aber nicht abgeschirmt. Bei den heute häufig verwendeten UTP's der Kategorie 5 haben die Verdrillungen Abstände von 1.3 bis 2 cm. Die Drähte haben typische Durchmesser von 0.6 mm. UTP wurden um 1880 erstmals als Telefonleitungen verwendet, sie sind das älteste und immer noch häufigste Übertragungsmedium.

Eine UTP-Leitung ist eine *balanced line*: der Aufbau bezüglich der beiden Drähte ist symmetrisch. In der Regel verwendet man Kabel die mehrere solcher Drahtpaare beinhalten (*multi-pair*).

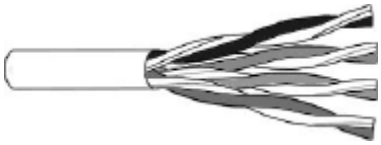


Abb. 1 UTP Kategorie 5

Abb. 2 Koaxialkabel

Ein Koaxialkabel hingegen besteht aus einem Innenleiter, einem Dielektrikum, einem Aussenleiter und einer Aussenisolierung. Er ist bezüglich seiner beiden Leiter nicht symmetrisch und ist deshalb eine *unbalanced line*.

Innen- und Aussenleiter dienen als Hin- bzw. Rückleiter. Das Signal wird im Raum zwischen diesen beiden Leitern (also im Dielektrikum) als elektromagnetisches Signal transportiert. Der Aussenleiter schirmt dieses Signal gegen aussen hin ab. Zur besseren Abschirmung wird er manchmal aus zwei oder noch mehr Schichten aufgebaut. Auch das Dielektrikum wird oft mehrschichtig aufgebaut, um den verschiedenen Anforderungen (Signaltransport, Isolierung zwischen den beiden Leitern, Stabilisierung des Innenleiters konzentrisch zum Aussenleiter) gerecht zu werden. Vielfach kommt Schaumgummi zum Einsatz, da es eine schnelle Ausbreitung des elektromagnetischen Signals ermöglicht. Als Aussenisolierung verwendet man häufig PVC (sehr flexibel und billig) und PE (Polyäthylen, hohe UV-Festigkeit).

2.1 Eigenschaften

Eine Leitung wird als *lange Leitung* bezeichnet, falls ihre Länge nicht vernachlässigbar klein ist gegenüber der Wellenlänge des Übertragungssignals. Beispiel: ein 50 m langer UTP der bei 50 MHz (Wellenlänge = 6 m) verwendet wird. Eine solche Leitung wirkt immer als Empfangs- und Sendeantenne, das Signal wird also gestört werden und kann seinerseits andere Signale stören. Weiters kann man feststellen, dass das Signal bei zunehmender Frequenz stärker geschwächt wird (Tiefpass-Verhalten). Da die Leitung nie verlustlos ist, nimmt die Dämpfung mit der Leiterlänge zu. Auch die Signallaufzeit steigt mit der Leiterlänge an. All diese Faktoren muss man berücksichtigen.

Impedanz: Eine wichtige Kenngrösse ist die Impedanz (Wellenwiderstand) einer Leitung. Diese hängt nicht von der Kabellänge ab, sondern vom Aufbau eines Leiters (d.h. Durchmesser, Materialien, Anordnung der Leiter...). Typische Impedanzen eines Koaxialkabels sind 50 Ohm (digitale Übertragung) und 75 Ohm (analoge Übertragung). UTP-Kabel haben meist 100 Ohm.

Störungsverhalten: Bei UTP gibt es eine sogenannte *common-mode noise rejection*: Durch die Verdrillung werden bei eventuellen Störungen beide Leiter der Störung in gleichem Maße ausgesetzt. Das Störsignal wird also in beiden Leitern gleichermaßen induziert und löscht sich theoretisch aus. In Multi-pair-Kabeln wird durch die Verdrillung das Nebensprechen herabgesetzt (Begrenzung des elektromagnetischen Feldes). Ein UTP stellt somit theoretisch keine Antenne dar, im Gegensatz zu zwei nicht-verdrillten Leitern. Trotzdem: UTP ist unabgeschirmt und hat in der Praxis nicht sehr hohe Störimmunität.

Beim Koaxialkabel wird das Signal durch den Aussenleiter abgeschirmt. Man erreicht bessere Störimmunität als bei UTP.

Dämpfung: Die mit steigender Frequenz zunehmende Dämpfung ist auf den *Skineffekt* zurückzuführen: bei hohen Frequenzen fliesst der Strom zunehmend nur mehr an der Oberfläche des Leiters, d.h. der durchflossene Querschnitt nimmt ab und der Widerstand steigt. Aufgrund der Dämpfung des Signals müssen Regeneratoren eingebaut werden, die das Signal verstärken.

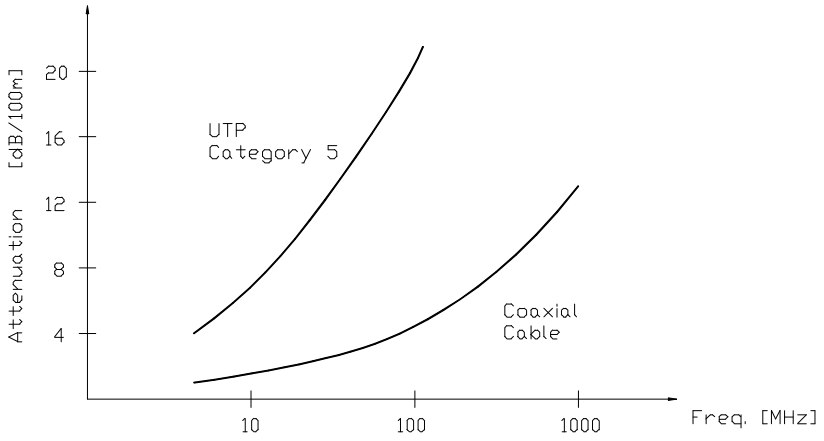


Abb. 3 Dämpfungsverhalten eines UTP Kategorie und eines sehr guten Koaxialkabels

Trotz des besseren Stör- und Dämpfungsverhaltens (Abb. 3) des Koaxialkabels im Vergleich zum UTP, verliert der Koaxialkabel zunehmend an Bedeutung, da man über kurze Distanzen meist UTP und über längere Distanzen Lichtwellenleiter bevorzugt.

Das klassische Ethernet wurde mit Koaxialkabel vernetzt. Für kurze Distanzen, z.B. bei der Verkabelung innerhalb von Gebäuden (z.B. LAN), verwendet man heute meist UTP-Kabel. Sie sind billiger, einfacher zu installieren und bieten bei Abständen bis zu 100 m sehr gute Datenraten von bis zu 100 Mbit (das gilt für UTP der Kategorie 5). UTP höherer Kategorien können sogar für das Gigabit-Ethernet eingesetzt werden.

Auch bei Verbindungen über grössere Distanzen verwendet man heute meist keine Koaxialkabel mehr, denn deren Regeneratorabstände liegen bei wenigen Kilometern und die maximalen Datenraten bei ca. 2 Gbps. Bei noch höheren Frequenzen steigt die Dämpfung stark an. Falls die Signalwellenlängen vergleichbar werden mit dem Durchmesser des Kabels, so verhindern bereits kleinste Unstetigkeitsstellen im Koaxialkabel die Übertragung. Dann ist die Grenzfrequenz des Kabels erreicht.

Für solche Anwendungen über grosse Distanzen (z.B. im Telefonnetz) verwendet man heute nahezu ausschliesslich Lichtwellenleiter.

3 Lichtwellenleiter (LWL)

3.1 Aufbau und Übertragungsprinzip

Bei der LWL-Übertragung verwendet man eine sehr dünne Glasfaser aus Quarz (*optical fibre*), die eingespeistes Licht überträgt. Als Lichtquelle (*light source*) verwendet man häufig Laser. Die Lichtsignale können auf der anderen Seite von einem lichtempfindlichen Sensor (z.B. eine Photodiode, v.a. PIN und Avalanche) erkannt werden. Das ist der Empfänger (*detector*). Dabei bedeutet Licht eine logische `1´, kein Licht bedeutet eine logische `0´. Dies bezeichnet man auch als OOK (*On-Off-Keying*).

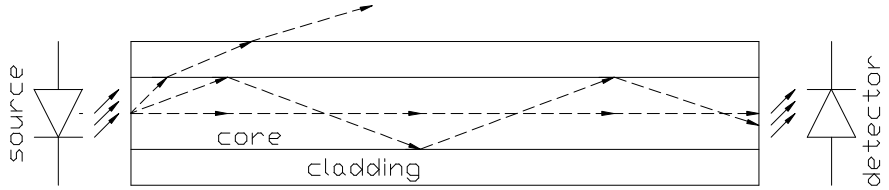


Abb. 4 Prinzip einer LWL-Verbindung mit Stufenindex-Glasfaserkabel

Abb. 4 zeigt den Aufbau eines Stufenindex-Glasfaserkabels, welcher aus einem Kern (*core*), einem Mantel (*cladding*) und einer Beschichtung (*coating*) besteht. Kern und Mantel sind aus Glas. Durch Dotierungen stellt man für den Mantel eine niedrigere Brechzahl als für den Kern ein. Damit werden genügend flach eingestrahlte Lichtstrahlen an der Grenzfläche dieser beiden Medien verlustlos totalreflektiert.

Bei sogenannten Multimode-LWL gibt es für das Licht verschiedene Wege (siehe z.B. die beiden Strahlen in Abb. 4). Da sich für diese Lichtstrahlen unterschiedliche Laufzeiten ergeben, kommt es zu Dispersion: das Eingangssignal erscheint am Ausgang „verschmiert“. Diese Art der Dispersion wird als *Modendispersion* bezeichnet. Sie bestimmt das Tiefpass-Verhalten des LWL und damit die nutzbare Bandbreite.

Man kann zeigen, dass sich bei genügend kleinem Kerndurchmesser nur mehr ein Lichtstrahl (eine Mode) ausbildet. Dadurch kann es keine Modendispersion mehr geben. Solche LWL bezeichnet man als Monomode-LWL. Monomode-LWL sind teurer, ermöglichen aber erst sehr hohe Datenraten und grosse Übertragungsdistanzen.

Bei Monomodekabeln tritt die *chromatische Dispersion* in Erscheinung: da die Lichtquelle nie perfekt monochromatisches Licht aussenden kann, und die Geschwindigkeit der Strahlen von deren Wellenlänge abhängt, gibt es erneut Laufzeitverzerrungen, allerdings wesentlich geringere als bei der Modendispersion.

Eine Möglichkeit, der Dispersion entgegenzuwirken, sind *Solitons*. Das sind Impulse mit einer ganz speziellen Form, die sich auch bei Kollisionen als äusserst resistent erweisen. Im Labor ist man in der Lage, Dispersionen nahezu vollkommen zu eliminieren.

Es gibt auch Kunststoff-LWL, die für Distanzen bis ca. 100 m eingesetzt werden können.

	Multimode-LWL	Monomode-LWL	Kunststoff-LWL
Kerndurchmesser	50	8	980
Manteldurchmesser	125	125	1020

Tab. 1 typische Abmessungen von Lichtwellenleitern in µm

3.2 Eigenschaften

Vorteile der LWL gegenüber Kupferkabeln:

- höhere Bandbreite
- kleinere Durchmesser und geringeres Gewicht
- Immunität vor elektromagnetischen Einflüssen
- galvanische Trennung, es gibt keine Erdungsprobleme
- hohe Abhörsicherheit
- gutes Dämpfungsverhalten

Nachteilig ist die schlechtere Handbarkeit eines LWL, z.B. beim Anschliessen eines LWL oder beim Verschweissen zweier LWL.

Dämpfung gibt es im LWL aufgrund von Absorption und Streuung. Letzteres wird durch Inhomogenitäten im LWL hervorgerufen. Hatten LWL im Jahre 1965 noch 1000 dB/km, erreicht man heute wesentlich tiefere Werte, sodass man Strecken von 120 km ohne Verstärkung zurücklegen kann. Man vergleiche die Werte in Tab. 2 mit denen von Koaxialkabeln (Abb. 3).

Wellenlänge [nm]	Multimode [dB/km]	Monomode [dB/km]
850	2.5	
1300	0.7	0.36
1550		0.22

Tab. 2 gute Dämpfungsbeläge

In der Praxis werden drei Wellenlängenbänder für die Übertragung genutzt, die je ca. 25 THz breit sind. Das 1300 nm und das 1550 nm-Band zeichnen sich durch geringe Dämpfung aus. Das 850 nm-Band hat den Vorteil, dass dort der Laser und die Elektronik aus demselben Material (Galliumarsenid) gefertigt werden können.

3.3 DWDM (Dense Wavelength Division Multiplexing)

Die Einspeisung mehrerer Kanäle in dieselbe Glasfaser durch Benützung verschiedener Wellenlängen (d.h. verschiedener Licht-Farben), sowie die anschliessende Trennung (*Demultiplexing*) der Kanäle am Ausgang ist das Prinzip des DWDM. Man ist heute in der Lage mehr als 100 Kanäle über einen LWL zu übertragen, für die man früher je eine Glasfaser benötigte. Bandbreiten von 10 Gbps und mehr pro Kanal sind heute erreichbar.

DWDM stellt hohe Anforderungen, denn Kanalabstände liegen heute schon unter 1 nm. Laser müssen sehr schmalbandiges und stabiles Licht aussenden, dazu verwendet man Laserdioden mit geregelten Frequenzen. Demultiplexing verlangt sehr schmalbandige Filter mit geringen Verlusten. Bei DWDM kann man auf einem LWL in beide Richtungen übertragen (auf verschiedenen Kanälen).

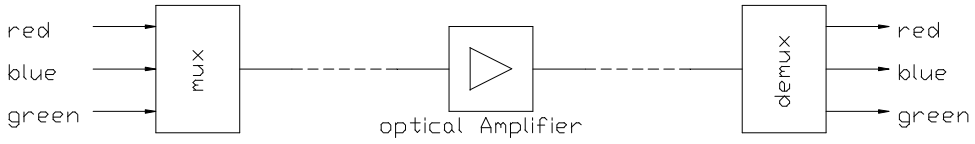


Abb. 5 Prinzip der DWDM-Übertragung

3.4 Optische Verstärker

Die rasante Entwicklung von optischen Verstärkern geht mit dem Streben nach immer grösseren Bandbreiten einher. Optische Verstärker ermöglichen Übertragungen über einige Tausend Kilometer ohne optisch-elektronisch-optische-Umwandlung, was v.a. bei DWDM sehr wichtig ist (Abb. 5). Um für DWDM verwendbar zu sein, muss der Verstärker ein geringes Nebensprechen zwischen den DWDM-Übertragungskanälen aufweisen. Im Vergleich zu einer eigenen elektronischen Verstärkung für jeden Kanal ist der Aufwand bei der Verwendung optischer Verstärker gering.

Bei EDFA (*Erbium Doped Fiber Amplifier*) verwendet man eine Glasfaser, die mit Erbium-Ionen dotiert ist. Durch Laser erregt man die Erbium-Ionen und zwingt sie zur Aufnahme von Photonen (*stimulated absorption*). Nach einer gewissen Zeit senden die Ionen das Photon wieder aus (*spontaneous emission*) und gehen in den Grundstatus zurück. Zuvor können erregte Ionen jedoch durch einfallende LWL-Signale im 1550 nm- Band in den Grundstatus zurückgeschossen werden, die ausgesandten Photonen haben die gleiche Wellenlänge wie das einfallende Signal (*stimulated emission*). Das Signal wird verstärkt.

3.5 Anwendungsbeispiele

- Eine DWDM-Übertragungsstrecke zwischen der ETH-Zürich und Genf ist seit kurzem im Einsatz. Sie bietet 16 Kanäle zu je 2.5 Gbps, später dann 10 Gbps. Die Übertragung erfolgt rein optisch, auf der Strecke werden nur optische Verstärker (EDFA) eingesetzt. Ein Ausbau des Netzes ist bereits geplant.
- Das Transatlantik-Untersee-LWL-Kabel TAT-14 verwendet DWDM auf 4 Glasfasern. Es bietet gesicherte 640 Gbps, die Gesamtkapazität liegt bei 1.3 Tbps.
- Ein Beispiel für ein reines LWL-Netzwerk ist FDDI (*Fiber Distributed Data Interface*), ein Token-Passing-Verfahren mit einem Doppel-LWL-Ring. Es erlaubt 100 Mbps über sehr grosse Distanzen. Im Normalbetrieb wird nur ein LWL-Ring genutzt, bei einer Fehlfunktion nutzt man den zweiten. So bietet FDDI hohe Zuverlässigkeit.
- Das FT-2000 System von Lucent Technologies ist ähnlich aufgebaut wie FDDI, verwendet aber DWDM. Es bietet bis zu 2.488 Gbps. Es zeigt, dass DWDM bald auch über kürzere Distanzen mit Vorteil eingesetzt werden kann.

Das Potential, das in Glasfasern steckt, ist enorm. Bandbreiten von über 50 Tbps sind sicherlich erzielbar. Im Moment bestimmt die Umwandlung von elektrischen in optische Signale die maximale Bandbreite.

Referenzen

1. Beuth/Hanebuth/Kurz: Nachrichtentechnik; Vogel Fachbuch, 2001
2. Couch: Digital and Analog Communication Systems; Prentice Hall, 2001
3. G. Mahlke/ P. Gössing: Lichtwellenleiterkabel; Siemens, 4. Auflage, 1995
4. D. Eberlein: Lichtwellenleiter- Technik; Expert Verlag, 2000
5. http://bwccat.belden.com/Bimages/TechInfo.htm
6. http://www-classes.usc.edu/engr/ee-s/558/sum/edfa.pdf
7. http://www.switch.ch/lan/switchlambda/lambda-article-cw-DE.html
8. http://www.lucent.com/livelink/157489_Brochure.pdf

PPS-Seminar
Grundlagen der Internet-Technologie, WS 01/02

Peer-to-Peer Networking

Schuler Valentin
schulerv@ee.eth.ch
Januar 2002

1 Einführung

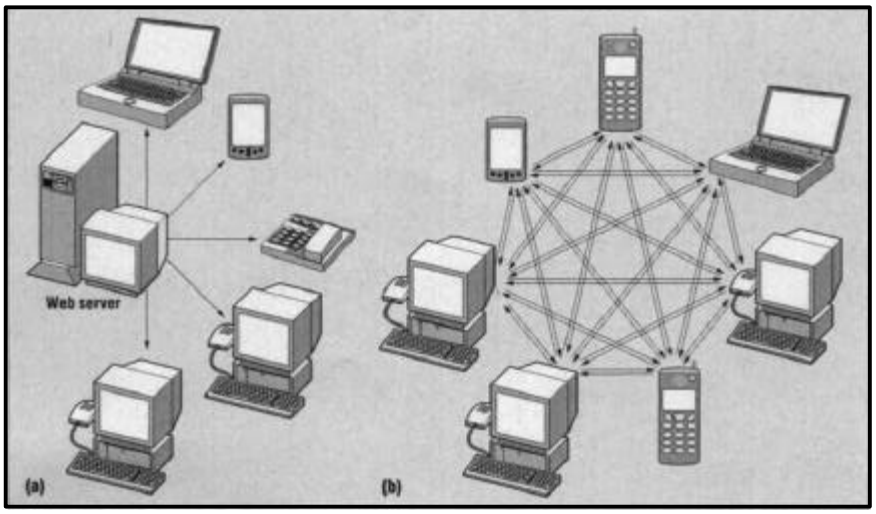
Seit dem riesigen Erfolg von Napster ist Peer-to-Peer (P2P) Networking und deren Entwicklung nicht mehr zu stoppen. Mächtige Firmen wie IBM, HP oder Intel sind daran interessiert und eröffnen Foren, entwickeln Applikationen. Bei Napster wurde ausschliesslich noch auf Filesharing gesetzt, inzwischen beinhalten die Anwendungen weit mehr Funktionen, welche das Internet revolutionieren könnten. Die Zukunft wird es zeigen. In diesem Text soll das grundsätzliche System von P2P, bisherige und auch zukünftige Anwendungsmöglichkeiten erklärt und dargestellt werden., wie auch die Vor- und Nachteile im Vergleich zum traditionellen WWW erläutern.

1.1 Was bedeutet Peer-to-Peer?

Sucht man im Internet nach dem Begriff Peer-to-Peer um eine Definition zu finden, so erhält man eine reiche Auswahl an Links, die aber sehr selten zum eigentlichen Begriff führen. Eher sind es Links zu Hilfeforen, Problemstellungen bezüglich Peer-to-Peer, Fragen von Internetbenutzern, die ihre PCs verkabeln wollen, oder auch häufig P2P Software zum Downloaden wie Gnutella, Morpheus, etc. Selbst bei gefundener Definition ist die Erklärung sehr diffus und weitgestreut bezüglich der Anwendungsmöglichkeiten. Vielleicht ist das Paradigma einfach noch zu unbekannt, und die Möglichkeiten, die es bietet noch zu wenig überblickbar um P2P eindeutig umschreiben zu können.
P2P aber trotzdem kurz definiert, ist ein Netzwerk von gleichen Computern, von denen jeder die Funktion des Servers und des Hosts übernehmen kann. Die dazu möglichen Anwendungsmöglichkeiten sind sehr vielseitig: Filesharing, distributed Computing, Collaboration, etc. nur mal die bedeutendsten erwähnt. Diese werden später noch genauer erklärt. Wie bei allen Netzwerken sind zwei Komponenten wichtig um als Netzwerk funktionieren zu können. Hardware, gemeint sind die Computer, die hardwaremässige Verbindung via Leitungen, Kabel, Router und eine gemeinsame Sprache, das Protokoll, damit sich die vernetzten Computer auch verständigen können.

1.1.1 P2P Hardware

Peer-to-Peer könnte man übersetzen mit „Kollege zu Kollege“ oder übertragen „gleich zu gleich“. Dies erklärt, weshalb dieser Begriff teilweise auch benutzt wird bei Computernetzwerken, die durch Ethernetkarten oder Nullmodemkabeln verbunden sind. Es handelt sich dabei ja einfach um eine Vernetzung von „Gleichen“. Der grosse Unterschied ist, wie die Darstellung zeigt, dass bei P2P kein zentraler Server benötigt wird, sondern die User direkt miteinander kommunizieren und Daten austauschen können.



(a) traditionelles Internet-Modell. Die User können Daten, die auf dem zentralen Web-Server gespeichert sind abrufen.

(b) P2P verbindet die am Netzwerk angemeldeten Computer direkt miteinander.

Abb. 1.1 Internetmodelle

1.1.2 P2P Protokoll

Protokolle sind in der Netzwerktechnik das zentrale Konzept. Sie legen fest, wie die Kontaktaufnahme erfolgt, in welcher Reihenfolge die Datenpakete verschickt werden, etc. Ein Protokoll ist eine gemeinsame Sprache der verbundenen Computer, ohne diese nicht kommuniziert werden kann.

Das wohl bekannteste und wichtigste Protokoll im Web ist das Internetprotokoll (IP), welches das allgemeine Kommunikationsschema definiert. Moderne Protokolle bauen in der Regel aufeinander auf. So setzt zum Beispiel das Transport Control Protocol (TCP) auf IP auf. Das WWW (http-Protokoll) wird häufig als Einbahnstrasse bezeichnet, da abgesehen von der Interaktivität eine Verbindung vom Client zum Server angefordert und im Wesentlichen die Daten in eine Richtung, also wieder zum Client transportiert werden. P2P erfordert einen anderen Protokolltyp, bei dem es keine feste Aufgabenteilung gibt. Da die Daten bidirektional gesendet und angefordert werden und jeder Netzwerkteilnehmer sowohl als Server als auch als Client fungieren kann, wird ein symmetrisches Protokoll benötigt. Inzwischen sind schon verschiedene Kunstworte wie Servent oder Clerver entstanden, die diesen Sachverhalt noch unterstreichen.

2 Informationssuche

Ein grosser Unterschied zum WWW ist die Suche von Informationen unter P2P. Da es keine spezifischen Suchmaschinen gibt, die wissen, wo sich was befindet und dem Anwender den Link zur gesuchten Page zurückgibt, muss die Suche nach einem anderen System funktionieren. Diese unterscheiden sich noch einmal geringfügig von Programm zu Programm.

Die Informationssuche könnte man vergleichen mit einem Wanderer der in einer unbekannten Gegend ohne Karte jemanden sucht. Auf dem Weg trifft er immer wieder andere Personen, die eventuell eine genauere Richtung kennen oder von Personen wissen, die genauer darüber Bescheid wissen. Mit jedem empfohlenen Weg kommt er (im Mittel) näher ans Ziel, und die Antworten werden auch immer konkreter.

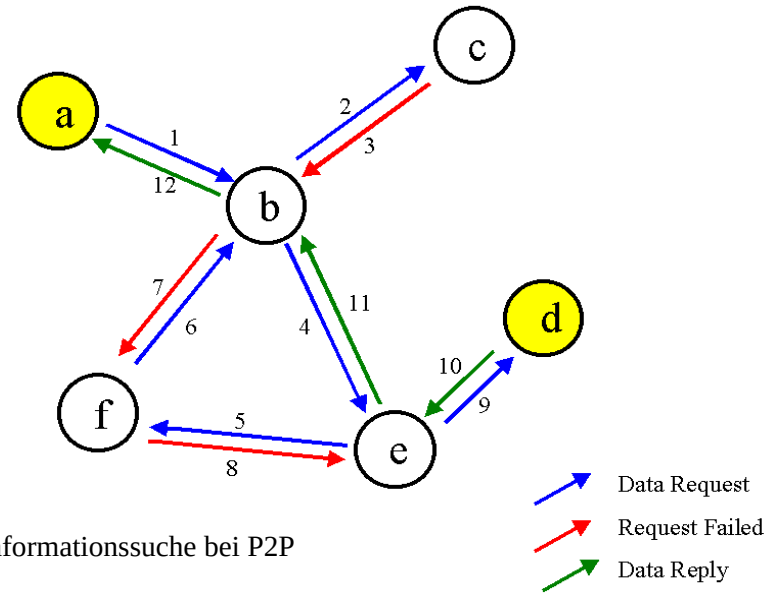


Abbildung 2.1: Informationssuche bei P2P

Ein vereinfachtes Modell sähe folgendermassen aus:
Der suchende Peer (a) fragt bei seinem nächsten Nachbarn (b) nach einer bestimmten Datei. Da dieser die gesuchte Datei nicht besitzt fragt er wiederum seinen nächsten Nachbarn (c). Dieser antwortet mit einer Request-Failed Nachricht, da er die Datei nicht besitzt und auch keine weiteren Nachbarn hat, die er fragen könnte. Daher fragt Peer (b) seinen zweiten Nachbarn (e) , der zwar die gesuchte Datei ebenfalls nicht besitzt, aber noch weitere benachbarte Peers hat. (e) fragt (f) und dieser wieder (b). Der Zyklus, der hier entsteht, wird dadurch gelöscht, das (b) die Anfrage löscht und ein Request-Failed zurückgibt, da er dieselbe Anfrage bereits von (a) erhalten hat. Nachdem (e) von (f) erfahren hat, dass die Information auf diesem Zweig des Netzwerks nicht zu finden ist, fragt er seinen zweiten Nachbarn (d). Dieser besitzt nun endlich die gesuchte Datei und schickt sie nun via (e) und (b) an den suchenden Peer (a).

Die wirkliche Suche läuft in einem komplexeren Rahmen ab. Um die Suchzeit zu verkürzen, speichern die Peers bestimmte Informationen, die der Suche hilfreich sind. So kann bei einer ähnlichen oder gleichen Suche eine genauere Richtung angegeben werden, wo die Datei eventuell zu finden ist, wie dies beim Wanderer auch der Fall ist. Da dieser Speicher beschränkt ist, wird er fortlaufend mit neuen Daten gefüllt. Weiter besitzt jede Anfrage eine TTL (Time to live), welche die Existenzzeit beschränkt. Bei jedem Durchlaufen eines Knotens wird die TTL um eins reduziert, bis die Anfrage schliesslich gelöscht wird. So wird verhindert, dass das P2P-Netzwerk von rotierenden Anfragen blockiert wird. Um die Suche zu beschleunigen wird bei den meisten P2P-Anwendungen die Anfrage jeweils an alle benachbarten Peers geschickt, und diese schicken sie wiederum an die Nachbarn. Die Anfrage verbreitet sich somit innert kürzester Zeit explosionsartig.
Bei manchen Programmen wird nach erfolgreicher Suche nicht die Datei zurückgeschickt, sondern nur die genaue Adresse des Peers, der das Dokument besitzt. Der suchende Peer verbindet sich dann direkt mit diesem und kopiert die Datei.

3 Anwendungsklassen

Nebst Filesharing, das vor allem durch Napster populär wurde gibt es eine Reihe weiterer Anwendungsbereiche, die vor allem für Firmen und grössere Projekte interessant sind. Oft wird für Anwendungen der Begriff Peer-to-Peer benutzt, obwohl es mehr mit dem traditionellen Internetmodell mehr Verwandtschaft aufweist, da es einen zentralen Server benötigt.

3.1 File Sharing

3.1.1 Napster

Obwohl Napster kein reines P2P-System ist, das es eine zentrale Datenbank, der vorhandenen Daten hat, löste es den enormen P2P Boom aus. Mit Napster konnten nur mp3-Files getauscht werden, was meistens eine Verletzung des Copyrights eines Musikstücks zur Folge hatte. Durch den Prozess wurde Napster in der Öffentlichkeit bekannt gemacht. Angriffspunkt war vor allem die zentrale Datenbank der mp3-Files.

3.1.2 Gnutella

Gnutella wurde Anfangs März 2000 von WinAmp-Entwickler Nullsoft herausgegeben und wurde kurze Zeit später von AOL übernommen. Es beschränkte sich nicht wie Napster auf mp3-Files, sondern bot allgemeines Filesharing an. Die rechtlichen Probleme bei Napster veranlasste AOL das Projekt einzustellen.

Gnutella arbeitet nicht mit einem zentralen Datenbank-Server und ist daher ein echtes P2P-Programm. Nach dem Zurückzug von Gnutella entstanden in kurzer Zeit diverse Gnutella-Clones (z.B. Morpheus, Imesh, KaZaA), die alle auf dem Gnutella-Protokoll basieren und allgemeines Filesharing anbieten

3.2 Distributed Computing

Wie der Name schon sagt, ist die Idee dahinter, Rechenarbeit auf mehrere ans Netz angeschlossene Computer zu verteilen. Die zu bearbeitenden Daten werden paketweise an die Clients verschickt und nach getaner Arbeit werden sie automatisch wieder an den Server zurückgesandt. Da es eine klare Rollenunterscheidung (Client / Server) gibt, kann hier nicht von einer echten P2P-Anwendung gesprochen werden. Der Begriff P2P wird also sehr verschieden ausgelegt.

3.2.1 Popular Power

Der Client verbindet sich, wenn der Computer unbeschäftigt ist um vom Server eine Aufgabe zu erhalten.. Diese erledigt er und schickt das Resultat automatisch zurück. Als Besitzer des Clients kann man bei diesem Projekt entscheiden, ob man kommerzielle oder unkommerzielle Projekte unterstützen will. Bei ersterem wird man von Popular Power entsprechende der aufgebrachten Rechenzeit entlöhnt.

3.2.2 SETI@home

Dieses Projekt benutzt weltweit unbeschäftigte Computer, deren Besitzer einen Client installiert haben, um nach ausserirdischer Intelligenz zu suchen. Die 500KB-Pakete werden von dem im Hintergrund laufenden Client analysiert und ausgewertet und die Ergebnisse an den SETI@home-Server zurückübertragen. Mitte Dezember 2000 nahmen daran über 2.5 Mio. Benutzer teil und überschritt die Marke von einer halben Million CPU-Zeit-Jahren. Die P2P-Eigenschaften sind wie schon bei Popular Power fraglich.

3.3 Collaboration

Immer häufiger arbeitet ein lokal verstreutes Team an einem Projekt gemeinsam. Da diese Arbeitsweise einen enormen Kommunikationsaufwand mit sich bringt, sind geeignete P2P-Andwendungen notwendig.

3.3.1 Groove

Dieses Programm beinhaltet alle wichtigen Kommunikationsmöglichkeiten um sich multimedial im Netz verständigen zu können:

- Instant Messaging
- Live voice
- File sharing
- Free-form drawing
- Video functions.

Da es keinen zentralen Server benötigt, kann hier von einer echten Peer-to-Peer Anwendung gesprochen werden. Zudem ist es sehr einfach zu verwalten, da die Rolle des Administrators wegfällt.

4 Vorteile - Nachteile von P2P

Da sich P2P stark vom traditionellen Internet-Modell unterscheidet und sich dahinter eine andere Architektur mit verschiedenen Eigenschaften verbirgt, ergeben sich zum einen neue Schwierigkeiten und andersherum lassen sich bisherige Probleme oder Projekte dadurch verwirklichen.

4.1 Vorteile

4.1.1 Ausfallssicherheit

Da Daten nicht mehr zentral auf einem Server liegen, sondern meist verbreitet auf mehreren Clients verfügbar sind, wiegt es weniger schwer bei einem Ausfall eines einzelnen Clients, da die anderen sofort die Rolle dessen übernehmen können.

4.1.2 Anonymität

In einem Peer-to-Peer Netzwerk Anonymität sowohl für den Informationsanbieter wie auch für den Nachfrager zu wahren ist recht einfach. Eine Anfrage nach einer Information oder bestimmten Datei wird von Peer zu Peer weitergegeben. In jedem Zwischenknoten ist nur bekannt, von welchem Nachbarn (Peer) die Anfrage kam, um die Antwort an ihn zurückgeben zu können. Wenn die Datei bei einem Peer gefunden wurde, wird die Antwort wieder auf dem selben Weg zurückgegeben. Nach Rückgabe der Antwort geht diese Zuordnungsinformation verloren, somit ist also auch der Weg, der die Anfrage zurückgelegt hat nicht mehr bekannt.

4.1.3 Adaptivität

Dies bedeutet, dass das System mit der sich ständig ändernden Netzstruktur zurechtkommt. Ein Peer-to-Peer Netzwerk ist also fast beliebig erweiterbar, da es keinen zentralen Administrationspunkt gibt, der die genaue Netzstruktur kennen muss. Diese adaptive Eigenschaft ist auch notwendig, um z.B. ein Filesharing Programm effizient einsetzen zu es können, das jedem Internetuser möglich sein soll, sich beim Datentausch zu beteiligen.

4.1.4 Echtzeitsuche

Die Suche nach Informationen, Dateien, Bilder, etc. findet über P2P in Echtzeit statt. Das heisst, dass das Suchresultat nur diejenigen Dateien anzeigt, die momentan erhältlich und verfügbar sind. Die Suche gestaltet sich somit effizienter, da keine toten Links angezeigt werden, was beim traditionellen Internetkonzept öfters vorkam.

4.2 Nachteile

4.2.1 Kommunikationsaufwand

Um eine Datei ausfindig zu machen, ist ein sehr grosser Kommunikationsaufwand notwendig, da jede Suchanfrage von Peer zu Peer weitergeleitet, ausgewertet und dann bei Erfolg das Resultat bei Erfolg an den Absender geschickt werden muss. Wenn der Peer die Datei nicht besitzt, wird die Anfrage weitergeleitet bis entweder die TTL abgelaufen ist, oder ein Peer fündig wurde.

Durch den zentralen Datenbankserver war bei Napster viel weniger Aufwand notwendig. Die Anfrage gelangte zum Napsterserver, eine Antwort kam zurück und anschliessend konnte der Peer die Anfrage direkt an einen Client senden, der diese Datei besitzt.

4.2.2 Tausch von illegalen Daten

Da die Daten nirgends auf einem Server abgelegt werden, können sie auch nicht auf Legalität überprüft werden. So kommt es leider häufig vor, dass verbotene Dateien wie Kinderpornographie, Nationalsozialistische Inhalte, etc. via P2P verbreitet werden. Besitzer von solchen Dateien sind zudem auch sehr schwer ausfindig zu machen.

4.2.3 Skalierbarkeit

Wie schon geschrieben, kann sich jeder in ein öffentliches P2P-Netzwerk einloggen. Durch stetig steigendes Interesse vergrössern sich die Netzwerke laufend, was unter anderem zu Performance-Problemen führen kann. Viele Peers werden miteinander verbunden, auch wenn nie eine Anfrage sie erreichen kann, da die TTL schon vorher abgelaufen ist und sich die Anfrage gelöscht hat. So werden viele P2P-Netzwerke, welche als ein einzig grosses keinen Sinn macht, in mehrere Segmente unterteilt. So entscheidet unter anderem auch das Glück, ob man beim anmelden in ein grosses Netzwerksegment gelangt und dadurch die Chance dann grösser ist, eine Datei zu finden.

5 JXTA

Durch den riesigen Boom von P2P sind in kurzer Zeit verschiedene Applikationen entstanden, die jeweils ein unterschiedliches Protokoll verwenden. Mit JXTA (vom englischen „juxtapose: nebeneinanderstellen“), soll nun auch da ein Standard geschaffen werden.

5.1 Was beinhaltet JXTA?

Wie einleitend geschrieben, soll JXTA einen Standard in die relativ ungeordnete und unübersichtliche Welt der P2P-Protokolle bringen. Unabhängig von einem bestimmten Betriebsystem ermöglichen die 6 dazugehörenden Protokolle alle wichtigen Funktionen, die eine P2P-Applikation benötigt. Wirklich neu an JXTA ist die Unterbringung all dieser Protokolle in einem API (Application Program Interface), das als Schnittstelle verschiedener P2P-Anwendungen dienen kann. Der Programmierer schreibt „nur“ noch die Applikation mit implementiertem API.

5.2 JXTA-Protokolle

Peer Discovery Protocol (PDP)
Grundkonzept des JXTA-Netzes: Mit dem PDP können Peers ihre Ressourcen (Content, Peer-Gruppen, Services, Pipes, etc. anderen Peers bekanntmachen und Ressourcen anderer Peers finden.

Peer Resolver Protocol (PRP)
Das PRP ermöglicht Anfragen an die durch das PDP entdeckten Peers. Es entspricht etwa dem Anfragemechanismus bei Gnutella oder Morpheus. Jeder Anfrage wird eine eindeutige ID zugeordnet, die mit der Antwort zurückgeschickt wird.

Peer Information Protocol (PIP)
Über das PIP kann ein Peer Statusinformationen erfragen. Ermöglicht gegenseitige Kontrolle.

Peer Membership Protocol (PMP)
Das PMP ermöglicht das Bilden von Peer-Gruppen, Betreten und Verlassen mittels PDP gefundener Gruppen. Ein Peer kann zu mehreren Gruppen gehören.

Pipe Binding Protocol (PBP)
Ein Peer stellt mit Hilfe des PBP eine Verbindung zu einem anderen Peer her. Diese Pipes stellen virtuelle Verbindungen zwischen Peers dar.

Peer Endpoint Protocol (PEP)
Im Gegensatz zum PBP stellt das PEP einen Mechanismus zur Verfügung, der eine Route zwischen zwei nicht direkt miteinander verbunden Peers findet.

6 Literaturverzeichnis

1. Ben Delaney. The Power of P2P.
   http://www.computer.org/multimedia/homepage/p2p.htm

2. Clay Shirky, What is P2P...And what isn't. O'Reilly Network.
   http://www.openp2p.com/lpt/a/p2p/2000/11/24/shriky1-whatisp2p.html

3. Geofrey Fox. Peer-to-Per Networks, Web Computing, 2001.

4. Giovanni Flammia, IEEE intelligent Systems, Peer-to-Peer ist not for Everyone, 2001.
   http://www.computer.org/intelligent

5. Gnutella
   http://www.gnutellanews.com/

6. JXTA v1.0 Protocols Specification,Revision 1.1.1. Sund Microsystems, Inc.2001.
   http://www.platform.jxta.org/spec/v1.0/JXTAProtocols.pdf
   http://www.jxta.org

7. P2P Netzte, Die andere Art von Suchmaschinen
   http://www.at-web.de/p2p/index.htm