Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Burkhard Stiller, Jan Gerke (Edt.)

PPS-Seminar: Grundlagen der Internet-Technologie 2

TIK-Report
Nr. 104, Februar 2001

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Burkhard Stiller, Jan Gerke (Edt.):
PPS-Seminar: Grundlagen der Internet-Technologie 2
Februar 2000
Version 1
TIK-Report Nr. 104

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 2

Einleitung

Nun bereits in der zweiten Auflage wurden Studierende des Departements für Elektrotechnik, die die Grundlagen und erste wichtige Begriffe eines weitverbreiteten Netzwerkes – dem Internet – erlernen möchten, in diesem Seminar des WS 00/01 angesprochen.

Das Seminar vermittelte dabei wesentliche Grundlagen für die Kommunikationstechnologie am Beispiel des Internet. Dabei wurden u.a. die folgenden Fragen aufgeworfen und Antworten hierzu gegeben: was ist ein Netzwerk, was bezweckt die Adressierung, wie funktioniert die E-Mail, welche Protokolle und Sprachen gibt es im Web, was ist IP-Telefonie, wie werden drahtlose Web-Zugriffe möglich? Ferner verarbeitete das Seminar diese Grundlagen an weiterführenden Details am gleichen Beispiel: was ist die Internet-Architektur, welche Protokolle gibt es, welche Rolle spielt die nächste Generation der Internet-Protokolle, welche Entwicklungstendenzen zeigen sich? Insbesonders wurden einige Themen behandelt, die mit dem Auftritt des Internet als Daten- und Informationspräsentationsmedium zusammenhängen, u.a. das HTTP-Protokoll, die Beschreibungssprache HTML, die Einbindung multimedialer Daten via SMIL sowie die Datenstrukturierungssprache XML.

Ablauf

Die Studierenden erarbeiteten wie im vergangenen Sommersemester dieses Mal zu elf 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. Neben dieser schriftlichen Arbeit hielt jeder Studierende einen Vortrag im Rahmen des Seminares, welcher zum Ziel hatte, in 15 Minuten das erarbeitete Wissen den Zuhörern nahezubringen, zu erläutern und zeigen zu können, daß selbständig erarbeitetes Wissen gut aufbereitet und verständlich präsentiert werden kann. Ein nachfolgende Diskussions- und Fragephase erlaubte das interaktive Behandeln von Unklarheiten, offenen Fragen sowie die Verküfung von den verschiedenen Themen.

Vorträge, Referenten und Titel

Vortrag 1:	Jens Temintzer:	Grundlagen des Internet
Vortrag 2:	Peter Niggli:	Netzwerktechnologien für das Internet
Vortrag 3:	Sarah Rüdiger:	IP, Adressierung und Routing im Internet
Vortrag 4:	Reto Zürcher:	IPng – Die nächste Generation
Vortrag 5:	Marco Graf:	Das HTTP-Protokoll des Web
Vortrag 6:	Samuel Zimmerli:	Die Beschreibungssprache HTML
Vortrag 7:	Lukas Haemmerle:	Die Datenstrukturierungssprache XML
Vortrag 8:	Fabio Pezzani:	Multimedia-Unterstützung im Web – SMIL
Vortrag 9:	Thomas Zaugg:	Elektronische Post im Internet
Vortrag 10:	Luigi Scoca:	Sichere Kommunikation – SSL, SHTTP
Vortrag 11:	Alain Randiramora:	Drahtlose Kommunikation – WAP

Grundlagen des Internet

Eine Zusammenfassung des Vortrages vom 21. 11. 2000,
im Rahmen des PPS-Seminars Grundlagen der Internet-Technologie

Verfasser:
Jens Temnitzer
temnitje@ee.ethz.ch

Betreuer:
Jan Gerke
gerke@tik.ee.ethz.ch

Prof. Dr. Burkhard Stiller
stiller@tik.ee.ethz.ch

1. Einleitung - Das Internet

Seit einiger Zeit hört man immer mehr von diesem die ganze Welt umspannenden Netzes. Doch was ist es genau? Wie funktioniert es? Wer entwickelte es und wie ist es aufgebaut? An solche Fragen denkt man meist gar nicht! In der heutigen Zeit ist es allgegenwärtig und nicht mehr aus dem Alltag weg zu denken. War vor wenigen Jahren noch die Faxnummer wichtig, so ist es heute die URL der Homepage und die E-Mail-Adresse. Die Welt zieht sich zusammen zu einem »Global Village«. Es ist egal, ob jemand mit John aus Los Angeles, mit Wu Yung aus Schanghai oder mit Juunta aus Ammassalik in Grönland kommunizieren will. In Kathmandu auf Trekkingurlaub, ist es möglich per Internet in der NZZ nachlesen, was in Zürich gerade los ist. Auch für Schule und Beruf hat sich einiges geändert. Sucht man einen Artikel über die neuste Errungenschaft in der Medizin oder so, schaut man zuerst einmal im Internet, bevor die Bibliothek an der Reihe ist. Wenn einer mit meinem Geschäftskollegen aus New York etwas besprechen möchte, bietet sich eine Videokonferenz an. Sogar ins Privatleben hält das Internet immer mehr Einzug. Es ist doch viel einfacher das neue Auto im Internet auszusuchen, oder sich die Pizza über das Web zu bestellen. »The world at your fingertips!« Mit dem Internet sind derzeit mehr als 70'000'000 Rechner miteinander verbunden, über welche man über 1'000'000'000 WWW-Seiten betrachten kann.

Der Phantasie sind keine Grenzen gesetzt, wenn man wissen will, was man alles mit dem Internet machen kann:

- E-Commerce: Reisen buchen
- Televorlesung: »Virtuelle Hörsäle«
- Internet-Telefonie: Weltweit zum Ortstarif telefonieren
- Unterhaltung: Spiele, Chat, Musik, Filme, RealAudio,...

Doch auch immer mehr wird negatives über das Internet erzählt. Vielfach kann man nicht genau beurteilen, wie gut die Information im Netz ist. Stimmen die Aussagen und Thesen? Über E-Mail wird viel Blödsinn verschickt. Viele Firmen haben diese kostengünstige Möglichkeit für Werbezwecke entdeckt. Viren lassen sich sehr gut über das Netz verbreiten, wie man kürzlich wieder einmal mitbekommen hat. Leider hört man auch des öfteren von pornographischem Material, welches im Internet publik gemacht wird. Eine Suche nach einer bestimmten Information ist auch nicht immer einfach. Auch mittels einer der vielen Suchmaschinen findet man nicht immer genau das, was man sich eigentlich erhofft hat.

Im Folgenden möchte ich zuerst einen Einblick in die Geschichte geben, anschliessend werde ich etwas über die Funktionsweise und den Aufbau des Internets sagen.

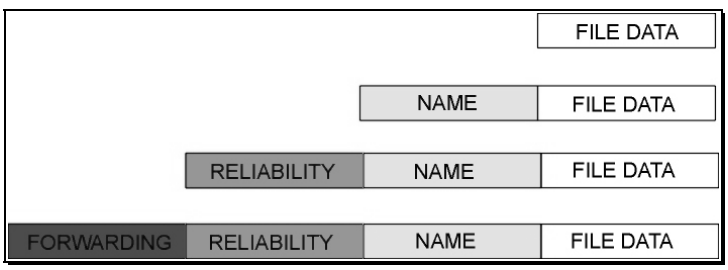
2. Geschichte

Wann genau das Internet erfunden wurde, ist schwierig zu sagen. Die Wurzeln des Internets reichen bereits in die sechziger Jahre zurück. Am 2. September 1962 wurde im Labor von Leonard Kleinrock an der Universität von Kalifornien in Los Angeles der erste Computer an einen Interface Message Processor (IMP) angeschlossen. Der IMP war ein mächtiger Klotz von einem Spezialrechner, der mittels eines Kranes in das Labor gehoben wurde. Seine einzige Aufgabe war es, Daten zu senden und zu empfangen. Er sollte ohne Unterbruch laufen, was zu der Zeit schon etwas spezielles war, mussten doch solche Rechner noch jede Woche für mehrere Stunden gewartet werden! Damals wollte das US-Verteidigungsministerium ein Netzwerk aufbauen, welches ver-

schiedene Rechner im Land verband. Es sollte auch noch nach einer Katastrophe funktionieren, wenn Teile des Netzwerkes, zum Beispiel durch eine Atombombe, zerstört worden waren. Dazu musste es dezentral organisiert sein. Das aufgebaute Netzwerk nannte man ARPANET (ARPA = Advanced Research Projects Agency). Seit den späten sechziger Jahren wurde das Projekt in USA staatlich unterstützt. Im Jahre 1968 wurden drei Rechner für einen Datenaustausch zusammengeschlossen. Die erste öffentliche Demonstration wurde Anfang der siebziger Jahre durchgeführt. 1971 wurde das ARPANET mit 15 unermüdlichen IMPs erstmals in der Öffentlichkeit vorgestellt. Mitte der siebziger Jahre wurde die Verwaltung dann an das »Department of Defence« übergeben. Man merkte bald, dass die Arbeiten nicht nur für den militärischen Einsatz praktisch waren. Auch ausserhalb des Militärs wurde das Netz genutzt und weiterentwickelt. Aufgrund dessen, erfolgte Anfang der achtziger Jahre die Aufteilung von ARPANET in zwei getrennte Bereiche, einen militärischen (MILNET) und einen nichtmilitärischen (ARPANET). Schon 1972 führte Ray Tomlinson den »Klammeraffen« @ als Teil der User-Adressen eines Programms ein, mit dem sich Nachrichten verschicken liessen. Er wählte dieses Zeichen, weil er es auf seinem 33-Tasten-Keyboard am wenigsten benützte. Am 22. November 1977 wurden drei unterschiedliche Netzwerke zusammen geschaltet. Am ersten Januar 1983 das TCP/IP (TCP=Transmission Control Protocol, IP = Internet Protocol; beides Protokolle, die für den Datentransport nötig sind) in den offiziellen Stand erhoben. Offiziell ging das Internet erst 1986 aus dem ARPANET hervor. Im Jahre 1991 wurde vom CERN (European Laboratory for Particle Physics) in Genf erstmals das World Wide Web (WWW) öffentlich vorgestellt. Noch im selben Jahr wurde die Software verbreitet, hauptsächlich auf CERN-Maschinen. Sie nutzte das Web als neue Infrastruktur zur Bereitstellung und zum Zugriff auf Informationen, denn hierfür wurde das Web anfangs hauptsächlich benutzt. Eine der wichtigsten Entwicklungen, die zum Erfolg des Web führten, war 1993 die Bereitstellung eines komfortablen und leistungsfähigen Browsers (eine Alpha-Version von NCSAs *Mosaic for X*, ein Vorgänger des *Netscape*). Erst jetzt nahm auch die Öffentlichkeit vom Web Kenntnis, mit Artikeln in der *New Yorks Times* und dem *Ecomist*. Im August 1995 veröffentlichte *Microsoft* den *Internet Explorer* in der Version 1.0. Damit begann der Konkurrenzkampf zwischen *Microsoft* und *Netscape*. Obwohl sich das Web seit 1990 enorm weiterentwickelt hat, ist seine grundsätzliche Architektur unverändert geblieben.

3. Funktionsweise

Da das ganze Internet viel zu komplex ist, um es einfach zu managen, wird es in mehrere Teile unterteilt. Es kann so auch viel besser überblickt werden. Als Beispiel kann man das mit dem Verkauf von CDs vergleichen. Für das Abspielen der Musik ist eigentlich nur die CD selber nötig. Doch um die CD geschützt lagern zu können, wird sie in einer Box geliefert. Damit der Transporteur nicht jede CD einzeln transportieren muss, werden immer gleich mehrere CDs in eine grosse Schachtel verpackt. Die Schachteln mit den CDs werden wiederum in grossen Containern verschifft. Die einzelnen Stufen der Verpackungen können als Schichten betrachtet werden. Durch diese verschiedenen Schichten wir jedes Problem einzeln behoben. Dem Transporteur ist es völlig egal, was sich im Karton befindet, er ist nur für den schnellen Transport derselben verantwortlich. Nur eine CD alleine, jetzt auch ohne CD-Hülle, kann man gar nicht verschicken. Man kann ja nicht einmal eine Adresse darauf schreiben, ohne Hülle wird sie den Transport mit grösster Wahrscheinlichkeit nicht überstehen. Mit den Datenpaketen, die über das Internet verschickt werden, ist es sehr ähnlich.



Vereinfacht gesagt, besteht das Internet aus verschiedenen Kabeln und durch sie verbundenen Rechnern. Durch diese können nur einzelne Bits verschickt werden. Im Grunde genommen ist ein File auch nichts anderes als eine Aneinanderreihung einzelner Bits. Um das File verschicken zu können, braucht es einen Namen, vielleicht sogar ein Zielverzeichnis. Diese Information muss ebenfalls als Segment einzelner Bits mitgeschickt werden. Sie werden den Bits des Files vorangestellt. Weiter müssen Netzwerkdesigner auch für einen zuverlässigen Transfer der Daten besorgt sein. So wird jedem Datenpaket noch mehr Information vorangestellt, mittels welcher man allfällig auftretende Fehler aufspüren und sogar korrigieren kann. Jetzt muss nur noch mitgeteilt werden, wohin das File überhaupt geschickt werden soll. Diese Information wird ganz an den Anfang gestellt. Beim Ethernet (lokales Netzwerk, bei dem die Computer mittels eines Koaxialkabels untereinander verbunden sind) wird ganz an den Anfang noch Information beigefügt, um den Anfang und das Ende der jeweiligen »Bit-Kolonne« zu kennzeichnen. Wie anhand des CD-Beispiels gezeigt wurde, muss sich jeder Ingenieur nur um seinen eigenen Bereich kümmern. Die Isolation der einzelnen Schichten wirkt sich positiv auf die Flexibilität der TCP/IP-Protokolle aus. Es können so einzelne Schichten verbessert werden, ohne das dadurch die anderen Schichten beeinflusst werden. Somit muss man auch nicht wissen, wie diese anderen Schichten funktionieren.

4. Verbindung zum Internet

Eine Verbindung zum Internet kann auf zwei Arten erfolgen. Wenn ein Computer ausschliesslich als *Client* fungiert, ist es nicht nötig, eine ständige Verbindung zum Netz aufrecht zu erhalten. Es reicht aus, die Verbindung nur dann herzustellen, wenn ein Server im Internet kontaktiert werden soll. Die gängigste Lösung für diesen Ansatz ist eine Modemverbindung.
Wenn der Computer, der mit dem Internet verbunden werden soll, Server-Funktionalität hat, muss er eine ständige Verbindung zum Internet bekommen. Theoretisch wäre es möglich, auch den Server mittels Modem und Telefonleitung zu betreiben, doch das käme auf die Dauer teuer zu stehen. Hier werden Standleitungen benutzt.
Um sich ins Internet einwählen zu können, benötigt man auch noch eine Adresse. Dies sind IP-Adressen. Sie bestehen aus einer 32-Bit-Zahl, mit der man einen Computer global eindeutig bezeichnet. Im Zusammenhang mit dem Web sind allerdings Computernahmen (wie etwa www.ee.ethz.ch) wesentlich gebräuchlicher als IP-Adressen. Namen im Internet sind sogenannte DNS-Namen. Das Domain Name System (DNS) ist vereinfacht gesagt wie ein Telefonbuch organisiert, welches die IP-Adressen auf die DNS-Namen abbildet und umgekehrt die IP-Adressen auf die DNS-Namen abbildet.

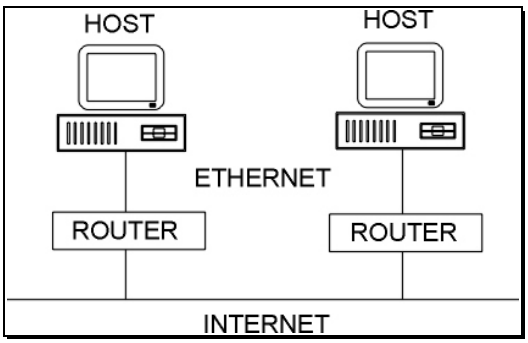
5. Hierarchien

Wie die erwähnten Protokoll-Schichten sind die Hierarchien ein anderes abstraktes Konzept, welches hilft, Computernetzwerke zu organisieren. Als Beispiel soll das globale Telefonnetz betrachtet werden. Wenn die Telefonnummern keine Hierarchie hätten, dass heisst, kein System, wäre es sehr schwierig, für jeden Telefonanschluss eine neue Nummer zu finden. Die Hierarchie macht das ganz einfach. Die Hierarchie unserer Telefonnummern, basiert auf der Geographie. Will jemand von Buenos Aires aus jemand in Würenlingen anrufen, muss er folgende Nummer wählen: 0041 56 281 14 02. Durch die 0041 wird das benutzte Telefon mit der Schweiz verbunden, mit der 56 mit dem Kanton Aargau und mit der 281 mit Würenlingen. Die restlichen Ziffern dienen nun dazu, die Leitung innerhalb des Dorfes zu verbinden. Soll jetzt in Würenlingen ein neues Telefon

angeschlossen werden, muss man sich bei der Vergabe der neuen Nummer nur um die letzten Ziffern kümmern. Dieses Problem der Wegwahl stellt sich in Bezug auf die in Kapitel 3 erwähnten IP-Adressen ebenfalls und wird durch Hierarchien gelöst.

6. Hosts und Routers

Das TCP/IP-Protokoll kombiniert unter anderem verschiedene Systeme zu einem einheitlichen Netzwerk. Das TCP/IP System kann als Host und Router handeln. Systeme, die gerade Nachrichten versenden oder erhalten, sind Hosts. Systeme, die diese Nachrichten über das Netz leiten, sind Routers.



7. Das TCP/IP-Protokoll

Damit das Internet funktioniert, muss es bestimmten Regeln gehorchen. Die Aufgaben des »host's« und die des »router's«, werden durch das TCP/IP (TCP=Transmission Control Protokoll; IP=Internet Protokoll) festgelegt. Wegen der Vielfalt an Protokollen ist eine geschickte Architektur unumgänglich. TCP und IP definieren mehrere verschiedene Kommunikationsprotokolle. Das IP ist für das Verschicken der Daten verantwortlich, deshalb muss es die Topologie des Netzwerkes verstehen. Das TCP ist für die Übermittlung der Daten verantwortlich. Ein Subnetzwerk besteht aus verschieden Systemen, die direkt miteinander kommunizieren können. Mehrere Subnetzwerke werden durch das TCP/IP-Protokoll zum Internet zusammen gefasst.
Trotz der internationalen Standardisierungsbemühungen der ISO hat sich die Internet-Welt mit ihrem TCP/IP-Protokollturm für die offene Kommunikation über Rechnergrenzen hinweg durchgesetzt. Ein Grund dafür ist sicher der, dass dieser Protokollturm ein interner Bestandteil des UNIX Betriebsystems ist. Eine erste derart implementierte TCP/IP-Implementierung wurde bereits 1983 ausgeliefert. Bis heute ist TCP nicht von Protokollen der ISO-Welt verdrängt worden, obwohl auch in den USA Initiativen für einen Übergang in Richtung ISO/OSI-Protokolle vorhanden waren. Den Durchbruch haben in der letzten Zeit allerdings eindeutig die Protokolle der Internet-Welt geschafft. Auch deren Bemühungen zu einer neuen Generation von Protokollen zur Unterstützung fortgeschrittener Anwendungen sind wesentlich populärer als diejenigen der ISO-Welt. Beim TCP/IP-Protokollturm handelt es sich um mehrere Protokolle, deren Entwicklung von der DARPA (Defense Advanced Research Projects Agency) für militärische Zwecke initiiert wurde. TCP/IP wurde 1983 als das Standardprotokoll für ARPANET verabschiedet. Heute ist die Internet-Protokollfamilie ein Synonym für das weltweite Internet, welches den problemlosen Datenaustausch innerhalb und zwischen den Kontinenten garantiert. Insbesondere das World Wide Web haben dem Internet einen grossen Popularitätsgewinn gebracht. Über der Transportschicht TCP befinden sich verschiedene Anwendungen und Dienste, wie zum Beispiel: FTP (File Transfer Protocol), SMTP (Simple Mail

Transfer Protocol), TFTP, NFS, SFTP, MIME, oder TELNET (Telecommunications Network). Das TCP/IP sorgt dabei für eine verbindungsorientierte, zuverlässige Datenübertragung. Das IP ist für die Wegewahl und die zuverlässige Ende-zu-Ende-Übertragung von Daten besorgt. Die Hauptaufgabe des TCP ist der Verbindungsauf- bzw. -abbau und der zuverlässige Datentransport. Bezüglich des Verbindungsauf- und -abbaus bietet TCP einen gesicherten Dienst.

8. Kommunikationsdienste

Moderne Kommunikationsprotokolle geben ihren Benutzern viel Freiheit. Die Verteilung im Netzwerk kann aufgeteilt werden in die »Connectionless Delivery« (Verbindungslose Datenübertragung) und »Connection-Oriented Delivery« (Verbindungsorientierte Datenübertragung):

8.1 Connectionless Delivery

Die »Connectionless Delivery« ist die einfachste Variante. Bei dieser Methode werden die Nachrichten unabhängig voneinander behandelt. Zu vergleichen ist diese Methode mit dem Versenden eines Briefes. Der Briefumschlag wird, gekennzeichnet mit einer Adresse, zum Postamt gebracht. Die Post kann ohne Rückfragen den Brief zum Empfänger weiterleiten, die Adresse steht auf dem Umschlag. Genauso ist der digitalen Nachricht die Adresse des Empfängers angehängt. Wie die Post, behandelt das Netzwerk jede Nachricht während der Weiterleitung separat. Wird die Ankunft der Nachricht über das Netzwerk bestätigt, wird wieder eine Nachricht gesandt. Diese wird auch als einzelne Nachricht behandelt. Das Netzwerk weiss nicht, dass es sich um eine Antwort auf die vorhergehende Nachricht handelt.

8.2 Connection-Oriented Delivery

Die Verbindungs-Orientierte Verteilung funktioniert nach einem andern Prinzip. In diesem Fall hat das Protokoll weiteren Regeln zu befolgen, als einfach unabhängige Nachrichten zu verschicken. Diese Methode ist für die Netzwerke viel aufwendiger. Dieser Service garantiert meist auch eine Bestätigung der Ankunft der Nachricht. Der Sender der Daten nimmt vor dem Versenden der Nachrichten eine Verbindung mit dem Empfänger auf. Beide stehen auch während des ganzen Vorganges in Verbindung miteinander. Dieser Vorgang lässt sich am besten am Beispiel eines Faxes erklären. Während ein Brief keine direkte Verbindung mit dem Empfänger aufnehmen muss, muss der Sender eines Faxes erst die Nummer des Faxempfängers wählen und warten, bis die Verbindung erstellt ist. Dann müssen sich die beiden Geräte für eine Methode entscheiden, mit welcher das Dokument übermittelt werden soll. Damit wird die Verbindung geschaltet. Erst jetzt kann der Sender mit der eigentlichen Datenübermittlung beginnen. Sogar während des Sendevorgangs kann es sein, dass es Rückfragen des Faxempfängers gibt. Wenn Fehler auftreten, kommt es vor, dass der Sender gewisse Daten noch einmal übermittelt. Erst wenn die gesamte Übermittlung beendet ist, beendet der Sender die Verbindung. Eine Verbindung im Verbindungs-Orentiertem Fall funktioniert praktisch gleich. Die Protokolle verschicken zuerst Verbindungsaufbaunachrichten, bevor mit der eigentlichen Übermittlung der Daten begonnen wird. Auch während der Übermittlung der Daten stehen die Protokolle in Verbindung miteinander und verschicken Kontrollnachrichten. In diesen Nachrichten kann quittiert werden, ob ein Datenpaket richtig angekommen ist. Wenn nicht, wird ein

Teil noch einmal übermittelt. Erst ganz am Ende wird die Verbindung aufgehoben.
Auch wenn diese beiden Übermittlungsmethoden verschieden sind, können sie kombiniert werden. So kann eine Schicht in einem Protokollturm die eine Methode nutzen, die nächste, die andere. Das IP ist zum Beispiel verbindungslos, das TCP verbindungsorientiert.

9. Zusammenfassung

Was ist das Internet nun eigentlich genau? Es wird oft als das »Netz der Netze« bezeichnet. Dies kann man wörtlich nehmen, denn das Internet ist ein Netz, welches verschiedene Netze miteinander verbindet. Es gibt keine Institution namens Internet und niemanden, der für »das Internet« verantwortlich ist, es sind immer nur Einrichtungen, die für Teilbereiche technisch zuständig sind. Das World Wide Web basiert als Anwendung auf dem Internet. Die TCP/IP Spezifikationen definieren dabei die entscheidenden Kommunikationsprotokolle.
Das Internet wurde vom amerikanischen Verteidigungsministerium als Forschungsprojekt entwickelt. Heutzutage hat es sich zu einem weltumspannenden Netzwerk ausgedehnt und verbindet Regierungen, Forschungsinstitute, Hochschulen, Firmen und Privathaushalte mit einander. Die Organisation des Internets basiert auf einer Hierarchie. Die Provider verbinden einzelne Nutzer sowie Firmennetzwerke. Am äussersten Ende des Internets befinden sich die Hosts.
Man erhoffte sich durch das Internet auch ein Schliessen der »sozialen Schere«, doch so wie es aussieht, ist das nicht die Realität. Die wichtigen Server stehen in Mitteleuropa und den USA, und nicht in Afrika.

10. Bibliographische Angaben

S. Thomas: *IPng and the TCP/IP Protocols*; John Wiley & Sons, Inc., New York, U.S.A, 1996, Seiten 7-26.
D. Brochers, M. Benning, J. Kuri: »*Hätt' ich dich heut' erwartet...*«; c't, Heft 21, 1999, Seiten 128-133.
E. Wilde: *World Wide Web - Technische Grundlagen*; Springer Verlag, Berlin, Deutschland, 1996, Seiten 40-42.
N. Klussmann: *Lexikon der Kommunikations- und Informationstechnik*, Hüthig, Seiten 250-255.
Und aus dem Internet:
www.learnthenet.com/german/
tecfa.unige.ch/guides/www/
www2.famvid.com/i101/
www.rvs.uni-hannover.de/people/gruen/vorträge/961106-MM-Seminar-Internet/

PPS Seminar

Grundlagen der Internet-Technologie

Bericht 2

Netzwerktechnologieen für das Internet

Verfasser:

Peter Niggli

1. Einleitung
Das Internet besitzt nicht nur eine einzige Netzwerktechnologie, die durchwegs durch das ganze Netz beibehalten wird. Es ist aus verschiedenen Arten von Netzwerken zusammengesetzt. Die Anforderungen an das Netzwerk beinhalten lediglich, dass Pakete einer bestimmten Grösse transportiert werden können. Trotz der geringen Anforderungen eignet sich nicht jedes Netzwerk gleich gut für das Internet.

Die populärsten Netztechnologien, die im Internet Verwendung finden, werden in zwei verschiedene Gruppen aufgeteilt:
1. Lokale Netze
2. Weitverkehrsnetze

2. Die Lokalen Netze
Die Lokalen Netze werden vor allem in Gebäuden, auf Firmengeländen oder lediglich in einzelnen Räumen verwendet, um eine Anzahl von Computern miteinander zu verbinden. Heutige Datenraten liegen im Bereich von Mbit/s bis zu 1 Gigabit/s. Viele dieser Netztechniken, die für jedes Medium individuell den physikalischen Eigenschaften inklusive der Zugriffsverfahren (Medium Acces Control, MAC) angepasst sind, wurden innerhalb des Institute of Electrical and Electronics Engineers (IEEE) standardisiert.

Zugriffsverfahren
Die meisten Techniken beruhen oft auf einem Medium, dass von verschiedenen Stationen gemeinsam benutzt wird. Dabei gibt es Bus-, und Ringstrukturen.

Eine Technik die auf dem Bussystem beruht ist das **Ethernet**. Die verschiedenen Rechner sind untereinander über ein gemeinsames Koaxialkabel verbunden. Wenn eine Station etwas senden möchte, muss sie zuerst das Übertragungsmedium abhören und nur dann senden, wenn keine andere Station gerade den Bus belegt. Während des Sendens muss jedoch der Bus weiterhin abgehört werden, um evtl. zu erkennen, dass eine andere Station ebenfalls zu senden begonnen hat. In diesem Fall brechen beide Sender die Übertragung sofort ab und versuchen es nach einer zufällig bestimmten Zeit erneut. Diese Zeitspanne ist zufällig, da sonst die selben Rechner nach der selben Zeit wieder kollidieren würden. Ethernet operierte zunächst mit Übertragungsraten von 10 Mbit/s. Weiterentwicklungen, wie Fast Ethernet mit 100 Mbit/s und schliesslich Gigabit-Ethernet mit 1 Gbit/s, kamen anschliessend auf den Markt.

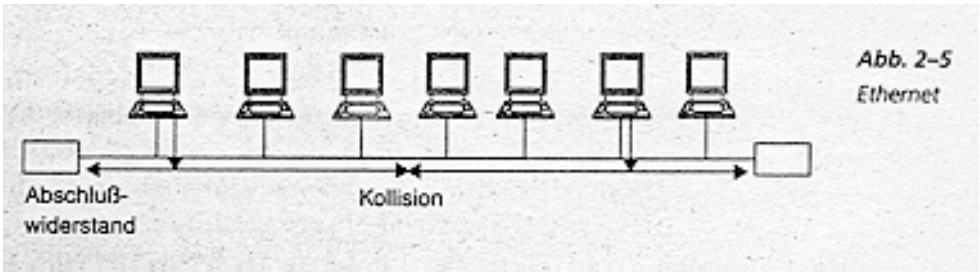
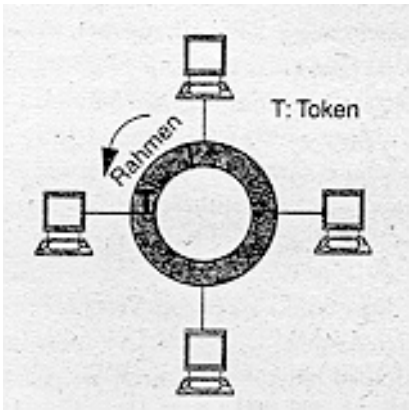


Abb. 2–5 Ethernet

Bei einem Ethernet mit 10 Mbit/s ist daher die maximal erlaubte Netzausdehnung ca. 2,5 km bei einer minimalen Paketlänge von 64 Bytes. Aus dem Grund, dass der Bus auf Kollisionen abgehört werden muss, hängt die Geschwindigkeit direkt mit der Länge des Busses und der minimalen Paketlänge zusammen, da eine Kollision sonst

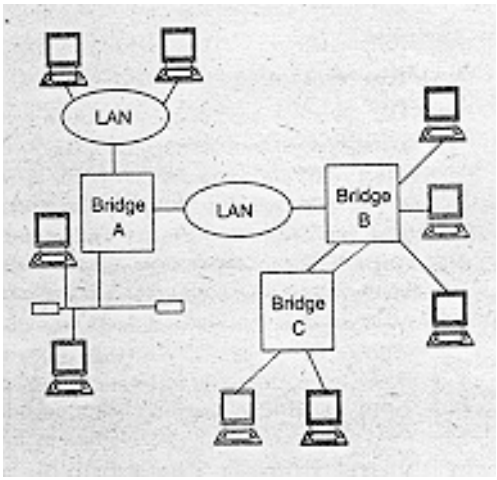
nicht mehr erkannt werden kann. Ein schnelleres Ethernet bedeutet somit eine grössere minimale Paketlänge, da es unsinnig währe die Buslängen auf wenige Meter zu verkürzen.
Eine andere Technik ist die **Token-Ring-Technik** und deren Weiterentwicklung für Glasfasernetze (Fiber Distributed Data Interface FDDI). Sie sind Vertreter von Ringtopologien. Bei diesem Ringsystem wird das Senderecht mit Hilfe eines Token geregelt. Der Token hat 2 Zustände: Belegt oder Frei. Beim Token-Ring darf sich immer nur ein Paket auf dem Ring befinden. Somit darf eine Station nur senden, wenn der Token auf frei ist. Es wird dann zuerst der Token auf belegt gesetzt und das Paket gesendet. Die Empfangsstation kopiert das Paket, und der Sender nimmt anschliessend das Paket vom Ring und setzt den Token wieder auf frei.



Beim FDDI wird der Token von ein Paket gesetzt, und es dürfen sich gleichzeitig mehrere Pakete auf dem Ring befinden.

LAN-Kopplung mit Bridges und Switches

Da nun so viele verschiedene Netzwerktechnologien vorhanden sind und alle anders funktionieren, benötigt man sogenannte Bridges damit zwei Rechner, die über zwei oder mehrere verschiedene LAN Netzwerke mit einander verbunden sind, miteinander kommunizieren können. Das ganze wirkt dann so, als wären sie am selben lokalen Netz angeschlossen. An eine Bridge können gleichzeitig mehrere Netzwerke angeschlossen werden. Bridges mit sehr vielen Anschlüssen werden auch Switches genannt. Die Aufgabe der Bridges besteht also darin, von Endsystemen gesendete Pakete an die Segmente mit den Zielstationen weiterzuleiten.



Bei den Bridges gibt es zwei verschiedene Systeme das **Source Route Bridging** und das **Transparent Bridging**. Bei letzterem werden alle Anschlüsse gesperrt, für die keine Verwendung ist, um eine Nachricht zu ihrem Empfänger zu schicken. Damit verhindert man, dass eine Nachricht über zwei verschiedene Kanäle doppelt ankommt. Beim erstmaligen Übermitteln einer Nachricht von A nach B ist der Weg und die Adresse für alle Bridges unbekannt. Aus diesem Grund leiten alle Bridges das Paket entlang eines von einer Bridge (Wurzel) aus aufgebauten Spannbaums weiter. Die einzelnen Bridges lernen von wo das Paket gekommen ist und wissen somit, an welchem Anschluss der Rechner A ist. Wenn von B aus nun eine Nachricht zurück kommt, wird A sofort über die gespeicherten Anschlüsse gefunden. Diejenigen Bridges, die nicht beteiligt waren, wissen nun auch, dass A auf der selben Anschlussseite liegt wie die Nachricht B und leiten somit das Paket nicht weiter. Für weitere Nachrichtenübertragung wird nun nur noch der speziell in den Bridges gespeicherte Weg benutzt. Nach einer gewissen Zeit wird die Information wieder verworfen und die Zuordnung wird neu gelernt, so dass auch Netzänderungen sich auswirken.

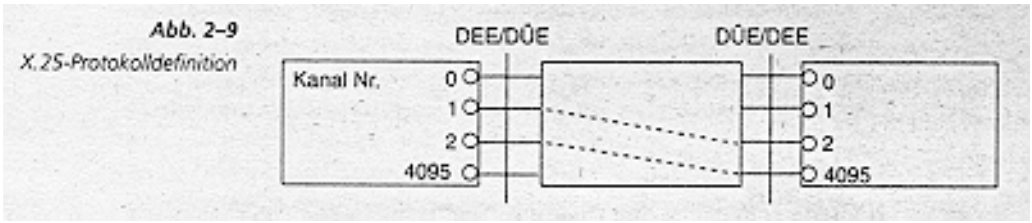
Das **Source Route Bridging** vermeidet das Verwalten von Adresstabellen. Bei diesem Verfahren gibt der Sender direkt an, welcher Weg zu benutzen ist. Den Weg muss der Sender jedoch zuerst lernen. Dazu schickt er zuerst ein Discovery Paket zum Empfänger, der dann eine Antwort zurückschickt. Dabei wird das Paket von jeder Bridge über alle anderen Kanäle weiter geleitet und gleichzeitig wird darin vermerkt, bei welcher Bridge das Paket vorbeigekommen ist. Wenn es zweimal bei der selben Bridge ankommt, wird es nicht mehr weitergeleitet. Am Ende kennt dann der Sender durch die Antwort alle Bridges, die zwischen ihm und dem Empfänger liegen und kann damit über einen all dieser Wege, die er nun kennt das Paket senden und das Datenpaket wird von den Bridges nur über den darin beschriebenen Weg zum Ziel geleitet.

3. Die Weitverkehrsnetze

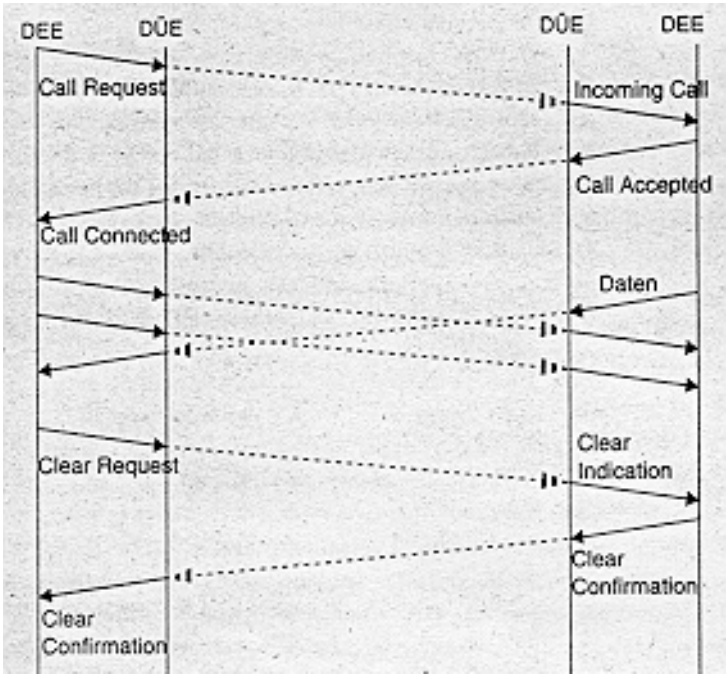
Weitverkehrsnetze (WANs) sind meist verbindungsorientiert, d.h. vor dem Datenaustausch müssen Verbindungen erstellt werden, die nach dem Austausch wieder abgebaut werden.

X.25

Der Übergang zwischen den Endgeräten (Datenendeinrichtung DEE) und dem Gerät zur Datenübertragung über das Weitverkehrsnetz (Datenübertragungseinrichtung DÜE) bildet eine Schnittstelle, die durch **X.25** definiert ist. Es wird dadurch über mehrere virtuelle Kanäle kommuniziert. Die Kanalkennungen sind auf beiden Seiten verschieden und werden in den dazwischenliegenden X.25-Vermittlungsstellen umgesetzt.



In der Datenübertragungsphase werden über die verschiedenen Schichten neben der Übertragung auch noch Fluss- und Fehlerkontrollfunktionen ausgeführt, sowie Funktionen zum Verbindungsauf- und abbau und den verschiedenen Verfahren zur Unterstützung des Datentransfers kombiniert. Die Flusskontrolle basiert dabei auf einem bestimmten Mechanismus. Die Fehlerbehebung erfolgt durch Übertragungswiederholungen.



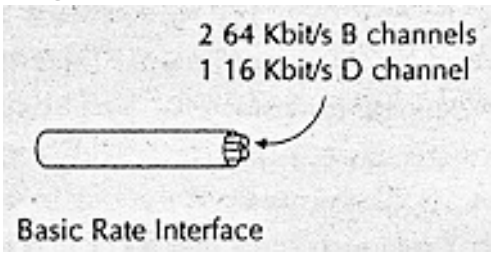
Nach der Übertragung wird der virtuelle Kanal wieder abgebaut.

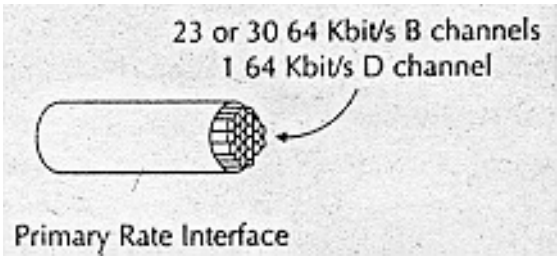
In der nebenstehenden Abbildung ist der Verbindungsaufbau visualisiert: Mit dem Senden der Nachrichten Call Request auf Initiator-

seite und Incomming Call auf der Seite des Empfängers wird der Aufbau eingeleitet. Mit Call Accepted und Call Connected wird die Verbindung angenommen bzw. der Aufbau der Verbindung angezeigt. Daten können nun übermittelt werden. Die Verbindungsabbau wird durch Clear Request eingeleitet, was beim Empfänger mit Clear Indication angezeigt wird. Schliesslich wird die Verbindung mit der Nachricht Clear Confirmation beendet.

Da X.25 durch die Existenz von Fluss- und Fehlerkontrollfunktionen eine relativ schwergewichtige Protokollarchitektur ist, existiert im Gegensatz dazu das sogenannte **Frame Relay**. Es arbeitet lediglich mit Signalisierungsfunktionen um damit Verbindungen auf- und abzubauen. Die Funktionen sind auf einem Protokoll (LAP-D), um den zuverlässigen Austausch von Signalisierungsinformationen zwischen DEE und DÜE zu gewährleisten. Für den Datentransfer wird neben diesen Signalisierungsprotokollen ein separater Protokollstack für die Benutzerebene geführt.
Die Signalisierung wird über eine reservierte Verbindungskennung durchgeführt. Die Kontrollebene kann ganz entfallen, wenn die Verbindungen nicht dynamisch über Signalisierung, sondern über Managementfunktionen fest konfiguriert werden.

ISDN
Das Integrated Services Digital Network (ISDN) basiert auf mehreren Schichten. Es handelt sich um eine in Kontroll- und Benutzerebene unterteilte Protokollarchitektur. Im Gegensatz zu X.25 ist ISDN jedoch keine Paketvermittlungstechnik, sondern leitungsvermittelnd. D.h. es wird bei ISDN zwischen den Kommunikationsteilnehmern eine Verbindung (B-Kanal) mit fester Bitrate (64 kbit/s) etabliert. Diese Verbindungen können auch ausgebaut werden. So kann ein Basic Rate Interface (BRI)eine Bitrate von 128 kbit/s oder ein Primary Rate Interface (PRI) von 1.472 Mbit/s oder 1.92 Mbit/s erreichen.





Die Signalisierung erfolgt über einen separaten Kanal (D-Kanal). Der Aufbau wird über eine SETUP-Nachricht eingeleitet, und die Verbindung wird mit einer CONNECT-Nachricht angenommen. Nach dem Datentransfer wird die Verbindung mit einer RELEASE-Nachricht aufgelöst.

Das ISDN ist die neue Technologie, die in den vergangenen Jahren vor allem durch die Verwendung bei der Telekommunikation bekannt wurde. Es ersetzte die analogen Telephonverbindungen durch leistungsfähigere und flexible digital Verbindungen. In vielen Ländern ist das BRI in sogenanntes single twisted pair cable also nur ein Kabel, dass zum Benutzer gelangt. Es wird auch das U Interface genannt. Dieses wird von einem Network Termination 1 (NT1) (ähnlich einem Modem) entgegengenommen, welches dann die Daten auf das zwei kabelige T Interface Übersetzt, um schneller arbeiten zu können. Von den zwei Kabeln ist dann eines für die eine Daten-

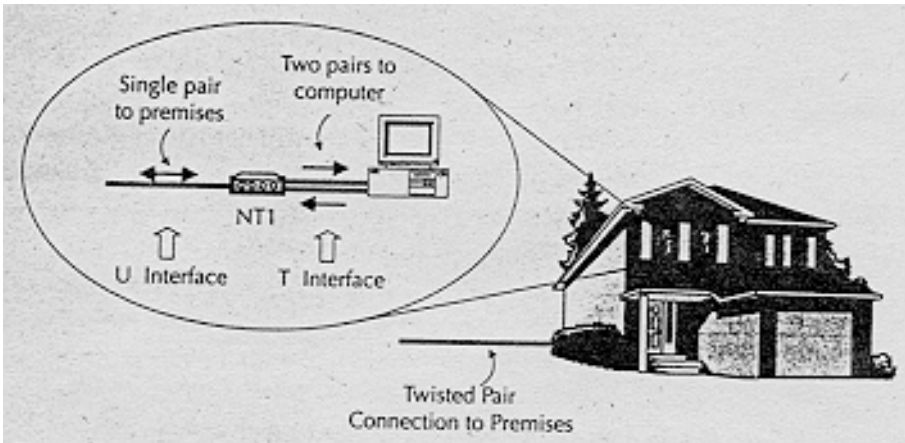


Figure 3.18 Interfacing to ISDN.

flussrichtung und das andere für die andere Datenflussrichtung vorgesehen. Im U Interface sind dann jeweils beide Datenflussrichtungen mit einander kombiniert.

Asynchronous Transfer Mode (ATM)

ATM war eigentlich als Basis für Breitband-ISDN, also für ein Weitverkehrsnetz vorgesehen und etablierte sich inzwischen im lokalen Bereich. Seine Protokollarchitektur wird in drei Ebenen Aufgeteilt: Benutzer-, Kontroll- und Managementebe.

Die Benutzerebene behandelt den Transfer von Benutzerdaten, die Kontrollebene ist für die Handhabung der Signalisierungsinformation zuständig, und die Managementebene führt Managementfunktionen aus und koordiniert die drei Ebenen.

Da in ATM immer dieselben Übertragungspfade für Zellen einer Verbindung verwendet werden, können keine Reihenfolgefehler auftreten, ausser es liegt ein Zellenverlust und eine anschliessende Datenübertragungswiederholung vor. Andere Fehler sind jedoch nicht auszuschliessen. Die Fehlererkennung wird in den höheren Schichten der Endsysteme durchgeführt.

Zur Datenübertragung wird jeweils nur soviel Bandbreite reserviert wie gewünscht wird der Rest bleibt für andere Benutzer verfügbar.

ATM verwendet zur Datenübertragung ausschliesslich Zellen fester Länge von 53Bytes. 5Bytes für die Kontrollinformation und 48 für die Nutzlast.

4. Zusammenfassung

Wir haben nun auf den vorigen Seiten Einiges über diverse Netzwerktechnologieen gelesen und wollen dies nun noch einmal zusammenfassen.

Es gibt eine grundlegende Unterscheidung der Netzwerke zwischen **Lokalen** und **Weitverkehrsnetzen.** Bei den Lokalen Netzen kennen wir nun 2 Hauptsysteme: Das Bussystem (**Ethernet**) und das Ringsystem (**Token-Ring-Technik**). Im weiteren haben wir erfahren, sie die einzelnen Verbindungen von einem Rechner zum andern über das sehr verworrene Netz zustande kommen können, und wie diese Verbindungen mit Hilfe von **Bridges** auf kürzestem Wege erstellt werden.

Bei den Weitverkehrsnetzen haben wir **X.25**, **Frame Relay** und **ISDN**, die diversen Schnittstellen zwischen DEE und DÜE kennengelernt, die auf verschiedene Arten diesen Übergang definieren. Die letzte Schnittstelle, die wir kennengelernt haben ist dann noch ATM welche sich jedoch hauptsächlich in lokalen Netzen etablierte, sie ist jedoch sowohl für das lokale, als auch für das Weitverkehrsnetz geeignet.

Bibliographische Angaben

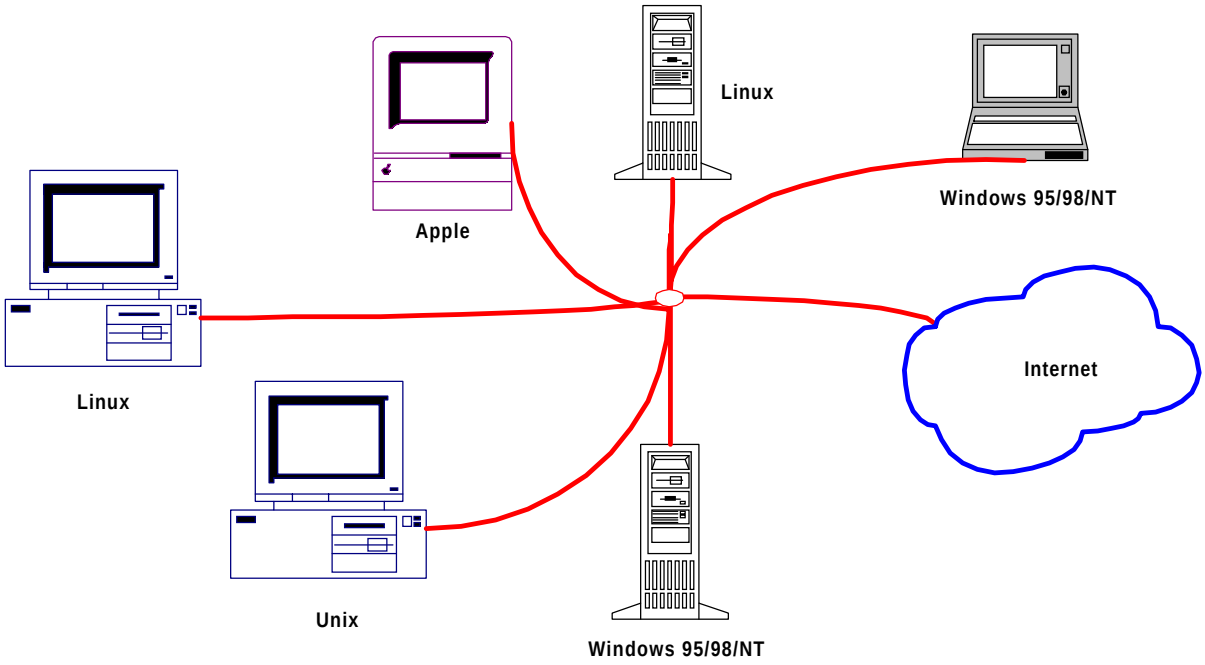
S. Thomas: Ipng and the TCP/IP Protocols; John Wiley & Sons, Inc., New York, U.S.A., Seiten 43-76

T. Braun: Ipng – Neue Internet-Dienste und virtuelle Netze; dpunkt Verlag, Heidelberg, Deutschland, 1999, Seiten 5-26

PPS Seminar

Grundlagen des Internet

Vortrag 3



IP, Adressierung und Routing im Internet

Sarah Rüdiger

1. Einleitung

Schon die Erfinder des Internet-Vorläufers ARPANet standen vor dem Problem, wie vier Grossrechner untereinander kommunizieren sollten, welche durch ein zirkuläres System miteinander verbunden waren. Um z.B. eine Datei von Rechner A zu Rechner C zu transportieren, musste man dafür sorgen, dass der dazwischenliegende Rechner B die Daten richtig weiterleitet. Ausserdem stellte sich die Frage, wer darf wie entscheiden, welchen Weg eine Datei zu ihrem Ziel nehmen muss oder soll? Und was passiert, wenn die Übertragung einfach unterbrochen wird? Diese und andere Fragen führten dazu, dass die Erfinder von ARPANet schliesslich TCP/IP entwarfen und einführten.

Generell gilt also, damit Daten über ein Netz transportiert werden können, dass ein einheitliches Uebertragungsprotokoll notwendig ist , auf dem alle Anwendungen eines Netzwerks aufbauen können. Im Internet heissen diese Protokolle TCP/IP. In den folgenden Abschnitten soll ein Ueberblick über den Aufbau dieser Protokolle und der Datenübertragung im Internet gegeben werden.

2. Internet Protocol (IP)

Das Internet-Protokoll ist eine grundlegende Voraussetzung zur Datenübertragung im Internet.
Die wesentlichsten Aufgaben von IP sind:

- Wegewahl
- Lebenszeitkontrolle (time-to-live)
- Segmentieren und Reassemblieren
- Fehlererkennung
- Adressierung

IP ist ein verbindungsloses Protokoll d.h. dass Pakete unabhängig voneinander weitergeleitet werden. Dies ermöglicht eine schnelle und flexible Datenübertragung, da Engpässe in Teilnetzen, z.B. durch Ausfälle oder Ueberbelastung umgangen, werden können. Jedoch garantiert IP nicht, dass die Pakete in der ursprünglichen Reihenfolge beim Empfänger ankommen.

2.1 Wegewahl

Jedes System (End- oder Zwischensystem) analysiert die Zieladresse eines Datenpakets. Hierfür wird sie mit der eigenen Adresse und Adressen direkt angeschlossenener Netzwerke verglichen. Sind keine Adressen identisch, so wird das Paket an ein nachfolgendendes Zwischensystem weitergeleitet.
Handelt es sich um eine nicht bekannte Adresse, wird eine Fehlermeldung über das Internet Control Message Protokoll (ICMP) generiert und an den Sender des Pakets verschickt.

2.2 Lebenszeitkontrolle:

Fehlfunktionen im Netzwerk können dazu führen, dass Datenpakete nicht an die Zieladresse gelangen, sondern im Netzwerk zirkulieren. Diese Daten beanspruchen unnötig Netzwerkkapazitäten und müssen folglich irgendwann gelöscht werden. Hierfür dient die Lebenszeitkontrolle (time-to-live), sie stellt einen Wert dar, der die Lebenszeit eines Pakets im Netz limitiert. Anfänglich sollte die Masseinheit die Anzahl verbrachter Sekunden im Netz darstellen. Schliesslich hat sich jedoch der sogenannte Hop-Count durchgesetzt. Er legt fest, wie viele Knoten bzw. Router ein Paket im Netz durchlaufen darf. In jedem Knoten wird der Zähler um eins dekrementiert. Ist dieser Wert vor dem Erreichen der Zieladresse Null, wird das Paket gelöscht und an die sendende Station wird über ICMP eine Fehlermeldung geschickt.

2.3 Segmentierung und Reassemblierung:

Verschiedene Faktoren wie Hard- oder Softwarebeschränkungen, Beschränkungen aufgrund eines verwendeten Standards oder als Massnahme zur Reduzierung der Fehlerquote bei der Datenübertragung haben dazu geführt, dass eine maximale Länge für IP-Datenpakete existiert. Aus diesem Grund verfügt IP über eine Funktion, welche Dateneinheiten beim Senden segmentiert und im Endsystem wieder reassembliert. In Zwischensystemen werden die Daten nicht reassembliert, da dies unnötige Verzögerungen der Datenübertragung zur Folge hätte und die unabhängigen Wege der Pakete ein sinnvolles Reassemblieren verunmöglichen.

2.4 Fehlererkennung

Die Fehlererkennung beruht auf einer Prüfsumme über dem Kopf des Pakets. Aus diesem Grund können Fehler auch nur im Kopf eines Pakets festgestellt werden. Damit sind die Nutzdaten eines IP-Pakets, somit Daten der Netzwerkschicht, nicht gesichert. Ausserdem muss die Prüfsumme in jedem Knoten neu berechnet werden, da durch den neuen Hop-Count sich auch der Kopf des Pakets und somit die Prüfsumme verändert.

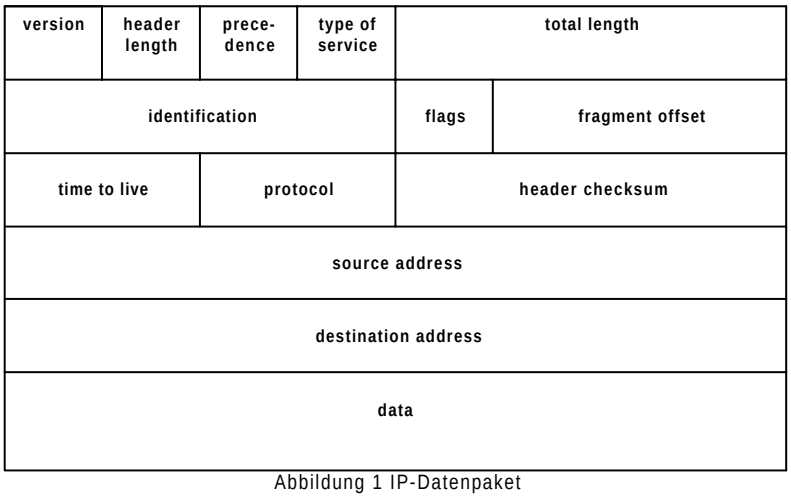


Abbildung 1 IP-Datenpaket

	Bedeutung
version	Versionsnummer des Protokolls
header length	Länge des IP-Kopfs in 32 Bit
precedence	Prioritätsinformationen sowie Routing-Protokoll-Daten
type of Service	Angabe des Diensttyps z.B. Art des Weiterleitens der Daten
total length	Gesamtlänge in Byte inklusive Header und Daten
identification	Identifikationswert von Paketen
flags	Bestimmte Werte auf 0 oder 1 z.B. ob Fragmentierung zugelassen
fragment Offset	Position im reassemblierten Teil des Pakets
time-to-live	Lebenszeit bzw. maximaler Hop-Count des Pakets
protocol	Kennung des 4 Schicht-Protokolls z.B. 7 für TCP
header checksum	Prüfsumme über den Kopf des Pakets
source address	IP-Adresse der Herkunftssystems
destination address	IP-Adresse des Zielsystems
options	Enthält Optionen wie Weginformationen etc.
options Padding	Ergänzt die durch die Options bestimmten Daten auf ein Vielfaches von 32
data	Enthält die eigentlichen Nutzdaten

Abbildung 2 Bedeutung der Felder eines IP-Datenpakets

3. IP-Adressierung

Jeder Rechner, der über das Internet kommuniziert, besitzt eine IP-Adresse. Somit ist jede Ressource im Internet mit einer eindeutigen Nummer erreichbar. In dem zur Zeit verwendeten IP-Protokoll IPv4 besteht jede IP-Adresse aus 32 Bit. Diese 32 Bit sind wiederum in je 8 Bit geteilt und diese können jeweils dezimal notiert werden. Durch diese Aufteilung können insgesamt (2^8)^4 = 4‘294‘967‘296 Adressen verwaltet werden.

Eine IP-Adresse besteht aus 2 Teilen:

- Netzerkennung: identifiziert in welchem Teilnetz ein Rechner angeschlossen ist
- Rechnererkennung bzw. Rechneradresse

Derzeit werden die IP-Adressen in fünf Klassen eingeteilt:

Klasse A
Die ersten 8 Bits bezeichnen das Netz und die restlichen 24 Bits den jeweiligen Rechner. A-Adressen sind für sehr grosse Netze geeignet. Hauptsächlich werden A-Adressen von grossen Firmen insbesondere amerikanischen Organisationen verwendet. Die Netzerkennung wird durch das erste Byte dargestellt wobei das erste Bit stets auf Null gesetzt ist. Die restlichen sieben Bits ermöglichen die Adressierung von 2^7=128 Netzen. Zwei Kombinationen werden jedoch nie zur Adressierung verwendet , nämlich alle Bit auf 0 bzw. 1. Diese Adressen sind für spezielle Zwecke reserviert. Ausserdem ist die Adresse 127 bzw. Loopback-Adresse ein Sonderfall. Sie dient für rechnerinterene Testzwecke. Schliesslich existieren also 126 A-Klasse Netze. Die restlichen drei Byte stehen zur Bezeichnung des Hosts zur Verfügung. Das sind genau 2^24 –2 =16‘777‘214 Kombinationen. Insgesamt können folglich über 16 Millionen Rechner an ein A-Klasse Netz angeschlossen werden.

	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
	0	Netzwerk							lokale Adresse																							
bin.	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
hex.	5				9				8				1				0				1				8				2			
dez.	089								129								001								130							

Adressformat der Klasse-A-Adressen

Klasse B
Die ersten beiden Bytes bezeichnen das Netzwerk, wobei die ersten zwei Bits immer 1 und 0 sind.
Es können 16384 Netze mit jeweils 65534 Hosts bezeichnet werden.

	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
	1	0	Netzwerk														lokale Adresse															
bin.	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
hex.	8				0				0				5				0				1				8				2			
dez.	128								005								001								130							

Adressformat der Klasse-B-Adressen

Klasse C
Die ersten drei Bytes bezeichnen das Netzwerk wobei davon die ersten drei Bit stets 110 sind. Dies stellt 2‘097‘152 Netze mit jeweils 254 Hosts dar.

	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
	1	1	0	Netzwerk																					lokale Adresse							
bin.	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0
hex.	C				0				0				0				0				6				0				2			
dez.	192								005								006								002							

Adressformat der Klasse-C-Adressen

Klasse D
Klasse D-Netze werden für Multicasting. Sie besitzen ein eigenes Format in welchem die ersten drei Bits des ersten Bytes stets auf 1 gesetzt sind.

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	1	1																													

Adressformat der Klasse-D-Adressen

Klasse E
Diese Adressen sind für Testzwecke reserviert und die ersten 4 Bits sind alle auf 1 gesetzt.

4. Transmission Control Protocol (TCP)

Das Transmission Control Protokoll steht in der Schichten-Architektur oberhalb des Internet-Protokolls. Es benutzt den Dienst von IP, welcher Daten versendet, und verfügt zusätzlich über Funktionen, welche überprüfen, ob ein Datenpaket tatsächlich beim Empfänger angekommen ist. Fehlt ein Datenpaket, so wird die Wiederholung der Übertragung angefordert. Folglich benötigt TCP eine Verbindung zur Gegenstation.

TCP segmentiert zunächst den ankommenden Datenstrom und numeriert diese Segmente. Der empfangende Knoten im System kann nun anhand dieser Segmentnummer den Empfang eines Paketes bestätigen. Es wird jedoch nicht jedes Paket einzeln bestätigt, sondern alle Pakete innerhalb eines sogenannten Fensters. Erst wenn der Sender diese Bestätigung erhalten hat, kann er mit dem Versenden weiterer Daten fortfahren. Der Empfänger bestätigt mit einem Paket immer auch den Empfang der vorhergegangenen Pakete. Tritt ein Fehler auf, müssen alle Pakete seit der letzten Empfangsbestätigung erneut übertragen werden.

Broadcast

Broadcast-Mechanismen werden von vielen Applikationen verwendet um schnell Daten zu versenden oder schnell Daten vom Kommunikationspartnern zu erhalten. Broadcasts sind Rundsendungen im Netz, welche an alle Rechner gerichtet sind. Alle angesprochenen Rechner untersuchen den Broadcast und reagieren entsprechend. Broadcasts können grundsätzlich auf zwei Arten verschickt werden:

- als Abfrage aller Knoten im Netz nacheinander und
- Informationsverteilung an alle am Netz angeschlossenen Rechner.

Ein Rechner benötigt Informationen zu einem bestimmten Rechner. Bei der ersten Methode sendet er nacheinander jedem Rechner eine Anfrage, bis ihn eine positive Antwort erreicht. Dies bringt jedoch Probleme mit sich. Das Vielfache Aussenden derselben Informationen kann das Netz unnötig belasten. Folglich wird dieses Verfahren allgemein nicht verwendet.

Wird die zweite Methode verwendet, so sendet der Rechner an alle Anderen die gleichen Informationen. Folglich erreicht die Information, bei geringer Netzbelastung, stets den gewünschten Rechner.

Subnet- Mask

Subnetze werden meist eingerichtet um topologische und organisatorische Probleme im Netz besser kontrollieren zu können. Ein Subnetz wird unter Zuhilfenahme einer Bitmaske (Subnet Mask) festgelegt. Die Subnet Mask wird mit der IP-Adresse durch eine logische UND-Funktion verknüpft. Ein Subnet ist nur den am lokalen Netz angeschlossenen Rechnern bekannt, auf allen anderen Rechner erscheint eine solche Adresse als gewöhnliche IP-Adresse. In der Subnet Mask sind alles Bits auf 1 gesetzt welche den Netzwerkanteil festlegen, z.B. für Klasse-B-Adressen ist die Default-Subnetzmaske: 255.255.0.0. Jedoch lassen sich auch Zwischenklassen definieren, so unterteilt die Subnetzmaske 255.255.255.192 ein Klasse-C-Netz in 4 Subnetze.

5. Routing

Durch die Funktion des Routings können verschiedene Datennetze zu einem gemeinsamen Gesamtnetz verbunden werden. Die Grundfunktion eines Routers besteht in der Wegewahl bzw. Wegefindung innerhalb eines Netzwerks. Zwischen verschiedenen am Netzwerk angeschlossen Routern werden zyklisch Datenpakete zur Wegefindung versendet. Durch dieses Verfahren lernt jeder Router die möglichen Pfade zwischen Netzen kennen. Kennt ein Router mindestens einen Pfad (Default) bzw. alle Pfade wird der Weg zu einem Datennetz in die sogenannte Routing-Tabelle eingetragen.

Die Initialisierung dieser Einträge kann auf zwei Arten erfolgen:

- statisch, durch manuelle Konfiguration der Tabellen
- dynamisch durch den Austausch der Routing-Informationen zwischen den Routern über Routing-Protokolle

In grossen Netzwerken, welche zusätzlich aus vielen Subnetzen bestehen, kann es ausserdem leicht vorkommen, dass diese Tabellen sehr gross werden, daher werden einzelne Netze in sogenannte Routing-Domänen unterteilt.

5.1 Routing- Protokolle

Die Routingprotokolle basieren auf zwei grundlegenden 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**
  Jeder Router besitzt Informationen über jeden Link der Domäne und berechnet die komplette Netztopologie der Domäne.

Routing Information Protocol (RIP)
RIP ist ein einfaches und weit verbreitetes Distanz-Vektor-Protokoll. Jeder Router sendet in zyklischen Zeitabständen seine aktuellen Routing-Informationen über die angeschlossenen Links und teilt den benachbarten Router mit wie gross seine Distanz zu anderen Netzen ist.

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 bis schliesslich die Topologiebeschreibung der ganzen Domäne bekannt ist.

5.2 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 Aufgabe

administrative 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.

6. Dynamic Host Configuration (DHCP)

TCP/IP hat den Nachteil, dass die Wartung und Einrichtung der Netzwerkparameter speziell bei grossen Netzwerken sehr aufwendig sind. DHCP ermöglicht es, alle TCP/IP-Konfigurations-Parameter zentral zu verwalten und zu warten. DHCP besteht prinzipiell aus zwei Komponenten. Zum einen ist dies ein Protokoll, dass die Uebertragung der Konfigurationsparameter von einem DHCP-Server zu den Clients steuert, und zum anderen einer Funktion für die Zuweisung von Netzwerkadressen an die Clients.
DHCP kann auf drei Arten IP-Adressen zuweisen:

- automatisch
- dynamisch
- manuell

Die automatische IP-Adresszuweisung ordnet jedem Rechner eine beliebige aber feste IP-Adresse zu.
Bei diesem Verfahren wird dem Client bei der ersten Anmeldung am Netz eine freie Adresse mitgeteilt. Diese Adresse wird ihm fortan bei jeder Anmeldung wieder zugewiesen und kann nicht an andere Anwender weitergegeben werden. Bei der dynamischen Adresszuweisung wird einem Rechner eine IP-Adresse nur für einen bestimmten Zeitraum zugewiesen. Der Vorteil dieser Methode besteht darin, dass eine IP-Adresse immer wieder an neue Clients vergeben werden kann, sofern sie nicht momentan in Verwendung sind. Bei der manuellen Zuweisung kann der Administrator Adressen explizit zuweisen. In diesem Fall dient DHCP lediglich als Transportmittel verwendet.

7. Adressauflösung

Kommuniziert ein Benutzer über das Internet, so ruft z.B. Anwendungen, Rechnernamen meist in Textform auf, da dies oftmals einfacher ist als stets gewünschte IP-Adresse zu wissen. Lautet die Eingabe des Anwenders z.B. www.ee.ethz.ch, so bezeichnet dies einen Rechner, auf dem die www-Seiten gespeichert sind. Jedoch muss erst die IP-Adresse dieses Rechners bestimmt werden, dies ermittelt der Dienst namens Domain Name System (DNS). Nun muss das IP-Protokoll in Endsystemen und Routern die gewünschten Datenpakete über Router an den Zielrechner senden. Dies geschieht durch das Address Resolution Protocol (ARP), in dem jeder Router zwischen Quelle und Ziel die Netzadresse des nächstens Knotens ermittelt.

Domain Name System (DNS)

Die Struktur des DNS ist ein Baum, dieser besteht aus einem Root, Top-Level Domains wie com, org, ch etc. und Domänen. Root sowie die Top-Level Domains werden von Organisationen verwaltet, welche grosse DNS-Server besitzen und alle Anfragen auf die Top-Level Domains beantworten. Die tieferen Eben sind die Domains, dies können z.B. Firmen, Privatleute sein. Schliesslich sind auf der untersten Ebene die eigentlichen Rechnernamen registriert.

8. Schlusswort

Das TCP/IP stellen an sich einfache Systeme dar um die Kommunikation über das Internet zu ermöglichen. Das Wachstum des Internets bringt jedoch Probleme mit sich, da die ursprüngliche Dimensionierung des Adressraums in Zukunft nicht mehr ausreichen wird. Aus diesem Grund gilt es, in absehbarer Zeit eine erweiterte bzw. neue Version des IP-Protokoll (Ipv6) einzuführen.

9. Quellenangaben

- T.Braun: IPng – Neue Internet-Dienste und virtuelle Netze; dpunkt Verlag, Heidelberg, Deutschland, 1999, Seiten 27-44.
- S.Thomas: IPng 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.

Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Departement Elektrotechnik

Institut für Technische Informatik
und Kommunikationsnetze

IPng - Die nächste Generation

PPS- Seminar: Grundlagen der Internettechnologie

Reto Zürcher

28. November 2000

1 Einleitung

Die Grundlage der Kommunikation verschiedener Internetteilnehmer wie Router, Server und auch Home PC ist ein universales Protokoll, das sogenannte Internet Protokoll (IP). Das zurzeit installierte, das IPv4, ist schon seit 1975 praktisch unverändert im Gebrauch. Somit und aus verschiedenen weiteren Gründen, die in den folgenden Abschnitten näher erläutert werden, wurde 1992 von der IETF (Internet Engeneering Task Force) entschieden eine neue Version zu entwickeln.
Diese Ausarbeitung ist der zweite Vortrag einer zweiteiligen Zusammenstellung, die das Internetprotokoll in groben Zügen erklären soll. Sie stellt kurz die Nachteile des IPv4 vor und geht dann auf die nächste Generation, das IPv6 oder IPng (Internet Protocol next generation) ein.

2 Der Weg zum IPv6

Die rasante Entwicklung der Internettechnologie, welche in naher Zukunft in bisher noch unberührte Bereiche, wie Auto, Haushalt u.v.m. vorstossen wird, lässt Experten davon ausgehen, das im Jahre 2020 10 Milliarden Menschen je ca. 100 IP-Adressen benötigen.
Ohne grosse Rechenarbeit ist sofort zu erkennen, dass sich diese Nachfrage mit einer 32 bit Kodierung nicht decken lässt. Es stellt sich aber nicht nur das Problem, dass der bisherige Adressraum erschöpft wird, sondern auch, dass mit der steigenden Anzahl Adressen eine Explosion der Routingtabellen verbunden ist. Somit ist auch in dieser Hinsicht eine neue Technologie erforderlich.

3 IPv6

3.1 Adressierung

Die Länge der Adressen wird im IPv6 auf 128 bit erweitert. Damit steigt die theoretische Anzahl an Adressierungsmöglichkeiten geradezu ins Unendliche. Neu wird es unserer Gesellschaft möglich sein, pro Quadratmeter Erdoberfläche ca. $6,65 * 10^{23}$ verschiedene Adressen zu definieren! Natürlich wird beim nüchternen Betrachten dieser Menge die Frage der Notwendigkeit einer 128 bit-Codierung aufgeworfen, auch wenn wir die Entwicklung des Internets bedenken. Dieser Abschnitt soll aber aufzeigen, dass mit dieser 16 Byte langen Adresse grosse Vorteile von Strukturierungen verbunden sind.

3.1.1 Allgemeines Format

Grundsätzlich existieren drei verschiedene Adresstypen, welche in den folgenden drei Abschnitten näher erläutert sind. Die Unterscheidung erfolgt im Format-Präfix, welches in den vordersten Bits codiert ist. Die allgemeine Notation hat folgende Struktur: Die gesamte Adresse wird in der vollständigen Schreibweise mit sieben Doppelpunkten unterteilt. Wie Tabelle 1 demonstriert, gibt das dann acht 16 bit Adresssegmente, welche übersichtlicherweise mit je vier hexadezimalen Ziffern dargestellt sind. Es ist möglich, die komprimierte Darstellung

Normal	Komprimiert	Bemerkung
1080:0:FF:0:8:800:200C:417A	-	normale IP-Adresse
FF01:0:0:0:0:0:0:43	FF01::43	IPv6
0:0:0:0:0:0:128:176:0:46	::128:176:0:46	IPv4 kompatibel
1080:0:0:0:8:0:1289:1C6F	1080::8:0:1289:1C6F	

Tabelle 1: Beispiel zu IPv6 Adressen

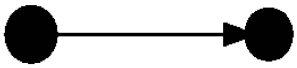


Abbildung 1: 1:1 Kommunikation mit der Unicast Adresse

zu wählen, wobei einerseits überflüssige Nullen weggelassen und andererseits mehrere zusammenhängende Nullen mittels eines doppelten Doppelpunkts abgekürzt werden. Das letztere darf aber der Eindeutigkeit halber in einer Adresse nur einmal angewendet werden.

3.1.2 Unicast-Adressen

Unicast-Adressen finden ihre Anwendung in der 1:1 Kommunikation (Abbildung 1) Jedes abgeschickte Paket hat einen eindeutigen Empfänger. Sie besteht aus drei Teilen:

- Der globale Teil setzt sich weiter aus dem Top Level Aggregator (TLA) und dem Next Level Aggregator (NLA) zusammen, welche den Ort im Internet spezifizieren und hierarchische Strukturen sind.
- Der Lokations-spezifische Teil (Site Level Aggregator, SLA) steht für die Subnetz Struktur zur Verfügung,
- und der letzte Teil stellt die Interface-ID dar, welche oft mit der MAC-Adresse übereinstimmt.

Mit dieser Struktur ist es möglich, ein schon existierendes Netzwerk nachträglich zu erweitern, ohne komplett neue Adressen zu vergeben. Es müssen lediglich die vordersten Bits im neuen Netz angepasst werden.
Tabelle 2 zeigt die verschiedenen IPv6 Unicast Adressen, welche je nach Netzwerktyp eine andere Struktur in den vorderen Bits aufweisen.

	3bit	13bit	8bit	24bit	16bit	64bit
Global	010	TLA ID	res.	NA ID	SLA ID	Interface-ID
Standortlokal	1111 1110 11 0...0				SLA ID	Interface-ID
Linklokal	1111 1110 10 0...0					Interface-ID
IPv4-kompatibel	0...0					IPv4 Adresse

Tabelle 2: IPv6 Unicast Adressen

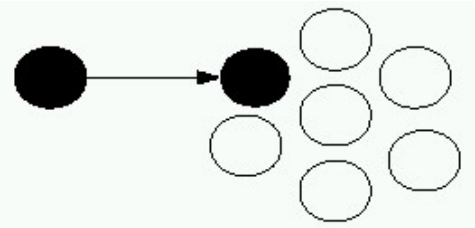


Abbildung 2: Prinzip der Anycast-Adressen

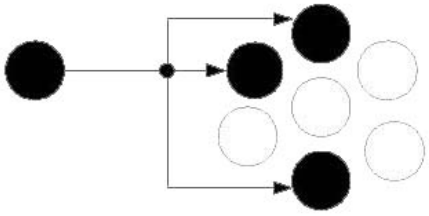


Abbildung 3: Prinzip der Multicast-Adressen

3.1.3 Anycast-Adressen

Mit der Anycast Methode wird ein Paket an eine grosse Anzahl von potentiellen Interfaces geschickt, wobei die Kommunikation mit demjenigen weitergeführt wird, der gemäss der Routing-Metrik am nächsten liegt (vgl. Abb. 2).

3.1.4 Multicast-Adressen

Eine Sendung an eine Multicast Adresse bewirkt, dass das Paket von mehreren bestimmten Gruppenmitgliedern empfangen wird (vgl. Abb. 3). Diese Adressen können zum Beispiel dafür verwendet werden, wenn ein Paket an alle Router geschickt werden muss.

8bit	4bit	4bit	112bit
1111 1111	Flags	Scope	Group ID

Tabelle 3: Ipv6 Multicast Adressen

Flags zeigt an, ob es sich um permanent eingerichtete Adressen handelt oder nicht und mit Scope wird der Gültigkeitsbereich der Gruppe definiert.

3.2 IPv6 Paketstruktur

Am Anfang eines Paktes befindet sich der sogenannte Header. Er geht jeweils den Nutzdaten voraus und bezweckt eine Organisation. Die wesentlichste Neuerung in IPv6 besteht darin, dass die Länge dieses Headers variabel ist. Die

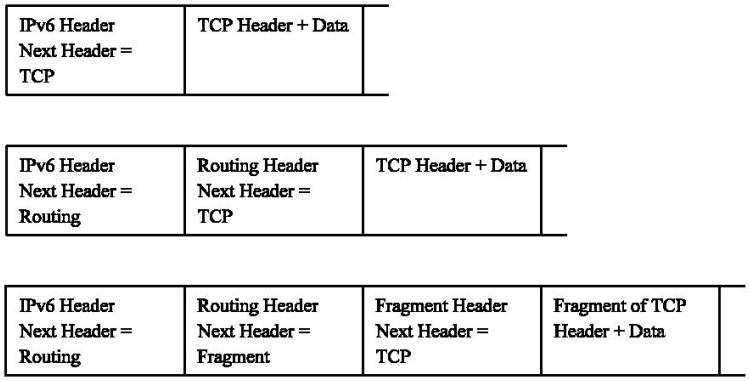


Abbildung 4: Prinzip der Erweiterungsheader

allgemeine Strukur ist so aufgebaut, dass sich am Paketanfang immer der IPv6-Header befindet, welchem je nach Bedürfnis mehrere Erweiterungs-Header folgen können (vgl. Abb. 4) Von den Routern werden nur der IPv6 Header und die beiden Erweiterungen Hop-by-Hop Options und Routing Header beachtet. Die restlichen sind nur im Endsystem von Interesse.

3.2.1 Der IPv6-Header

Abbildung 5 zeigt den IPv6-Header schematisch. Das Feld *Version* definiert die IP-Version des Pakets. Mit dem Setzen des *flow label* kann ein betimmter Datenstrom identifiziert werden. Das heisst, es können Unterscheidungen unterhalb den verschiedenen Internet-Diensten wie z.B. HTML, FTP, Audiotransfer u.v.m vorgenommen werden, was die Vorraussetzung für die komfortable Übermittlung von realzeitfähigen Anwendungen ist. Die Lebenszeit der Pakete wird mit dem *hop limit* beschränkt. Es stellt im wesentlichen ein Counter dar, der bei jedem Router dekrementiert wird. Der Eintrag *next header* beschreibt die Art des angefügten Erweiterungs-Header, in welchem sich dann wieder ein solcher Eintrag befindet. Die totale Länge des Pakets findet sich im Feld *payload length*.

3.2.2 Hop-by-Hop Options

Diese Optionen werden bei jedem Knoten ausgewertet. Es kann zum Beispiel von den Routern eine höhere Beachtung verlangt werden.

3.2.3 Routing Extension

Mit der Routing Erweiterung kann der Weg des Pakets beeinflusst werden. Aus verschiedenen Gründen, wie Dienstqualität oder geringere Kosten, können ganz spezifisch Knoten angegeben werden, die bis zum Ziel passiert werden müssen.

3.2.4 Fragment Extension

Die Paketgrösse ist je nach Netzwerktyp beschränkt, was mit MTU (maximum transmission unit) bezeichnet wird. Die Fragment Extension erlaubt es trotz-

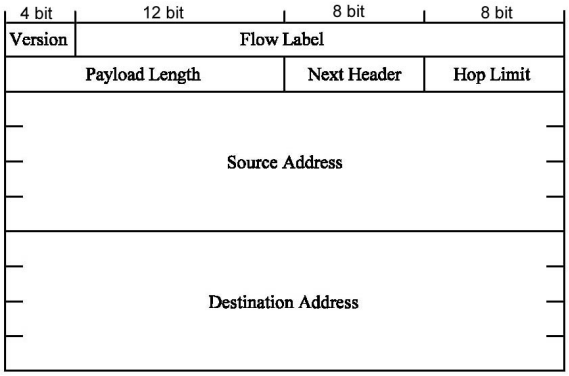


Abbildung 5: Der IPv6-Header

dem eine grössere Einheit zu versenden. Die Daten werden dann in Fragmente aufgeteilt und so versandt.

3.2.5 Authentication Extension

Dieses Erweiterungspaket bestätigt den Urherber des Absenders. Mehr detailliert werden die Sicherheitsaspekte im Kapitel 3.3.2 erläutert.

3.2.6 End-to-End Options

In den End-to-End Options werden diejenigen Informationen übertragen, welche nur vom endgültigen Empfänger von Interesse sind. Dies können Instruktionsanweisungen sein, die über den Paketinhalt Aufschluss geben oder über seine Zusammensetzung.

3.3 Sicherheit

Die Sicherheit des Datenpakettransfers wird standardmässig von zwei Bereichen unterstützt. *Die Authentifizierung* stellt sicher, dass die Daten vom beabsichtigten Absender kommen und dass sie unterwegs nicht verfälscht wurden. Durch die *Verschlüsselung* wird gewährleistet, dass unterwegs keine unberechtigte Einsicht in den Informationsaustausch stattgefunden hat.

3.3.1 Verschlüsselung

Im IPv6 wird der DES-CBC (Data Encryption Standard - Cipher Block Chaining) Algorithmus verwendet. Der IPv6 Verschlüsselungs-Header enthält die dafür nötigen Spezifikationen.
Die Verschlüsselung kann auf zwei Arten erfolgen. Der *Tunnel-Modus* verschlüsselt das gesamte IP-Paket, welchem dann natürlich einen neuen Header vorangestellt werden muss. Dies kann mit dem *Transport-Modus* umgangen werden, welcher lediglich die Nutzdaten verschlüsselt.

3.3.2 Authentifizierung

Die Authentifizierung erfolgt in der speziell dafür vorgesehenem Authentifizierungs-Extension. Mit einem Schlüssel, der beiden Kommunikationspartnern bekannt ist, und dem Datenpaket wird eine 128 bit Kennung berechnet und zur Bestätigung verglichen.

4 Übergang und Kompatibilität

Tabelle 1 zeigt eine IPv4 Adresse im IPv6 kompatiblen Format. Während der Übergangsphase ist es also möglich, beide Formate für eine eindeutige Adressierung zu gebrauchen. Und diese Phase hat am 31. Juli 1999 begonnen, als die erste IPv6-Adresse vergeben wurde. Seither verlief die Geschichte aber ein wenig harzig. Einerseits ist es darauf zurückzuführen, dass noch keine direkte Notwendigkeit zur Einführung besteht und andererseits, dass noch keine IPv6 basierenden Anwendungen existieren.

5 Schlusswort

Das neue Protokoll birgt, wenn es leistet was es verspricht, viele grundlegende Neuerungen in sich. Einerseits sind das Sicherheitsanpassungen und Adresserweiterungen, andererseits aber auch eine Grundlage für neue Technologien. Gerade dies wird dann, wenn der Stein endlich ins Rollen kommt, den Übergang beschleunigen. Experten rechnen, dass bis ins Jahr 2002 die IPv6 Adressen einen Anteil von 10% des gesamten Adressraums erobert haben sollen.

6 Bibliographische Angaben

- T. Braun: IPng - Neue Internet-Dienste und virtuelle Netze, dpunkt Verlag, Heidelberg, Deutschland, 1999, Seiten 55-82.
- S. Thomas: IPng and the TCP/IP Protocols; John Wiley & Sons, Inc., New York, USA, 1996, Seiten 93-126.
- Corporation for National Research Initiatives, IETF - The Internet Engeneering Task Force, http://www.ietf.org, 2000
- Robert M. Hiden, http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html, 1995

HTTP

Hypertext Transfer Protokoll

Marco Graf
Grundlagen der Internet Technologie
Vortrag 5 / 12. 12. 2000

1. Einleitung

Wenn man im World Wide Web herumsurft und nach Informationen sucht, benutzt man einen Browser und alles geht eigentlich wie von alleine. Doch was alles dahinter steckt sieht man nicht. Wieso jetzt diese Seite angezeigt wird, wenn man jene Internet-Adresse eingibt. Wieso ein Browser überhaupt mit anderen Computern kommunizieren kann und wie das geschieht. Wenn man sich jedoch mit dem Thema befasst, stösst man rasch auf das HTTP, das Hypertext Transfer Protocol, welches in dieser Ausarbeitung erläutert wird.

2. Geschichte

Die Geschichte des HTTP beginnt ca. im Jahre 1989. Die Idee war es ein Protokoll zu kreieren, damit zwei Computerprogramme, Server und Client, auf von einander entfernten Computern miteinander kommunizieren können.

HTTP/0.9

Die erste Version, das HTTP/0.9, legte ein Protokoll fest, das nur die Methode `GET` beinhaltete. Es war damit möglich eine Textdatei anzufragen und zu übertragen. Es gab jedoch noch keinen Authentifizierungsmechanismus oder eine Möglichkeit binäre Dateien, wie Bilder oder Musikdateien, zu übertragen. Ausserdem war es nicht möglich festzustellen, ob die Datei vollständig war oder nicht, denn die Anfrage mit `GET` wurde durch das Dokument selbst beantwortet und das Ende durch das Abbrechen der Verbindung gekennzeichnet.

HTTP/1.0

Um die Einschränkungen des HTTP/0.9 zu beheben wurde 1992 mit der Entwicklung des HTTP/1.0 begonnen. Die endgültige Version wurde jedoch erst im Mai 1996 freigegeben. Es war klar, dass es bald schon durch eine neuere Version ersetzt werden sollte und stellte so auch kein Standardprotokoll dar.
Das HTTP/1.0 enthielt nun das Medientypenkonzept MIME womit es möglich war nun auch binäre Dateien zu verschicken. Ausserdem enthielten nun die Nachrichten auch ein Header mit Headerfeldern, die dazu benutzt werden konnten verschiedene Angaben zu Client und zum Transfer zu übermitteln. Es wurden auch neue Methoden implementiert. So war es nun möglich mit der Methode `POST` auch Formulare auf Webseiten auszufüllen und den Inhalt an den Server zu senden. Doch das Problem der Request/Response-Interaktion bestand weiterhin. Das Problem bestand darin, dass bei jedem Request eine TCP/IP-Verbindung aufgebaut werden musste und nach der Response wieder unterbrochen wurde. Dies hatte einen wesentlichen Verlust der Übertragungsgeschwindigkeit zur Folge, weil zum Beispiel bei einer Internetseite nicht nur das HTML-File übertagen werden muss, sondern auch jegliche Bilder, Javascripts, Javaapplets und weitere Dateien, die im HTML-File eingebettet sind. Es war nun auch ein Authentifizierungsmechanismus implementiert, der jedoch noch sehr unsicher war und deshalb eine Verbesserung erforderte.
Weitere Verbesserungsbereiche stellten das primitive Cache-Modell und die fehlende Unterstützung zur teilweisen Übertragung von Dateien dar.

HTTP/1.1

Das HTPP/1.1 erschien im Januar 1997 und hatte wesentliche Neuerungen im Vergleich zu seinem Vorgängermodell.

Um das Problem des Modells der Request/Response-Interaktion zu verbessern wurden verschiedene Methoden entwickelt. Die beiden hervorstechenden Methoden waren P-HTTP (Persistent HTTP) und T/TCP (HTTP over Transaction TCP)

P-HTTP

Die Idee von P-HTTP war es die TCP-Verbindung offen zu lassen und abzuwarten, ob noch weitere Requests für den selben Server eintreffen. Dies hat zur Folge, das die TCP-Verbindung nicht mehr so oft geschlossen, respektive aufgebaut werden muss. Da das Aufbauen der TCP-Verbindung viel Zeit benötigt, erhöht sich somit die Übertragungsgeschwindigkeit.

T/TCP

Bei dieser Methode wird die Verbindung nach der Übertragung zwar immer noch geschlossen, aber mit der Verwendung des T/TCP (Transaction TCP) an Stelle des TCP wird das Aufbauen mehrer Verbinungen zum selben Server wesentlich effizienter.

Schliesslich hat man sich für das P-HTTP, das Modell der persistenten Verbindung, entschieden und es in die neue Version HTTP/1.1 integriert.
Durch die Einführung der persistenten Verbindung konnten die Enden der Ressourcen nicht mehr durch das Abbrechen der Verbindung geschehen. Bei den meissten Ressourcen ist die Grösse jedoch schon von vornherein gegeben, bei allen anderen kommt das Chunked Encoding zum Einsatz. Das Chunked Encoding ermöglicht es Daten in Stücken (chunks) mit ihren eigenen Grössenangaben zu versenden.
Mit der Unterstützung des Header-Felds `HOST` war es nun auch möglich nicht IP-basierende virtuelle Hosts anzusprechen. Das Problem lag darin, dass virtuellen Hosts mehrere IPs zugeordnet werden mussten, was eine rasche Abnahme der zur Verfügung stehenden IPs hatte und auch zu weiteren Problemen im Netzwerk führte. Mit der Angabe des Clients im Headerfeld `HOST`, an wen der Request gesendet werden soll können nun mehrere virtuelle Hosts die selbe IP tragen. Da das neue Header-Feld HOST nur einen Sinn ergibt, wenn es auch verwendet wird, erhält man ohne Verwendung des Feldes eine Fehlermeldung.
Es wurden auch wieder weitere Methoden implementiert wie zum Beispiel `DELETE`, `OPTIONS`, `PUT` und `TRACE`. Auch war es nun möglich nur Teile von Dateien anzufordern. Dies ist sehr nützlich, wenn bei der Übertragung grosser Ressourcen die Verbindung abbricht. So kann man mit der Übertragung dort weiterfahren wo man war und muss nicht von vorne beginnen.
Mit der eingeführten Content Negotiation kann zwischen verschiedenen Darstellungsformen wie zum Beispiel Sprache, Qualität oder Codierung gewählt werden. Die Funktion der Content Negotiation wird im Abschnitt 5 noch erläutert.
Das Authentifikationsverfahren wurde so verbessert, dass nun Benutzername und Passwort verschlüsselt versendet werden.
Somit waren einige wichtige und nützliche Verbesserungen des HTTPs gemacht womit die neue Version auch einige Zeit anhalten sollte.

3. Aufbau einer Nachricht

```
[ start-line ]
[       message-
header
  ...
  ...]

[ message-body
  ...
  ...]
```

Der Aufbau eines Requests und eines Responses sind weitgehend ähnlich. Die erste Zeile stellt die `START-LINE` dar gefolgt vom `MESSAGE-HEADER`. Die `START-LINE` wird bei einem Request `REQUEST-LINE` und bei einem Response `STATUS-LINE` genannt. Im `MESSAGE-HEADER` sind die verschiedenen Headerfelder mit ihren Informationen, je ein Headerfeld pro Zeile. Nach dem `MESSAGE-HEADER` steht eine Leerzeile

und anschliessend kommt der MESSAGE-BODY, der den Inhalt der Nachricht darstellt. Die MESSAGE-HEADER werden in verschieden Gruppen eingeteilt:

- *General Header*
  General Header findet man sowohl bei Request- als auch bei Response-Nachrichten, haben aber keine Auswirkung auf die übertragen MESSAGE-BODY, auch Entity genannt.

- *Entity Header*
  Im Entity Header können Angaben über den Medientyp der Ressource angegeben werden, die im Response geschickt wird oder im Request angefragt wird.

- *Request Header*
  Der Request Header dient dazu Angaben über den Client oder über den Request zu machen, aber ist nicht über die angefragte Ressource.

- *Response Header*
  Der Response Header wird dazu benutzt Informationen zur Übertragung zu übermitteln, die nicht in der STATUS-LINE angegeben werden können. Aber enthalten wiederum keine angaben über die zu übermittelnde Ressource.

Im Anhang A ist eine Tabelle dargestellt, die die verschiedenen Header-Feldern und kurze Erklärungen dazu auflistet.

4. Die verschiedenen Wege des HTTP

Die einfachste Verbindung stellt die direkte Verbindung zwischen Client und Server dar, wie in der Abbildung 4.1 gezeigt.

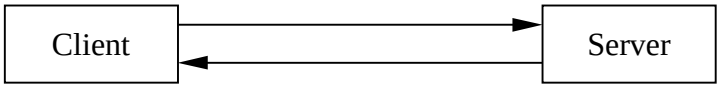


Abb 4.1: Grundlegende Funktionsweise von HTTP

Neben dieser Verbindung können auch weitere Stationen zwischen Client und Server liegen, die sowohl als Client wie auch als Server fungieren. In HTTP sind Proxies, Gateways und Tunnels als übliche Zwischenstationen definiert.

Proxy

Ein Proxy speichert häufig angefragte Ressourcen auf einem Cache und kann diese direkt ausgeben, ohne jeweils eine Anfrage an den Server zu stellen. Er entlastet damit den Server

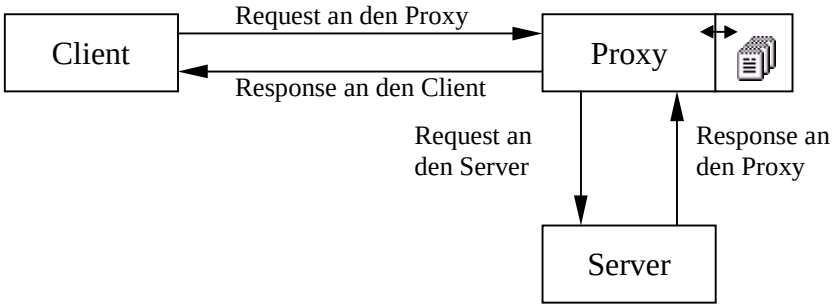


Abb 4.2: HTTP unter Einbeziehung eines Proxies

wesentlich und die angefragten Ressourcen können zum Teil früher beim Client eintreffen.

Der Client muss den Proxy aber speziell angeben, da die Requests an den Proxy gesendet werden und erst der Proxy selbst Requests an den Server schickt.

Gateway

Ein Gateway ist dazu da ein Netzwerk hinter dem Gateway zu verstecken. Der Gateway selbst gibt sich für den Server aus und schickt die Anfragen des Clients an den Server weiter. So können zum Beispiel auch mehrere Computer des lokalen Netzwerkes unter der selben IP mit dem Internet verbunden sein.

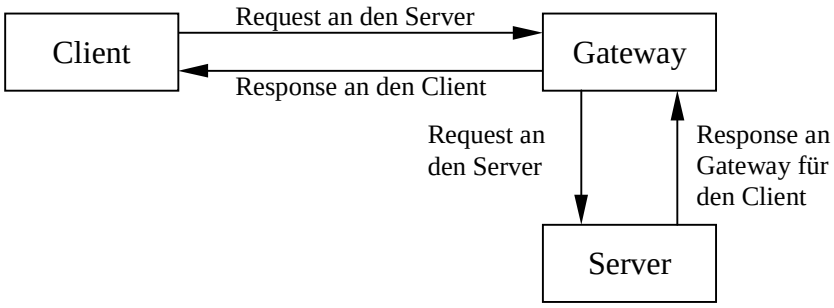


Abb 4.3: HTTP unter Einbeziehung eines Gateways

Tunnel

Bei einem Tunnel wissen, anders als beim Proxy oder Gateway, weder Server noch Client, dass ein Tunnel dazwischen geschaltet ist. Der Tunnel versteht und verändert auch keine Angaben im Request oder Response sondern leitet sie lediglich weiter.

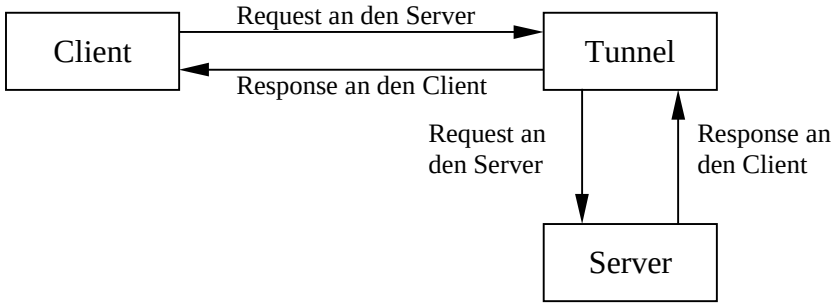


Abb 4.4: HTTP unter Einbeziehung eines Tunnels

5. Content Negotiation

Mit den Angaben, die der Client im Header angibt, wählt der Server zwischen den verschiedenen Arten der Ressource aus und schickt diese dem Client zurück. Die häufigsten Artunterschiede sind Sprache, Qualität und Codierung. Anhand der Kapazität der Netzwerkverbindung eines Benutzers kann zum Beispiel die zumutbare Menge an Daten bestimmt werden und je nach dem eine Graphik oder sonstige Dateien in einer entsprechenden Qualität zurückgesendet werden. Es gibt zwei Arten von Content Negotiation, die Server-Driven Content Negotiation und die Agent-Driven Content Negotiation. Die Kombination der beiden Arten wird Transparent Content Negotiation genannt.

Server-Driven Content Negotiation

Bei der Server-Driven Content Negotiation sendet der Client einen Response mit Angaben über sich. Der Server entscheidet über einen Algorithmus mit den Angaben in den Headerfeldern und aus weiteren Informationsquellen, wie zum Beispiel der Netzwerkadresse, welche der zur Verfügung stehenden Dateien er an den Client zurückschickt. Diese Art der Content Negotiation weist jedoch verschiedene Nachteile auf. So muss ein Algorithmus im Server implementiert werden, der einiges an Rechenleistung verlangen kann. Ausserdem ist es

ineffizient für den Client bei jedem Request alle Angaben über sich mitzuteilen, da die meisten Ressourcen nur in einer Darstellungsform vorhanden sind. Ausserdem kann der Server nicht alles über den Client und die Wünsche des Benutzers wissen.

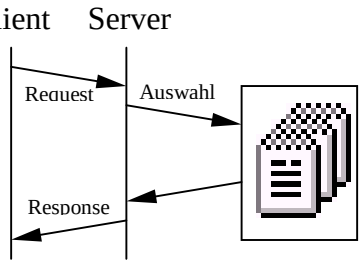


Abb 5.1: Server-Driven Content Negotiation

Agent-Driven Content Negotiation

Bei der Agent-Driven Content Negotiation Antwortet der Server mit einer Liste der zur Verfügung stehenden Arten der Datei. Der Client schickt einen weiteren Request mit der genauen Angabe der Art der Resource und erhält diese im folgenden Response.

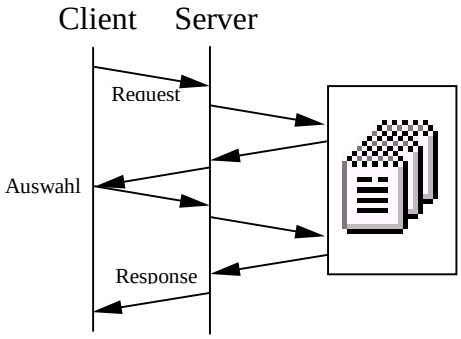


Abb 5.2: Agent-Driven Content Negotiation

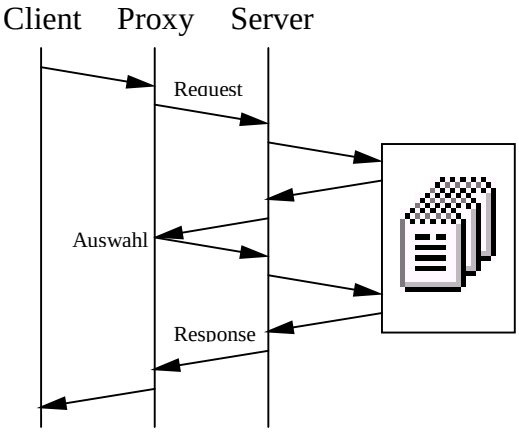


Abb 5.3: Transparent Content Negotiation

Transparent Negotiation

Bei der Transparent Negotiation wird ein Proxy oder eine andere Zwischenstation dazwischen geschaltet der auf der Client Seite eine Server-Driven Content Negotiation ausführt, wobei er den Algorithmus selbst ausführt und auf der Server Seite eine Agent-Driven Content Negotiation. Wenn nun ein Server eine Auswahl der Ressource angibt errechnet der Proxy, welche Ressource für den Client in Frage kommt, schickt den entsprechenden Request an den Server, und der Client erhält schliesslich die ihm entsprechende Datei.

6. Ausblick

Eigentlich sieht man in der heutigen Zeit noch nicht viel von den Möglichkeiten des HTTPs, was darauf hindeutet, dass das HTTP noch nicht ganz ausgenutzt wird. Es wäre weit mehr möglich auf dem Internet. Doch es gibt auch schon wieder andere Möglichkeiten zur Verwirklichung, wie zum Beispiel mit CGI-Scripts oder über Realtime-Streams. Das bedeutet jedoch nicht, dass dies das HTTP ablösen wird - nein im Gegenteil. Die ganzen Möglichkeiten ergänzen sich und es bildet sich langsam ein Netz der Multimedialen Darstellung. Wer weiss was man in Zukunft alles auf dem Internet erleben kann.

7. Quellenangabe

[1] E.Wilde: World Wide Web – Technische Grundlagen; Springer Verlag, Berlin, Deutschland, 1999
[2] http://www.tecchannel.de/internet/208/index.html
[3] http://www.stefan-lenz.ch/glossar/url.htm
[4] http://www.telemedia.de/funktion/glossar.html
[5] http://msdn.microsoft.com/library/psdk/iisref/devs1pev.htm

Angang A: Liste der Headerfelder und ihre Bedeutung

Name des Header-Felds	*Kurzbeschreibung*	*Typ des Header-Felds*
Accept	Typ der Response, die akzeptiert wird	Request
Accept-Charset	Akzeptierte Zeichensätze	Request
Accept-Encoding	Akzeptierte Kodierungen	Request
Accept-Language	Akzeptierte Sprachen	Request
Accept-Ranges	Zulassung von Bereichs-Requests	Response
Age	geschätztes Alter einer Cache-Datei	Response
Allow	Methoden, die von der Ressource unterstützt werden	Entity
Authorization	Authentifizierung des Clients	Request
Cache-Control	Direktiven Angaben für Caching-Systeme	General
Connection	Verbindungsoptionen	General
Content-Base	Absolute URI der Entity zur Auflösung relativer URIs	Entity
Content-Encoding	Kodierung des Message-Bodys	Entity
Content-Language	Angabe der Sprache	Entity
Content-Length	Länge des Message-Bodys	Entity
Content-Location	Angabe der Location der Entity	Entity
Content-MD5	MD5 Digest des Message-Bodys (Fehlerprüfung)	Entity
Content-Range	Grösse und Position einer partiellen Übertagung	Entity
Content-Type	Angabe des Medientyps der Nachricht	Entity
Date	Entstehungsdatum einer Nachricht (Datei)	General
Etag	Das als Validierer verwendete Tag	Entity
Expect	bestimmtes Verhalten das vom Server erwartet wird	Request
Expire	Ablaufdatum der Nachricht	Entity
From	E-Mail-Adresse des menschlichen Benutzers	Request
Host	Internet-Host und Port-Adresse der Ressource	Request
If-Match	Wenn nicht Response mit Statuscode 412	Request
If-Modified-Since	Wenn nicht Response mit Statuscode 304	Request
If-None-Match	Wenn nicht Response mit Statuscode 304/412	Request
If-Range	Anfrage auf partielle oder totale Response	Request
If-Unmodified-Since	Wenn nicht Response mit Statuscode 412	Request
Last-Modified	Datum und Uhrzeit der letzten Änderung	Entity
Location	Addresse der neu kreirten oder verschobenen Daten	Response
MIME-Version	MIME-Version mit der die Nachricht erstellt wurde	General
Max-Forward	Anzahl maximaler Weiterleitungen	Request
Pragma	Implementierungsspezifische Angaben, z.B. no-cache	General
Proxy-Authenticate	Optionen zur Authentifizierung bei einem Proxy	Response
Proxy-Authorization	Authentifizierung beim Proxy	Request
Range	Festlegung des Bereichs zum partiellen Request	Request
Referer	URI der Ressource	Request
Retry-After	Zeitangabe zum erneuten Laden bei einem Fehler	Response
Server	Informationen über den Server	Response
TE	Akztepierte Kodierung, bei Fehler Statuscode 406	Request
Trailer	Header-Felder sind im Trailer einer Nachricht	General
Transfer-Encoding	Angabe der Kodierung der Nachricht	General
Upgrade	Angabe von möglichen weiteren Protokollen	General
User-Agent	Angabe über Client z.B. Webbrowser Mozilla	Request
Vary	Dimension erner Response	Response
Via	Weg der Nachricht zur rückverfolgung	General
Warning	Informationen anstelle von Statuscodes	Response
WWW-Authenticate	Optionen zur Authentifizierung	Response

Anhang B: Methoden des HTTP

Methode	***Beschreibung***
CONNECT	Zur Erstellung einer SSL-Verbindung ohne Einwirken von Proxies
DELETE	Löschen der Ressource auf dem Server fordern
GET	gewöhnliches Anfragen einer Ressource
HEAD	Anfragen einer Ressource ohne MESSAGE-BODY nur den Header
OPTIONS	Info über Kommunikationsoptionen einholen
POST	senden von Daten an eine Ressource
PUT	abspeichern der gesendeten Daten unter der angegebenen URI
TRACE	der Request der beim Server ankommt, wird als Ressource zurückegeben

Anhang C: Glossar

- *Server*

  Ein Server ist ein Computerprogramm, das Requests (Anfragen) entgegennimmt und mit einem Response (Antwort) beantwortet.

- *Client*

  Ein Client ist ein Computerprogramm, das Requests (Anfragen) sendet und darauf einen Response (Antwort) erwartet. Browser wie Netscape und Internet-Explorer sind Programme die einen Client implementiert haben.

- *URL, URI, URN*

  URI (Uniform Resource) ist der Überbegriff für URL (Uniform Resource Locator) und URN (Uniform Resource Name). Eine URL ist der eindeutige Zeiger auf eine Ressource im Internet z.B. „http://www.ee.ethz.ch/“. Sie unterstützt die Protokolle `http:` `ftp:` und `file:`. Bei einer URN ist anders als bei der URL nicht festgelegt, wo die Ressource physisch ihren Standort hat. Es ist so möglich Kopien einer Datei auf verschiedene Orte zu verteilen.

- *MIME*

  Das Medientypenkonzept MIME (Multipurpose Internet Mail Extensions) wird auch in Mails verwendet und stellt eine Möglichkeit dar, Dateien zu verschicken mit Angaben über den Typ der Datei.

- *Host*

  Ein Host ist ein an ein Netzwerk angeschlossener Computer, auf den zugegriffen werden kann.

- *Virtueller Host*

  Ein virtueller Host ist in einem Host integriert, aber träg einen anderen Namen. Bei IP-basierenden virtuellen Hosts stellen die IPs die verschiedenen Namen dar, bei einem nicht IP-basierenden virtuellen Host hat der Name zum Beispiel die Form „www.ethz.ch“.

Die Beschreibungssprache

HTML

Vortrag Nr. 6 des PPS-Seminars „Grundlagen des Internet“
12.12.2000
Autor: Samuel Zimmerli
szimmerli@student.ethz.ch

1. Einführung

HTML (Hyper Text Markup Language) ist die zum Gestalten einer Website verwendete standardisierte Beschreibungssprache. In dieser Sprache legt der Autor die logische Struktur seiner Dokumente fest (Absätze, Überschriften usw.). Die Text- und Hypertext-Informationen werden auf den Server (Verkäufer, Bedienender) übertragen und gespeichert. Auf die gespeicherten Informationen greift der Benutzer mit einem Web-Browser (Client-Programme für den Zugriff auf WWW-Server) zu. Der Browser verwendet die HTML-Dokumente danach zum Erstellen einer formatierten Darstellung, also einer Website.

Die Entwurfsziele von HTML lassen sich wie folgt zusammenfasse:

- Einfachheit: HTML soll einfach zu verwenden sein, so dass seine Anwendung leicht ist und viele Anwender ermutigt werden HTML einzusetzen.
  - → Anzahl der HTML-Konstrukte ist begrenzt und standardisiert, die Seiten können sich daher sehr ähneln, wenn sie auf gleiche Konstrukte zurückgreifen.
  - → Interessenskonflikt mit der Zielsetzung der Leistungsfähigkeit.
  - → HTML-Files können mit einfachen Text-Editoren (sogenanntes Klartext-Format) erstellt werden. Es ist also keine bestimmte Software nötig.
- Leistungsfähigkeit: HTML soll leistungsfähig genug sein, um eine große Anzahl möglicher Anwendungen zu unterstützen.
  - → HTML muss allgemein genug gehalten werden. Interessenskonflikt mit der Zielsetzung der Einfachheit der Bedienung.
- HTML soll eine Sprache sein, die sich mehr mit dem Inhalt als mit der Darstellung befasst.
  - → HTML wird dadurch von einer Plattform unabhängig.

2. Geschichtliche Entwicklung von HTML und dessen Funktionen

2.1 Der SGML-Standard

Als "Presentation-based Markup" bezeichnet man das Verfahren, die Angaben für die Formatierung der Texte mit Hilfe spezieller Zeichenkombinationen direkt in den Text einzugeben. Darauf basierten die ersten Programme für die Ausgabe von Texten im Zeitalter der Lochkarten.
Als "Content-based Markup" bezeichnet man das Verfahren, den Inhalt mit einem logischen Markup zu versehen, d.h. mit Angaben, welche logische Bedeutung die Text-Elemente haben (z.B. Überschrift, Liste usw.). In welchem Layout diese Elemente dargestellt werden (z.B. Schriftarten, Seitenformat usw.), wird separat davon in sogenannten Style-Files festgelegt (→ Medienunabhängigkeit). Dadurch kann derselbe Inhalt durch die Verwendung verschiedener Style-Files in verschiedenen Formaten verwendet werden oder umgekehrt. Das Prinzip des "Content-based Markup" wurde Mitte der 80er-Jahre zur "Standard Generalized Markup Language" SGML verallgemeinert, und die im World Wide Web verwendete "Hypertext Markup Language" (HTML) baut auf diesem SGML-Standard auf.

2.2 HTML und HTML+

Im Mai 1989 wurde ein Vorschlag für ein verteiltes Hypermedia-System am europäischen Kernforschungszentrum CERN in Genf verfasst. HTML als Sprache für die Übertragung von Hypermedia-Dokumenten bildete einen zentralen Teil dieses Konzepts. Im September 1990 begann die Entwicklung eines ersten Prototyps, genannt HTML, zum Erstellen von Web-Softwares. Ende 1991 erschien die verbesserte Version HTML+.
Verglichen mit dem neusten Standard war die Funktionsweise von HTML und HTML+ sehr einfach. Sie enthielten Elemente für Textüberschriften, Listen und das Element A (siehe Exkurs 1).

Exkurs 1: Element A

Eine der wichtigsten Eigenschaften von HTML ist die Möglichkeit, Verweise zu definieren. Verweise (Hyperlinks/Element A) können zu anderen Stellen im eigenen Projekt führen, aber auch zu beliebigen anderen Adressen im World Wide Web. Das Bewegen zwischen räumlich weit entfernten Rechnern wird so auf einen Mausklick reduziert. Auf dieser Grundidee beruht das gesamte World Wide Web.
Mit Hypermedia bezeichnet man Multi-Media-Systeme (Texte, Bilder und Töne) mit Querverweisen wie bei Hypertext. Z.B: anklickbare Werbe-Grafiken (Banner)

2.3 HTML 2.0 und HTML 3.0

Es zeichnete sich ein schnelles Wachstum des WWW ab. Von kommerziell orientierten Browserherstellern, wie dem 1994 gegründeten Unternehmen Netscape, wurden ständig neue browserspezifische Erweiterungen von HTML entwickelt um sich von der Konkurrenz abzuheben.
Die Verwendung solcher browserspezifischer Erweiterungen hatten jedoch zur Folge, dass der Aufruf der Seite mit anderen Browsern nicht mehr möglich war. Im November 1995 wurde daher eine neue, HTML 2.0 genannte Version offizieller Sprachstandard, die die neuen, von den verschiedenen Browser-Herstellern hinzugefügten Funktionen, versuchte zusammenzufassen.
Doch HTML 2.0 wurde allgemein als Enttäuschung empfunden, da gerade Netscape in seiner Entwicklung schon wieder viel weiter war. Auch der, auf der Empfehlung vom 1994 gegründeten World Wide Web Consortium (W3C), basierende Entwurf von HTML 3.0 hinkte schon während des Ratifizierungsprozesses der Entwicklung hinterher und erreichte nie einem offiziellen Status.

2.4 HTML 3.2

Im Januar 1997 erschien die offizielle Version von HTML 3.2, deren Standard von W3C festgelegt wurden. In dieser Zeit begann die verstärkte Zusammenarbeit zwischen dem W3-Konsortium, dem vor allem Akademiker angehören, und kommerziell orientierten Software-Herstellern.
Die Version 3.2 sah eine Möglichkeit vor, vom Autor definierte Markups für spezielle Elemente zu verwenden und in eigenen "Style-Sheets" (Siehe Exkurs 2) anzugeben. Auch Tabellen und Bilder wurden zu offiziellen Bestandteil von HTML.
Ungefähr gleichzeitig mit dem Erscheinen der Version 3.2 erweiterete Netscape ihren gleichnamigen Browser mit der Framesfunktion (Frames dienen dazu mehrere HTML Dokumente auf einer Seite anzuzeigen. Zum Beispiel in Form eines Hauptmenüs seitlich des Bildschirms, durch dessen Auswahl man navigieren kann, ohne dass es verschwindet). So entsprach auch die Version 3.2 bereit bei der Veröffentlichung nicht mehr dem Stand der Dinge.

Exkurs 2: Style-Sheets

Es handelt sich dabei um eine Sprache zur zentralen Definition von Formateigenschaften einzelner HTML-Befehle. Also eine Definition eines Befehls (im Head des Dokuments oder in einem seperaten File gespeichert) die während des ganzen HTML-Text gültig ist. Style-Sheets geben dem Autor die Möglichkeit, das zum logischen Markup gehörende Layout zu verfeinern und progammiertechnisch zu vereinfachen bzw. die HTML-Datei zu verkleinern.
Auf diese Weise kann man zum Beispiel für große Projekte einheitliche Layouts entwerfen. Mit ein paar kleinen Änderungen in einer zentralen Style-Sheet-Datei kann dann für hunderte von HTML-Dateien bequem das Layout geändert werden.

2.5 HTML 4.0

HTML 4.0 wurdeim Februar 1998 als Sprachstandard verabschiedet. Mit HTML 4.0 hat das W3-Konsortium gezeigt, daß es bereit ist, sich mit den kommerziellen Entwicklungen im WWW

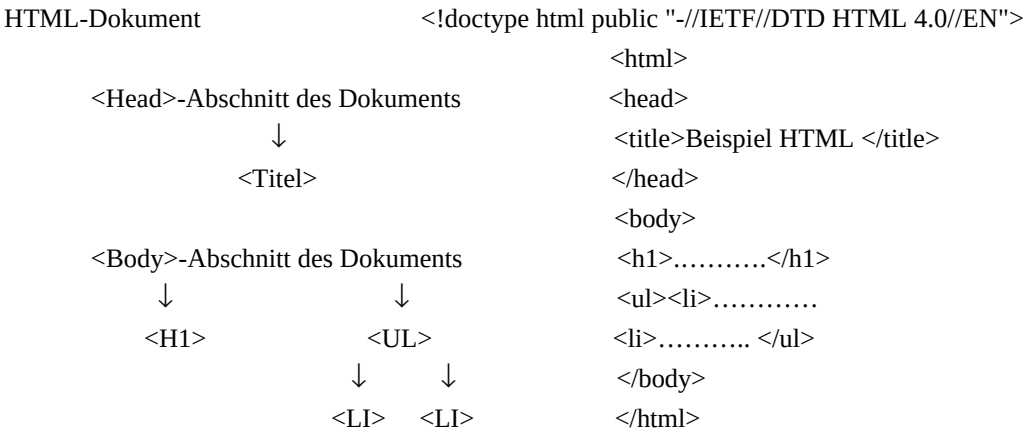
auseinanderzusetzen und seine Rolle als unabhängiges Standardisierungs-Gremium zu nutzen um den Versuche einzelner Hersteller, eigene Entwicklungen durch die Verbreitung eigener Browser als Pseudo-Standard festzulegen, entgegenzuwirken.
HTML 4.0 bindete neben Frames auch Scriptsprachen ein. Zudem ein verbessertes Tabellenmodell, Formulare und eine Unterstützung für die allgemeine Eingliederung von Multimediaobjekten.
Damit im Zuge folgender HTML-Versionen älteren Websites nicht ungültig würden, definierte HTML drei DTD`s (Document Type Definitions), welche beschreiben, wie HTML-Elemente zur Bildung eines gültigen HTML-Dokument verwendet werden müssen:

Transitional DTD: Diese DTD dient zur Interpretation und nicht zu Erzeugen von HTML-Seiten. Da ältere Elementen zwar enthalten aber nicht gebraucht werden sollten.
Strict DTD : Hier sind nur die aktuellen HTML 4.0 Konstruktionen enthalten.
Frameset DTD : Diese ist speziell für die nun unterstützten Frames. Die Framesets werden damit genauer spezifiziert.

3. Grundlegender Aufbau eines HTML 4.0 Dokuments

3.1 Übersicht

Das Beschreibungsschema von HTML geht von einer hierarchischen Gliederung aus. Wobei die DTD`s eine aus potentiellen Bäumen bestehender Menge beschreiben. HTML beschreibt Dokumente. Dokumente haben Eigenschaften wie zum Beispiel einen Titel oder eine Hintergrundfarbe. Der eigentliche Inhalt besteht aus Elementen, zum Beispiel einer Überschrift. Einige dieser Elemente haben wiederum Unterelemente wie zum Beispiel eine Aufzählungsliste besteht aus einzelnen Listenpunkten:



Aufbau eines HTML-Dokuments. Bei dem in diesem Beispiel verwendeten Tag <UL> handelt es sich um eine Liste mit zwei Listenpunkten <LI> und einer Überschrift <H1>

Der Head-Teil enthält Informationen über das Dokument, die nicht zum eigentlichen Dokumentsinhalt gehören. Im Body-Teil wird dann der Dokumentsinhalt definiert, d.h. der Teil des Dokuments, der tatsächlich vom Browser als Website dargestellt wird.

3.2 Der Document Head

Er wird mit <HEAD> und </HEAD> abgegrenzt und enthält den Titel <TITLE> des HTML-Dokuments, welcher zum Beispiel in den Suchmaschinen angegeben wird. Außer dem Titel kann der Header noch weitere Befehle enthalten wie JavaScripts <SCRIPT>, Dokumentbeschreibungen, die über das Element <META> eingefügt werden und Style Sheets.

Der Document Head ist insbesondere zum Angeben von Informationen von Bedeutung, die von automatisierten Clients wie beispielsweise Suchmaschinen zur Verarbeitung verwendet werden können.

3.3 Der Document Body

Der Body-Teil enthält den eigentlichen HTML-Text. Diesen kann man mit Hilfe von bestimmten Elementen strukturieren und formatieren (Siehe dazu das im Anhang beigefügte Beispiel mit Quellentext).
Der BODY-Befehl selbst kann folgende Attribute enthalten, die die Seitengestaltung betreffen:

- BACKGROUND bestimmt ein Hintergrundbild
- BGCOLOR bestimmt eine Hintergrundfarbe
- TEXT die Farbe des Textes
- LINK die Farbe von noch nicht benutzten Hyperlinks
- VLINK die Farbe von benutzen Hyperlinks
- ALINK die Farbe der Hyperlinks beim Draufklicken.

4. Schlusswort und Ausblick

HTML wird sicher auch noch in nächster Zukunft eingesetzt werden,doch die zukünftige Webpage wird man mit Hilfe von XML (Extensible Markup Language) und Style Sheets strukturieren. XML ist eine Metasprache zur Definition von eigenen Markup-Sprachen. Wie die Elemente dann sichtbar dargestellt werden sollen, kann mit Style Sheets definiert werden. Somit wäre man nicht mehr an die beschränkte Anzahl von standardisierten HTML-Konstrukten gebunden.
XML-File mit dem Inhalt der Information und Style Sheet-File mit den Layout-Angaben werden separat auf dem Server abgespeichert. Im Gegensatz zu HTML-Dateien, wo eine konsequente Trennung zwischen Information und Layout fehlt, hat man hier die Möglichkeit, denselben Inhalt wahlweise in verschiedenen Layouts darzustellen.

5. Quellenangaben:

- E.Wilde: World Wide Web - Technische Grundlagen; Springer Verlag, 1999 Berlin, Seiten 191-249
- PPS-Seminar-Unterlagen
  http://www.tik.ee.ethz.ch/~stiller/GIT.d.html
- Hubert Partl, Einführung HTML
  http://www.boku.ac.at/htmleinf
- Stefan Münz, Selfhtml;
  http://computing.ee.ethz.ch/.soft/selfhtml/selfhtml.htm;
  http://www.teamone.de/selfaktuell/

6. Anhang: Beispiel eines einfachen HTML-Dokuments

```
<!doctype html public "-//IETF//DTD HTML 4.0//EN">

<html>
<head>
<title>Beispiel HTML </title>
</head>
<body>
<h1 align=center >Grundlegende Tags in HTML</h1>
<pre>
</pre>
<p>
<ol>
<li>Textformationen
<ol type=a>
<li>Absatz  <b> p </b>;  Zeilenwechsel <b> br </b>; Anordnung <b>  align= </b> und formatierte
Texteingabe <b> pre </b>.
<li>hervorgehobenes Wort <b> em und strong  </b>; Schriftgrösse <b> font size= </b> und Farben
<b> font color= </b>
<li>Tabellen <b> table </b>
</ol>
<li>Links  <b> a href=</b>
<li>Bilder  <b> img src=</b>
</ol>
<pre>
</pre>
<ul>
<li>Den folgenden Text lässt sich mit den Elementen <b> p; br und align= </b> oder mit der
formatierten Texteingabe <b> pre </b> darstellen.
<p align=center>
Telefonat:
<p align=center>
 "Ist da die Beratungsstelle für anonyme Alkoholiker?"
<br align=center >
 "Ja, da sind sie hier richtig. Kann ich ihnen Helfen?"
<br align=center >
 "Ich glaube schon, können sie mir sagen, wie man eine Kiwibowle ansetzt!"

<pre>
                                                Telefonat:

               "Ist da die Beratungsstelle für anonyme Alkoholiker?"
                "Ja, da sind sie hier richtig. Kann ich ihnen Helfen?"
 "Ich glaube schon, können sie mir sagen, wie man eine Kiwibowle ansetzt!"


</pre>
<li>Zur Darstellung  des nächsten Textes werden die Tags <b> strong; em; font size= und font
color= </b> verwendet.

<p align=center>
Die Kunst des <strong><font color="#0000FF"><font size=4>Umgangs mit
Menschen</font></font></strong> besteht darin, sich geltend zu machen, ohne andere
<br align=center>
<strong><font color="#0000FF"><font size=4>unerlaubt zurückzudrängen. </font></font></strong>
<p align=center><em>Knigge</em>
<pre>
</pre>
<p>
<li>Nun folg noch eine Tabelle <b> table </b> zwei Links <b> a href=…</b>, die auf Websites
mit HTML-Dokumentationen verweisen und ein Bild  <b> img src=</b>.
<ul>
<pre>
</pre>
<table border align=center>
<tr><td align=right> Name des Autors
    <td align=left> Verweis und Name der Dokumentation
<tr><td align=right> Hubert Partl
    <td align=left><a href="http://www.boku.ac.at/htmleinf"> Einführung HTML </a>
<tr><td align=right> Stefan Münz
    <td align=left><a href="http://computing.ee.ethz.ch/.soft/selfhtml/selfhtml.htm"> Selfhtml
</a>
</table>
<img src=" http://www.ethz.ch/img/ethlogo.gif " alt="Text" width ="300" align=right >
</body>
</html>
```

7. Darstellung des HTML-Dokumentenbeispieles

Grundlegende Tags in HTML

1. Textformationen
    a. Absatz **p** ; Zeilenwechsel **br** ; Anordnung **align=** und formatierte Texteingabe **pre**
    b. hervorgehobenes Wort **em und strong** ; Schriftgröße **font size=** und Farben **font color=**
    c. Tabellen **table**
2. Links **a href=**
3. Bilder **img src=**

- Den folgenden Text lässt sich mit den Elementen **p; br und align=** oder mit der formatierten Texteingabe **pre** darstellen.

Telefonat:

"Ist da die Beratungsstelle für anonyme Alkoholiker?"
"Ja, da sind sie hier richtig. Kann ich ihnen Helfen?"
"Ich glaube schon, können sie mir sagen, wie man eine Kiwibowle ansetzt!"

Telefonat:

"Ist da die Beratungsstelle für anonyme Alkoholiker?"
"Ja, da sind sie hier richtig. Kann ich ihnen Helfen?"
"Ich glaube schon, können sie mir sagen, wie man eine Kiwibowle ansetzt!"

- Zur Darstellung des nächsten Textes werden die Tags **strong; em; font size= und font color=** verwendet.

Die Kunst des **Umgangs mit Menschen** besteht darin, sich geltend zu machen, ohne andere **unerlaubt zurückzudrängen**.

Knigge

- Nun folg noch eine Tabelle **table** zwei Links **a href=,** die auf Websites mit HTML-Dokumentationen verweisen und ein Bild **img src=**.

Name des Autors	Verweis und Name der Dokumentation
Hubert Partl	Einführung HTML
Stefan Münz	Selfhtml

Die Datenstrukturierungssprache XML

XML

Extensible
Markup
Language

Lukas Haemmerle

Vortrag Nr. 7 / 23. Januar 2001

PPS Seminar: Grundlagen der Internet-Technologie

Einleitung

Die *Hypertext Markup Language (HTML)* dürfte heute den meisten Menschen ein Begriff sein, woran man erkennen kann, dass sich diese Webseiten-Beschreibungssprache als Standard durchgesetzt hat, was vor allem wegen ihrer einfachen Programmierung und durch die Unmgengen an HTML-Editoren erklärt werden kann.
In letzter Zeit ist jedoch vermehrt von der *Extensible Markup Language (XML)* die Rede, die als Nachfolge-Sprache von HTML gehandelt wird. In diesem Dokument wird beschrieben, was XML ist, wozu es gebraucht werden kann und wo es angewendet werden kann.

XML-Grundlagen

Was ist XML ?

Die *ExtensibleMarkupLanguage (XML)* ist eine Teilmenge der Standard Generalized *Markup Language (SGML)* und ist wie jene eine Metasprache für das Definieren von anwendungsspezifischen Dokumenttypen.

Entstehung von XML

XML wurde von einer speziellen Arbeitsgruppe entworfen, die 1996 unter der Schirmherrschaft des *World Wide Web Consortium (W3C)* gegründet wurde. Bereits im November 1996 waren die Grundzüge von XML festgelegt, und im April 1997 war man sich einig bezüglich den meisten Feinheiten. Im Dezember 1997 wurde aus dem Entwurf eine 'Recommended Proposition'. Seit dem 10. Februar 1998 ist XML eine 'Recommendation'. Wenn Veränderungen vorgenommen werden sollen, dann müssen diese in eine neue Version einfliessen. Die aktuelle Version ist XML 1.0.
Die Motivation XML zu entwerfen bestand hauptsächlich in folgenden drei Beweg-gründen:

1. Die mangelnde Flexibilität von HTML:
   HTML definiert nur einen bestimmten Dokumenttyp. Dies mag zwar für eine grosse Anzahl von Anwendungen genügen, doch in vielen Bereichen wäre eine anwendungsspezifischere Dokumentstrukturierung geeigneter.

2. Die Komplexität von SGML:
   XML beruht - wie in der Einleitung erwähnt - auf SGML, welche auch die Grundlage für HTML bildet. SGML ist allerdings sehr umfangreich und enthält eine Reihe von Features, die eine Implementierung von SGML-verarbeitender Software erschweren.

3. Das Fehlen von Meta-Informationen bei HTML:
   HTML hat sich im WWW weit verbreitet, doch die Art wie sie Informationen beschreibt ist limitiert. Man hat den Text und kreiert mit HTML ein Layout, so wie es nachher im Browser erscheinen soll. Aber die einzelnen Tags sagen nichts darüber aus, was für Informationen sie enthalten.
   Was bei HTML fehlt, sind Information darüber, was in dem jeweiligen Element enthalten ist.

Man hat sich deshalb für die Entwicklung von XML folgende Ziele gesetzt:

- *XML soll sich im Internet auf einfache Weise nutzen lassen.*
- *XML soll ein breites Spektrum von Anwendungen unterstützen.*
- *XML soll zu SGML kompatibel sein.*
- *Es soll einfach sein, Programme zu schreiben, die XML-Dokumente verarbeiten.*
- *XML-Dokumente sollen leicht zu erstellen sein.*
- *XML-Dokumente sollten für Menschen lesbar und angemessen verständlich sein.*
- *Die Zahl optionaler Merkmale in XML soll minimal sein, idealerweise Null.*

XML wurde also so entworfen, dass die nützlichsten Teile von SGML erhalten bleiben ohne dabei deren Fähigkeiten zu opfern, beliebige Datentypen zu strukturieren. Man wollte auch, dass die Verarbeitung von XML-Dokumenten einfacher und schneller erfolgen kann, als dies bei SGML der Fall ist.
So entstand eine Spezifikation von XML, die einen recht einfachen Dialekt SGML beschreibt und die es ermöglicht XML-Dokumente auf einfache Art und Weise zu kreieren, zu verwalten und auf dem Netz zu publizieren.

Dokumenttyp und Elementtyp

Im obigen Abschnitt war die Rede von Dokumenttyp. In diesem Abschnitt wird beschrieben, was ein Dokumenttyp ist.
Zwei Dokumente haben den selben Dokumenttyp, wenn sie gleiche Elementtypen verwenden und diese alle in der selben Weise verschachtelt sind. Ein Elementtyp wiederum besteht aus einem Start-Tag und möglicherweise einigen Attributen, dem Inhalt des Elements und einem abschliessenden End-Tag.
Dies sieht dann zum Beispiel so aus:

```
<!ELEMENT autor #PCDATA>
<autor geschlecht=”maennlich”>Max Frisch</autor>
```

Die erste Zeile ist die Definition des autor-Element. Dazu mehr im nächsten Unterkapitel.
Ein Element kann drei mögliche Arten von Inhalten haben: Daten, weitere Elemente oder sowohl Daten als auch weitere Elemente.
Ähnlich wie bei HTML ist es möglich für ein Element eines oder mehrere Attribute zu definieren um das Element weiter zu spezifizieren. Dies geschieht in sogenannten Attributlisten. Ein kleines Beispiel dazu:

```
<!ATTLIST autor
geschlecht (maennlich|weiblich) #REQUIRED
nationalitaet CDATA #IMPLIED
>
```

Diese Attributliste spezifiziert das Element <autor> etwas näher indem es weitere Informationen bietet zum Geschlecht des Autors, das unbedingt angegeben werden muss und zur Nationalität des Autors, die nicht unbedingt angegeben werden muss.
Falls es eine Menge von Dokumenten mit den selben Elementtypen gibt, dann gehören diese Dokumente alle dem selben Dokumenttyp an. HTML-Dokumente haben zum Beispiel alle den selben Dokumenttyp, nämlich den Dokumenttyp HTML.
Die Grundidee von XML ist also dafür zu sorgen, dass es Dokumente gibt, die alle in ihrem Aufbau gewissen Grundmustern folgen. Dies erleichtert die programmierung von XML-verarbei-

tenden Applikationen insofern, als dass man weiss, welche Strukturen in Dokumenten zu erwarten sind.

Dokumenttypdefiniton

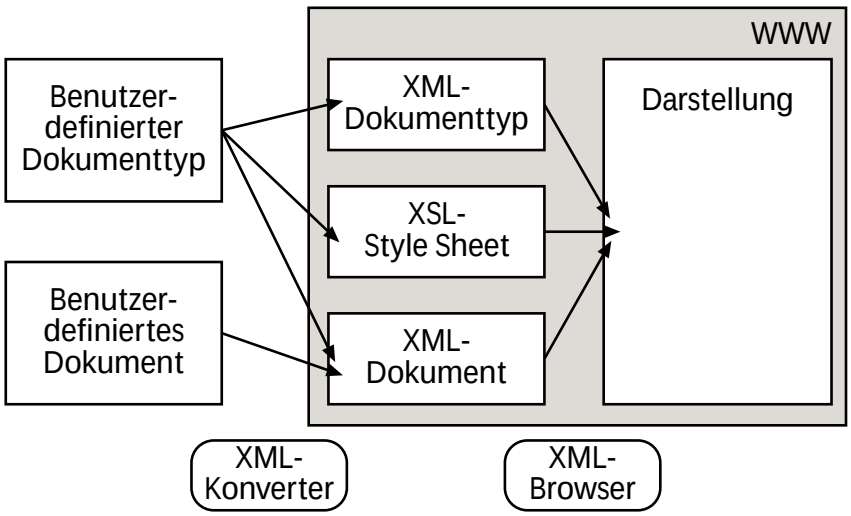
Eine wichtige Rolle bei XML spielen die sogenannten *Dokumenttyp-Definitionen (DTDs)*, welche die Gemeinsamkeiten von Dokumenten festlegen und anhand derer man erkennen kann, ob ein XML-Dokument gültig ist. In der DTD werden alle Elemente und Attribute definiert, die im XML-Dokument benutzt werden. DTDs bestimmen also welche Strukturen, Elemente und Attribute in einem Dokument erlaubt sind und sie sagen den Programmierern, auf was ihre Programme vorbereitet sein müssen. Eine DTD kann in einer separaten Datei untergebracht werden; man spricht dann von einer *externen DTD*, die in einem XML-Dokument folgendermassen angegeben wird:

```
<!DOCTYPE chapter SYSTEM "homofaber.dtd">
```

Die Definitionen können aber auch an den Anfang von einem Dokument erscheinen. In diesem Fall spricht man von einer *internen DTD*. Ein Beispiel für ein Element mit Attributen in einer DTD:

```
<!DOCTYPE webpage SYSTEM [
<!ELEMENT ulink (#PCDATA)*>
 <!ATTLIST ulink
       xml-link            CDATA   #FIXED   "SIMPLE"
       xml-attributes      CDATA   #FIXED   "HREF URL"
       URL                 CDATA   #REQUIRED
]>
```

Es ist auch möglich mehrere DTDs in einem Dokument zu verwenden.
XML ist also einfach ausgedrückt eine Anleitung für das Verfassen von Dokumenttyp-Definitionen, mit denen es möglich ist beliebige Elemente und Strukturen zu definieren.
Mit der DTD sieht eine XML-Seite im Web schematisch so aus wie auf dem nebenstehenden Bild.



Zur Darstellung wird einerseits das XML-Dokument mit dem Inhalt benötig, dessen Dokumenttypdefinition und in den meisten Fällen noch ein Style Sheet, das beschreibt wie der Inhalt darzustellen ist. Style Sheets werden weiter unten speziell behandelt.

XML-Tags

Im Gegensatz zu HTML, wo die Tags in erster Linie Formatierungshinweise für den Browser sind, ist die Aufgabe der XML-Tags vor allem den Inhalt der Tags zu beschreiben. XML-Tags

sagen nichts über die Dokumentstruktur oder die Formatierung aus, sondern liefern lediglich Informationen zum Inhalt des Elements. Dies wird *deskriptives Markup* genannt oder man spricht in diesem Zusammenhang auch von *semantischen Tags*.

Markup

Anders als bei HTML und SGML, wo Markup Minimization erlaubt ist, schreibt XML ein vollständiges Markup vor, also muss optimalerweise zu jedem Start-Tag auch ein Ende-Tag vorhanden sein.
Des weiteren müssen alle Elemente vollständig ineinander verschachtelt sein, so dass ein logischer Dokumentbaum erstellt werden kann, was in erster Linie die Verarbeitung von XML-Dokumenten erleichtern soll.

Aufbau von XML Dokumenten

Ein XML-Dokument sieht im Grossen und Ganzen wie ein normales SGML-Dokument aus. Es beginnt immer mit einigen Deklarationen, die der benutzten Anwendung mitteilen, wie es die Datei verarbeiten soll. Das Programm muss dazu wissen, dass es sich um ein XML-Dokument handelt, und welche Version von XML im Dokument benutzt wurde. Zusätzlich können noch Angaben zum Verwendeten Zeichensatz und zur Verwendung der DTD gemacht werden. Diese Versionsangabe nennt man Processing Instruction:

```
<?XML version="1.0" encoding="ISO-8859-1" standalone="yes"??>
```

XML Processing Instructions stehen zwischen einem *Processing Instruction Open Delimiter (PIO)* und einem *Processing Instruction Close Delimiter (PIC)*, die durch die Zeichenfolge ‘<?’ und ‘?>’ dargestellt werden.
In der zweiten Zeile folgen nun die Angaben zur DTD. Man nennt diese Zeile *Dokumenttyp-Deklaration*. Sie gibt an, in welcher Datei sich die externe DTD befindet (siehe oben). Dabei muss der Name der DTD mit dem sogenannten Wurzelelement, dem äussersten Element, des Dokuments übereinstimmen (Das Wurzelelement ist vergleichbar mit dem <html>-Element bei HTML). Es ist aber auch möglich eine interne DTD zu benutzen oder eine interne DTD in Verbindung mit einer externen. Normalerweise wird die DTD jedoch extern angegeben, was dann etwa so aussieht:

```
<!DOCTYPE Name SYSTEM "Name.dtd">
```

Wenn man die DTD inline angeben will, wird System “Name.dtd“ durch eckige Klammern ersetzt und der Inhalt der DTD in diese Klammern eingesetzt. Danach folgt vielfach die Deklaration des verwendeten Stylesheets:

```
<?xml-stylesheet href="homofaberstyle.xsl" type="text/xsl"?>
```

Die zwei Attribute dienen dazu, die XSL-Datei anzugeben und der Anwendung mitzuteilen, dass diese Datei eine XSL-Datei ist.
Schliesslich kommt der eigentliche Inhalt des Dokuments mit den einzelnen Elementen, die sich alle innerhalb des Wurzelelement des Dokumts befinden.

Beziehung zu HTML

Vielerorts hört man, dass HTML einst durch XML abgelöst werden wird. Dies stimmt allerdings nur bedingt. Einerseits wird HTML auch für die nächste Zeit das Standardformat für einfache Web-Dokumente bleiben, so dass nur solche Anwendungen auf XML zurückgreif, die eine umfassendere Datenstruktur benötigen und andererseits wird das zukünftige HTML als Grundlage XML haben. Beim World Wide Web Consortium wird eine Reihe von HTML-Modulen geschaffen, die auf XML basieren (siehe XHTML). Es gibt dann ein Grundmodul, das bei jeder Arbeit an XML-Dokumenten verwendet wird. Bei Bedarf können weitere Module hinzugezogen werden, ähnlich den Plugins für die Webbrowser. Es gibat nicht nur einen Browser für XML-Dokumente sondern viele, denn kaum ein Browser kann alle XML-Dokumente verarbeiten. So gibt es spezielle Browser für verschiedene XML-Anwendungen. Zum Beispiel für das *WWW*: Der *InternetExplorer 5.0* kann XML-Dokumente fürs Web bereits teilweise darstellen, wenn auch noch über ein *CSS*, das zum grössten Teil von HTML abgeleitet wurde. Der Browser *Mozilla* beeinhaltet ebenfalls schon gewisse XML-Funktionen, wenn auch vorerst nur zu Testzwecken. Alle älteren HTML-Webbrowser hingegen werden die Tags von XML-Dokumenten einfach ignorieren, weil sie diese nicht erkennen können.
Da HTML grosse Ähnlichkeit mit XML hat, kommen HTML-Dokumente sehr oft als Kandidat für die Umwandlung von und zu XML-Dokumenten in Frage, wobei die zweite Variante die kompliziertere ist, weil die meisten HTML-Dokumente oft nicht standardmässige Erweiterungen enthalten und auch sonstige Verstösse gegen die SGML-Syntax die Konvertierung von HTML zu XML erschweren.
Es ist auch möglich HTML komplett mit XML nachzubilden. Deshalb hat das W3C am 26. Januar 2000 eine ‘Recommandation’ zur *Extensible HyperText Markup Language (XHTML)* veröffentlich. XHTML ist ein Satz von einigen DTDs und Modulen, die versuchen HTML 4 nachzubilden. Dokumenttypen der XHTML Familie sind XML basierend und letztlich bestimmt, um in Verbindung mit XML-basierenden Benutzeragenten zu arbeiten.

Einsatzmöglichkeiten und Anwendungen von XML

XML-Dokumenttypen können grundsätzlich überall da zum Einsatz kommen, wo ein Datenaustausch stattfindet, also nicht nur im WWW als Seitenbeschreibungssprache. Die Zahl der möglichen Anwendungsgebiete ist letztlich unbegrenzt.
SGML, der Vorgänger von XML, wird schon seit 1986 erfolgreich eingesetzt. Tausende von Auszeichnungssprachen sind geschaffen worden, welche SGML als Grundlage haben. Es liegt deshalb nahe, XML-kodierten Datenmengen für die Nutzung im Web oder im Intranet aufzubereiten.
Das wichtigste Einsatzgebiet von XML sind momentan noch die Weitergabe von Daten innerhalb der Firmen eines Industriezweigs und zwischen Forschungseinrichtungen. Es hat sich gezeigt, dass sich mit XML alle relevanten Daten so kodieren lassen, dass der Datenaustausch einfach wird und sichergestellt ist, dass die Daten weiterverarbeitet werden können.
Mit XML erhalten Firmen und Organisationen die Möglichkeit, an die eigenen Bedürfnisse angepasste Auszeichnungssprachen zu schaffen. Dies ist natürlich auch besonders interessant für E-Commerce, weshalb beispielsweise einige Firmen in den USA gemeinsam ein neues Protokoll entwickelt haben namens Commerce XML (cXML). Es soll die Bildung von Handelsgemeinschaften im Netz unterstützen so dass sich Verkäufer und Käufer besser über Inhalte und Transaktionen verständigen können.

Es gibt bereits weitere sehr weit verbreitete Anwendungen von XML, auch wenn sie nicht gleich am Anfang als solche zu erkennen sind. Die bekannteste ist sicherlich das *Channel Definition Format (CDF)* von Microsoft, welches zum Datenaustausch dient, und das *Resource Description Framework (RDF)*.
Man sieht also, dass XML nicht primär ein Nachfolger von HTML ist, sondern noch viele andere Verwendungszwecke und Einsatzmöglichkeiten hat.

Vorteile von XML

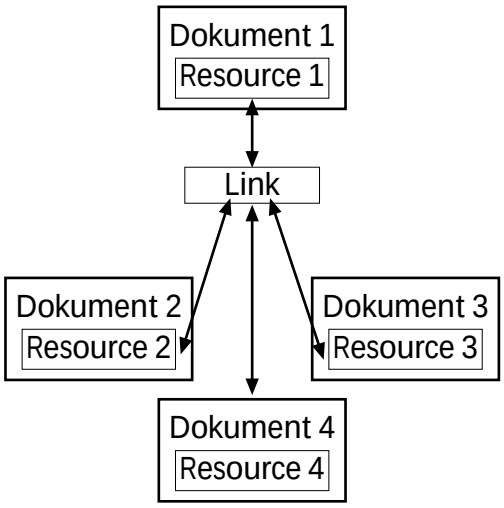
Die Hauptvorteile von XML sind sicher, die Standardisierung und die Offenheit. Mit Offenheit ist in diesem Zusammenhang gemeint, dass lauter Auszeichnungssprachen entstehen, die leicht durchschaubar sind und daher von vielen genutzt werden können. Ein Blick auf die DTD wird zeigen, mit welchen Datenstrukturen man es zu tun hat.
Durch die öffentliche Standardisierung wird sichergestellt, dass mit den zugehörigen Dokumenten gearbeitet werden kann.
Die schon von HTML bekannte Platformunabhängigkeit ist sicher auch erwähnenswert und gewährleistet den problemlosen Datenaustausch zwischen verschiedenen Systemen.
Des weiteren haben nun die Autoren - wie oben erwähnt - die Möglichkeit ihre eigenen Markup-Elemente zu kreieren, die speziell auf ihre Anwendung hin optimiert sind ohne, dass man dabei durch den Umfang von HTML limitiert wird.
Dadurch dass alle Elemente korrekt ineinander verschachtelt sein müssen, resultiert eine Steigerung der Geschwindigkeit um ein Dokument zu prüfen oder darzustellen.
Alle validierten XML-Dokumente können auch in einer SGML-Umgebung benutzt werden, da XML ja eine Teilmenge von SGML ist.
XML bietet auch beim Linking viel mehr als dies bei HTML der Fall ist. Dazu aber mehr weiter unten.

Extensible Style Language (XSL)

Weil durch XML beliebig viele selbstkreierte Tags vorkommen können, benötigt man sogennante Style Sheets, die selbst auch XML-Dokumente sind, zur Darstellung des Inhalts eines XML-Dokumentes.
Was bei HTML die *Cascading Style Sheets (CSS)* sind, ist für XML die *Extensible Style Language (XSL)*, deren Möglichkeiten diejenigen von CSS übertreffen. In den meisten Fällen, wo man XML-Dokument ähnlich darstellen möchte wie heutige HTML-Dokumente, wird man allerdings immer noch die CSS der durch ihren grossen Leistungsumfang etwas komplexeren XSL vorziehen.
Zu erwähnen ist an dieser Stelle auch noch die *Extensible Style Language Transformation (XSLT)* mit deren Hilfe Transformationen von einer Struktur zu einer anderen möglich sind. Dies ist vor allem nützlich um Daten von verschiedenen Strukturen auf ein gemeinsames Format zu bringen.

XML Linking Language

Wie wir gesehen haben ist es in XML-Dokumenten möglich beliebige Elemente und Attribute zu definieren. Falls man aber wie bei HTML sogenannte Hyperlinks gebrauchen will, muss ein definiertes Umfeld vorliegen, dass es XML-Anwendungen ermöglicht, Links innerhalb von Doku-



menten zu erkennen und zu interpretieren. Dieses Umfeld wird durch die *XML Linking Language (Xlink)* definiert, die festlegt wie Links in XML-Dokumente eingefügt werden. Bestehende HREF-Links, die wir von HTML her kennen, sind mit XLL immer noch benutzbar. Aber XLL bietet weit mehr als das:
Eine Neuerung gegenüber von HTML-Links ist, dass es mit XLink möglich ist multidirektionale Links zu benutzen, bei denen man eine Liste mit Verzweigungsmöglichkeiten angezeigt erhält.
Ausserdem ist es auch möglich sogennante 'Out-of-Line-Links' zu benutzen, die in einer externen Datei definiert sind und mit denen man bidirektionale Links definieren kann.
Eine weitere Besonderheit von XLL-Links ist, dass man Links zu und von Daten erzeugen kann, die selbst kein Linking unterstützen und dass man Links von Schreibgeschützten Medien oder Dateien zu anderen Stellen einrichten kann.

Schlusswort

Vorwiegend im industriellen und wissenschaftlichen Bereichen im Zusammenhang mit Datenbanken und Austauschformaten hat XML als Alternative zu SGML schon an Terrain gewonnen und wird vielleicht schlussendlich SGML dank der einfacheren Handhabung verdrängen.
Trotz all der innovativen Möglichkeiten von XML wird HTML auch in absehbarer Zeit noch im *World Wide Web (WWW)* benutzt werden. XML wird jedoch im Web mehr und mehr von grösseren Unternehmen - vorwiegend im Bereich des eCommerce - gebraucht. Das zeigt, dass es sich erst für grössere Anwendungen überhaupt lohnt XML als Markup-Language fürs WWW zu benutzen, da sich erst für solche grossen Seiten die volle Funktionalität von XML entfalten kann.
Eine wichtige Bedingung für die Verbreitung XML im Web ist sicher auch, dass die zwei grössten Webbrowser-Hersteller Netscape und Microsoft ihre Programme weiter auf XML optimieren und dass XHTML auf Anklang stossen wird. Falls dies gelingt hatte XML-Mitentwickler Michael Sperberg-McQueen gar nicht so unrecht als er vor einigen Jahren meinte: "*XML bedeutet einen grossen Schritt nach vorn zu den ursprünglichen Zielen des World Wide Web: Das menschliche Wissen in eine für jedermann erreichbare Form zu packen*".

Quellenangaben:

- E. Wilde: World Wide Web - Technische Grundlagen; Springer Verlag
- H. Behme: XSLT: Transformation von XML-Dokumenten; iX
- Christiane Schulzki-Haddouti: XML- Eine Metasprache knüpft das Netz neu; http://www.first-surf.de/xml_t.htm
- W3C:Deutsche Übersetzung von XML 1.0 Recommendation; http://www.mintert.com/xml/trans/REC-xml-19980210-de.html
- Kuno Dünhölter: Extensible Markup Language; http://members.aol.com/xmldoku

PPS Seminar : Grundlagen der Internet-Technologie
Vortrag 8

Multimedia-Unterstützung im Web

Oder

SMIL

Synchronized Multimedia Integration Language

(ausgesprochen: SMILE !)

Vortrag von Fabio Pezzani
16.01.2001

‘**Synchronized Multimedia is becoming increasingly important on the Web**‘
(Tim Berners-Lee, W3C Director and Inventor of the World Wide Web)

1. Einführung

Am Anfang der Web-Technologie und des Internets konnte man Szenarios wie

Zuerst zeige man Bild1 für 10 Sekunden, nachdem Bild1 5 Sekunden erschienen ist, startet man Audio1 und zeige Bild2 parallel für weitere 20 Sekunden

nicht auf einfache Art und Weise ausführen und programmieren. Also setzten sich verschiedene Gruppen, aus Industrie und des W3C Konsortiums zusammen, um eine einfache und leicht einsetzbare Lösung zu finden. So entstand SMIL.

In dieser Ausarbeitung werde ich versuchen, die wichtigsten Aspekte von SMIL zu erläutern. Ausserdem werde ich auf die Programmiersprache eintreten, indem ich die wichtigsten Befehle erwähne und erkläre. Am Schluss werden Anwendungen gezeigt, d.h. SMIL-Players, die schon mit grossem Erfolg auf der ganzen Welt eingesetzt werden.

1.1 Was ist SMIL ?

SMIL steht für **S**ynchronized **M**ultimedia **I**ntegration **L**anguage

1.2 Wozu dient SMIL ?

SMIL wurde erschaffen, um Medienelemente auf einem Bildschirm oder in einem Fenster zu positionieren und diese aufeinander abzustimmen, d.h. zu synchronisieren. Ausserdem kann sich ein SMIL Dokument auf individuelle Einstellungen, wie z.B. Sprache, Bitrate, Bildschirmgrösse, usw. anpassen.

1.3 Wie erstellt man ein SMIL Dokument ?

Das Einzige was man braucht, um ein SMIL Dokument zu erstellen, ist ... ein einfacher Texteditor !
SMIL sieht der HTML-Sprache sehr ähnlich, basiert aber auf XML und das macht diese Programmiersprache sehr einfach zu lesen und zu verstehen. Dennoch gibt es einige Unterschiede zwischen SMIL und HTML. So z.B. ist bei SMIL die Hierarchie sehr wichtig. Alle Tags müssen daher hierarchisch korrekt geschrieben werden und geschlossen werden (auf ihrer Ebene).

2. Geschichte

Hier sind die wichtigsten Ereignisse tabellarisch aufgelistet:

Mai 1996	5te WWW Konferenz “Sound and Video on the Web “
Nov 1997	Erste Veröffentlichung des SMIL Entwurfes
Feb 1998	Zweite Version des Entwurfes erhältlich
März 1998	Erste SMIL Implementation: HPAS
15.06.1998	**W3C veröffentlicht SMIL 1.0**

Jul 1998	CWI und REAL geben ihre SMIL-Players heraus: GriNS und RealPlayer
Okt 1998	HELIO veröffentlicht den Player SOJA (Antipolis)
Mai 1999	MP3 und Flash4 Unterstützung im RealPlayer G2 angekündigt
Mai 1999	HELIO veröffentlicht SOJA (Cherbourg)
Mai 1999	GriNS 0.5 erscheint
03.08.1999	W3C kündigt die erste Version von SMIL-Boston an
Nov 1999	Apple QuickTime 4.1 unterstützt nun ebenfalls SMIL
Dez 1999	SMIL wird auf Chinesisch übersetzt
Jan 2000	Microsoft stellt den Internet Explorer 5.5 mit SMIL Unterstützung vor
Sep 2000	SMIL-Boston wird zu SMIL 2.0
Dez 2000	SunTREC erstellt einen SMIL 1.0 Java Player

3. Die SMIL Sprache

3.1 Das Erste SMIL Dokument

Die Grundstruktur eines SMIL Dokumentes besteht aus zwei Hauptteilen, dem <head> und dem <body> Teil. Diese Teile werden umrahmt vom <smil> Tag, damit das Ganze als SMIL Dokument erkannt wird.
Die Grundstruktur sieht folgendermassen aus:

```
<smil>
        <head>
                <meta name="copyright" content="Mein Name" />
                <layout>
                        < Layout Tags />
                </layout>
        </head>
        <body>
                < Synchronisations und Medien Tags />
        </body>
</smil>
```

Besonders auffallend ist die Ähnlichkeit zu HTML.

3.2 Der Code im Detail

Der Code beginnt immer mit <smil> und endet mit </smil>. Beide Hauptteile müssen <smil> als Elternteil besitzen.

3.2.1 Layout Abschnitt

Alles, was mit Layout zu tun hat, wird gehört in diesen Abschnitt. Diese Einstellungen gehören in den Kopf des Programms (<head>).

```
<head>
        <layout>
                < layout tags />
        </layout>
</head>
```

3.2.2 Fenster Einstellungen

Um ein Fenster auf dem Bildschirm zu generieren, braucht man den <root-layout> Tag. Dort kann man Grösse, Farbe, Rahmen, usw. einstellen. Dieser Tag wird im <layout> formuliert.
Wir werden hier als Beispiel ein Fenster der Grösse 300x200 Pixels generieren, mit Weiss als Hintergrundfarbe.

```
<root-layout width="300" height="200" background-color=    "white" />
```

3.2.3 <region> Tag

Dieser Tag wird genutzt, um ein Icon im Fenster zu plazieren. Dazu muss man ein **id** (Name) definieren, damit man anhand dieser **id** später auf dieses Icon zurückgreifen kann.

```
<region id="icon" left="75" width="32" height="32" />
```

Dieses Fenster heisst ab jetzt "icon" .

3.2.4 <img> Tag

Um nun tatsächlich ein Icon, z.B. ein Bild in diese Region einzufügen, benutzt man den <img> Tag. Dies geschieht im zweiten Teil des Programms, im <body> Teil.

```
<img src="vim32x32.gif" alt="the vim icon" region="icon" />
```

Mit dem Befehl region="icon" wird nun das Bild in die vorher definierte Region "icon" abgebildet. Der Titel des Bildes definiert man mit dem Befehl 'alt' .
Natürlich kann man auch andere Medien einfügen, die dazugehörigen Tags heissen: audio, text, textstream, video.

3.2.5 Wie passt man ein Element der Region an ?

Dazu wird der Tag <fit> benötigt. Mit diesem Tag kann man entweder das Element an beide Ränder (fit="fill"), an die grössere Seite (fit="meet") oder an die kleinere Seite (fit="slice") anpassen, oder das Element mit einem Scroller einpassen (fit="scroll").

3.2.6 ZEIT : Die Vierte Dimension

Stellen sie sich vor, sie wollen ein Icon nur während 6 Sekunden zeigen. Wie soll das gehen ? Ganz einfach. Wir ergänzen in unserem <img> (oder anderem Medien Tag) die Einstellungen mit dur (duration = Zeitdauer), hier mit dur="6s" . Und wenn man erst nach 2 Sekunden starten will, fügt man den Befehl begin an, hier dur="6s" begin="2s".

```
<img scr="..." alt="..." region=" ..." dur="6s" begin="2s" />
```

Um nun verschiedene Medien in der Abfolge aufeinander abzustimmen, kann man folgende Tags einführen, die die Befehle entweder nacheinander ausführen oder parallel: <seq> dient der sequentiellen Abfolge und <par> der parallelen Abfolge. Diese werden im Body Abschnitt eingesetzt, und zwar vor den Medien Tags.

3.2.7 Synchronisation

In einer Folge oder einem parallelen Abspielen drückt jedes Element (Kind) seine begin und end Eigenschaften bezüglich des Elternteils aus. Man kann aber auch Medien synchronisieren, indem man sie voneinander abhängig macht, z.B. Bild2 startet 3 Sekunden nach Bild1.
Wenn ein Element startet sendet es sein begin Event aus. Ein anderes Element, das wartet, erkennt das begin Zeichen des anderen und fängt je nach Befehl auch an, oder beginnt die Zeit zu zählen. Um ein Element warten zu lassen, muss man es folgendermassen programmieren:

- Wenn man mit einem Anderen gleichzeitig starten will
  ```
  <tag begin="id(specifiedId)(begin)" />
  ```
- Wenn man z.B. 3 Sekunden warten und dann starten will
  ```
  <tag begin="id(specifiedId)(3s)" />
  ```
- Wenn man starten will, nachdem das Andere beendet wurde
  ```
  <tag begin="id(specifiedId)(end)" />
  ```

Beispiel:

```
<body>
        <par>
                <img scr="vim.gif" region="v_icon"  id="vim"
                begin="4s" />
                <img scr="real.gif" region="s_icon"
        begin=id(vim)(2s)" />
        </par>
</body>
```

⇨ Laut diesem Code erscheint das real.gif Bild 2 Sekunden nach dem vim.gif Bild. Dazu kommt noch, dass das vim.gif Bild erst 4 Sekunden nach Programmstart erscheint.

3.2.8 Switching

Ein grosser Vorteil von SMIL ist das sogenannte Switching, indem eine Präsentation dem System des Users angepasst wird.
Der Gedanke ist, dass sich die gleiche Präsentation z.B. in der Sprache, der Bildschirmgrösse, der Bitrate, usw. den Anforderungen selber anpasst. Um das zu ermöglichen, benutzt man den <switch> Tag. Die Regel zur Ausführung des Switch Befehls lautet: Das erste Kind innerhalb des <switch> Tags, dessen Angaben mit den Anforderungen übereinstimmen, wird ausgeführt und als TRUE retourniert. Ein Kind ohne Anforderungen wird automatisch als TRUE zurückgegeben.
<switch> gehört in den <body> Abschnitt.

Beispiel:

- You are English
- Your bitrate is 14000 bps

Und

- Vous êtes français
- Le débit est de 28000 bps

```
<body>
        <switch>
                <par system-language="en">
                        <switch>
                                <text scr="14000bps.en.txt"
                region="bitrate" system-bitrate="14000"/>
                                <text src="28000bps.en.txt"
                region="birate" system-bitrate="28000"/>
                                <text src="another_bitrate.en.txt"
                region="bitrate" />
                        </switch>
                </par>
                <par system.language="fr">
                        < ... />
                </par>
                <text src="unknown_language.txt" region="main" />
        </switch>
</body>
```

Selbstverständlich kann man auch verschiedene Überprüfungen in einen Tag zusammenziehen. Dies wird jedoch etwas komplizierter.

4. SMIL Anwendungen

In diesem Kapitel werden verschiedene SMIL Players und andere Anwendungen erläutert und auf ihre Kompatibilität mit File-Formaten untersucht.

4.1 SMIL Players

4.1.1 RealPlayer G2

Der RealPlayer wurde 1998 von der Firma REAL entwickelt, die auch an der Entwicklung von SMIL teilnahm. Im Juli 98 kam der erste RealPlayer G2 an die Öffentlichkeit.
Dieser Player wurde kostenlos ins Internet gestellt. Bis Heute kann man den RealPlayer (unterdessen Version 8) kostenlos herunterladen, genauer gesagt nur eine Basic Version des Players. Dieser Player unterstützt sehr viele File-Formate und kann sogar Plug-Ins verwenden (siehe Tabelle).
RealPlayer benützt sogenanntes Streaming um eine Präsentation zu zeigen. D.h. der

Benützer muss nicht warten, bis das ganze File heruntergeladen ist, der Player beginnt wenn er genügend in den Zwischenspeicher geladen hat (Buffer), und lädt weiter, während beim Benutzer die Präsentation bereits läuft.
RealPlayer ist momentan der meist benutzte und populärste SMIL Player auf der Welt.

4.1.2 SOJA

SOJA steht für **S**MIL **O**utput in **J**ava **A**pplet. SOJA wurde von HELIO 1998 entwickelt. HELIO ist eine französische Firma mit Sitz in Melun. Der Player ist gratis und kann von deren Homepage heruntergeladen werden.
SOJA ist ein Applet, das ein SMIL Dokument auf einer Web Page oder in einem separaten Fenster öffnet. Leider unterstützt SOJA nicht so viele Formate wie andere Player (Siehe Tabelle). Im Gegensatz zum RealPlayer funktioniert SOJA nicht mit Streaming. Das File muss zuerst vollständig heruntergeladen werden, bevor man es abspielen kann. Die Präsentation wird somit auch ohne Unterbrechung gezeigt.

4.1.3 GRiNS

GRiNS steht für **Gr**aphical **In**terface for **S**MIL. Dieser Player wurde von CWI, einer holländischen Firma entwickelt und herausgegeben. Auch CWI arbeitete an der Entwicklung von SMIL mit. Wie der SOJA Player, unterstützt GRiNS nur relativ wenige Formate (siehe Tabelle). GRiNS stimmt am ehesten mit dem SMIL Standart überein, er wird sich jedoch höchstwahrscheinlich nie gegen den RealPlayer durchsetzen.

4.1.4 andere Player

Es gibt noch eine Anzahl andere SMIL Player, so z.B. QuickTime, Shockwave, ... auf die hier aber nicht eingegangen wird.

4.2 HPAS

HPAS steht für **H**ypermedia **P**resentation and **A**uthoring **S**ystem.
HPAS ist ein System um zeitbasierende Hypermedia Dokumente zu präsentieren, integrieren und managen. Mit HPAS können Web Benutzer Text Files, Bilder, Audio Dateien und Videofilme zu einer Präsentation zusammenfügen. Das Erstellen eines Hypermedia Dokumentes bringt verschiedene Probleme mit sich, bezüglich der Dynamik von Multimedia allgemein und von zeitlich variierenden Umgebungen speziell. Das HPAS Projekt beschäftigte sich hauptsächlich mit den Problemen, die durch die zeitliche Dynamik entstanden. Es gibt zwei verschiedene Formate um ein HPAS Dokument zu erstellen: mit HSL (Hypermedia Synchronization Language) und mit SMIL. Dazu kommen zwei verschiedene Implementationsarten. Man kann ein HPAS Dokument als Java Applet benutzen, jedoch läuft es nur mit Windows und kann HSL und SMIL nur präsentieren. Die zweite Möglichkeit ist das Dokument als C/C++ Applikation zu öffnen. Dieses geht wiederum nur mit Unix. Dazu kommt noch, dass man nur HSL präsentieren und bearbeiten kann, nicht jedoch SMIL. HPAS unterstützt einige verschiedene Formate, wie in untenstehenden Tabelle gezeigt ist.

4.3 Vergleich : RealPlayer – SOJA – GRiNS – HPAS

	Tag	G2	SOJA	GRiNS	HPAS
GIF	Img	OK	OK	OK	OK
JPEG	Img	OK	OK	OK	OK
Microsoft Wav	Audio	OK	-	OK	OK
Sun Audio	Audio	OK	OK	OK	OK
Sun Audio Zipped	Audio	-	OK	-	-
MP3	Audio	OK	-	-	-
Plain text	Text	OK	OK	OK	OK
Real Text	Textstream	OK	-	-	-
Real Movie	Video	OK	-	-	OK
AVI	Video	OK	-	OK	OK
MPEG	Video	OK	-	OK	OK
MOV	Video	OK	-	-	-

5. SMIL 2.0

5.1 Allgemein

SMIL Boston, SMIL2.0 oder SMIL 20 wie es auch genannt wird, wurde erst kürzlich, d.h. im September 2000, als eigenständige Version von SMIL angesehen und vom W3C veröffentlicht. Wie auch SMIL 1.0 hat SMIL20 das Ziel, eine Programmiersprache zu definieren, mit der man multimediale Elemente in Web Präsentationen einbinden und gegenseitig synchronisieren kann und die sich benutzerdefiniert darstellen lässt. Die Syntax und Semantik von SMIL20 soll auch in anderen auf XML basierenden Sprachen eingesetzt werden, in denen Timing und Synchronisation eine wichtige Funktion erfüllen. So werden z.B. schon SMIL20 Komponenten benutzt, um Zeiteinstellungen auf XHTML vorzunehmen. XHTML steht für E**x**tensible **H**yper**t**ext **M**arkup **L**anguage. Im Vergleich zu SMIL 1.0 benützt SMIL20 eine leicht verschiedene Syntax, die benutzerfreundlicher sein sollte. Der Hauptunterschied besteht darin, dass doppelnamige Attribute neu aneinander geschrieben werden, anstatt mit einem Bindestrich getrennt werden. So wird z.B. clip-begin neu clipBegin geschrieben.
Leider funktionieren die alten SMIL Player nicht mit SMIL20, da SMIL 1.0 diese Änderung nicht erkennt und unterstützt. Die neuen SMIL Anwendungen müssen jedoch sowohl mit SMIL 1.0 als auch mit SMIL20 arbeiten, was zusätzliche Implementationen notwendig macht.

5.2 SMIL20 Player

Bis jetzt gibt es erst einen SMIL20 Player und zwar von GRiNS. Dieser wird von ORATRiX, einer weiteren holländischen Firma, mit einem Lizenz-Kit offeriert, um nach ihren Angaben SMIL20 populärer und verständlicher zu machen. Die erste Auslieferung erfolgte anfangs Januar 2001. Die Kosten für diesen Kit betragen 12500 Dollars ! Es gibt auch einen sogenannten SMIL20 Evaluation Player, der nicht der neuesten Version von SMIL20 entspricht, aber gratis von deren Homepage heruntergeladen werden kann. Jedoch muss man nach 7 Tagen das Package bestellen, wenn man weiter mit SMIL20 arbeiten möchte, da diese Version zeitlich limitiert ist.

6. Schlussfolgerung und Blick in die Zukunft

SMIL ist ein wichtiger Baustein des Internets und wird es zweifelsohne bleiben. Um so mehr, da SMIL einfach zu lernen und anzuwenden ist. In dieser relativ kurzen Zeit des Internets bedeutete SMIL einen weiteren grossen Fortschritt und ist bereits nicht mehr wegzudenken.
Auch da die Web Pages je länger je mehr Multimedia Anwendungen beinhalten. In der heutigen Zeit gilt sowieso: Je animierter die Page, desto länger bleibt der Surfer/User aud der Site (je multimedialer desto attraktiver). Deshalb ist es auch fürs E-Business immer wichtiger, die Sites mit Multimedia, d.h. Musik, Videos, animierten Bildern, usw. auszustatten. Diese Tendenz wird noch stärker zunehmen, da das Internet immer populärer wird und immer mehr Personen Zugang zum Internet erhalten. Und Da SMIL eine einfache Programmiersprache ist, werden auch immer mehr Personen diese Sprache aktiv einsetzen.

7. Anhang : Referenztabelle

<smil>	Muss am Anfang aller SMIL Dokumente stehen Kinder: head, body
<head>	Alle Layout Elemente sind hier definiert. Dies ist der 1te Hauptteil Kinder: meta, layout, switch
<meta>	Enthält zusätzliche Infos über die aktuelle Präsentation
<layout>	Das Layout des Bildschirms wird hier eingegeben Kinder: root-layout, region
<root-layout>	Braucht man um genaue Angaben bez. des Bildschirms zu machen
<region>	Definiert ein Rechteck im Hauptschirm
<body>	Die Tags in diesem Abschnitt werden nacheinander ausgeführt. Dies ist der 2te Hauptteil und wird fürs synchronisieren und die Medien genutzt Kinder: switch, media tag, a (Hyperlink), par, seq
<seq>	Die Tags nach <seq> werden nacheinander ausgeführt Kinder: switch, media tag, a, par, seq
<par>	Die Tags nach <par> werden zeitgleich ausgeführt Kinder: switch, media tag, a, par, seq
<switch>	Kinder werden zuerst analysiert und dann wird das Kind ausgeführt, welches TRUE zurückgibt, d.h. bei welchem alle Systemattribute übereinstimmen Kinder: (head) layout, (body) switch, media tag, a, par, seq
<a>	Erfüllt Link-Funktion. Man gibt eine URL an, wohin das Programm gehen soll, falls das Kind ausgeführt wird Kinder: switch, media tag, par, seq
<anchor>	Gleich wie <a>, ist aber nur mit einem Media Tag verbunden und besitzt Zeiteinstellungen. <anchor> ist Kind eines Media Tags
<img> und <text>	Media Tags um bestimmte Medien in ein Fenster einzufügen Kind: anchor
<video> und <audio>	Media Tag mit Clip-Eigenschaften Kind: anchor

Alle Tags müssen in ihrer Ebene geschlossen werden !

BIBLIOGRAPHISCHE ANGABEN:

- L.Ruteledge: SMIL; iX; Heft 10; 1999; Seiten 58-63
- W3C Consortium: SMIL 1.0 Specification; 15.06.1998
- W3C Consortium: Synchronized Multimedia; 13.12.2000
- W3C Consortium: SMIL 2.0 Specification; 2000
- W3C Consortium: Testimonials; 15.06.1998
- Internet.com: W3C Issues SMIL as proposed Recommendation; 15.06.1998
- Helio.org: SMIL Tutorial; 1999
- Oratrix Developpement BV: GRiNS/SMIL20 Player; 2000

Elektronische Post im Internet

Vortrag 9

Thomas Zaugg

16. Januar 2001

PPS Seminar: Grundlagen der Internet-Technologie

1. Einführung

Mit der Entwicklung des Internets wurde nicht nur das Beschaffen von Informationen, sondern auch die Kommunikation wesentlich vereinfacht. Für die Geschäftswelt wie auch für die Privatsphäre ist das Internet als Kommunikationsmittel kaum mehr wegzudenken. Das Versenden elektronischer Post (E-Mail) ist sehr einfach und deshalb bei vielen Internetbenutzern sehr beliebt.
Lange vor der Web-Euphorie war E-Mail zweifellos die wichtigste Netzwerkanwendung. Weil in wissenschaftlichen Kreisen Computer mit Netzwerkzugang zuerst vorhanden waren, fand E-Mail vor allem dort erste Anwendung. Heute ist ein Netzwerkzugang kein Luxus mehr. Deshalb hat eigentlich jeder die Möglichkeit, per E-Mail zu kommunizieren. Infolge von E-Mail hat nicht nur die Bedeutung der konventionellen Briefpost (Snail Mail = Schneckenpost) stark abgenommen, Telefon und Fax werden im Vergleich zur elektronischen Kommunikation auch weniger benötigt.
Aufgrund der grossen Bedeutung der elektronischen Post kommt man heute sehr einfach an eine eigene E-Mail Adresse. Die meisten Internet-Anbieter (Provider) stellen kostenlos eine oder manchmal auch mehrere E-Mail Adressen zur Verfügung. Diese E-Mail Accounts sind dann normalerweise stationär. E-Mails können also nur von der beim jeweiligen Provider angemeldeten Workstation abgerufen werden. Man kann sich aber auch ein E-Mail Account reservieren, auf das man von überall aus Zugriff hat (z.B. Hotmail). Die grössere Flexibilität hat man aber mit Speicherlimitierung (z.B. maximal 2 Megabyte) und langen Wartezeiten zu bezahlen.
Doch was bietet E-Mail eigentlich für Vorteile gegenüber anderen Kommunikationsmitteln? Naheliegend ist, dass das Versenden von E-Mails weitgehend kostenlos ist. Der wesentliche Punkt ist aber sicherlich, dass eine Nachricht in Kürze den entlegensten Winkel der Welt erreichen kann. Das fördert die allgemeine Flexibilität, weil Informationen aller Art schneller ausgetauscht werden können. Hinzu kommt, dass die Adressierung oft viel einfacher ist als beispielsweise bei konventioneller Briefpost. Auch klar als Vorteil zu nennen ist, dass man seine Post dank nicht stationären E-Mail Accounts (z.B. Hotmail,...) von einer beliebigen Workstation betrachten kann. Jeder Computer mit Internet Zugang wird somit zum Briefkasten.

2. Aufbau und Codierung einer E-Mail

Gut dokumentierte und seit Jahren unveränderte Standards haben die explosionsartige Verbreitung von E-Mail unterstützt. Mailprogramme können weltweit auf unterschiedlichsten Rechnern und Betriebssystemen miteinander kommunizieren. Das Versenden von E-Mails ist ein Kinderspiel. Doch es kommt auch immer wieder vor, dass Umlaute entstellt oder Dateien verstümmelt werden. Es lohnt sich also, genauer auf den Aufbau einer E-Mail zu achten, um solche Phänomene zu begreifen.

2.1 Aufbau

Eine Mail besteht aus einem Header und einem Body. Im Header sind Informationen über Absender und Empfänger gespeichert. Die wichtigsten sind „From:“ für Absender, „To:“ für Empfänger und „Date:“ für das Absendedatum.

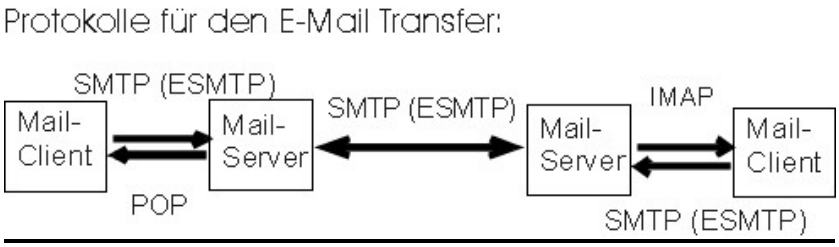
„Cc:“ (Carbon Copy) gibt an, wer das E-Mail noch zusätzlich erhält, sozusagen als Durchschlag. Mit „Bcc:“ (Blind Carbon Copy) können zusätzliche Kopien an andere Empfänger verheimlicht werden. Oft enthalten Mails noch den Header „Subject:“, hinter dem sich der Betreff verbirgt.
Nach einer Leerzeile folgt der Body der E-Mail. Im Body ist die eigentliche Nachricht enthalten.

2.2 Codierung

Mit dem Kernstandard RFC 822, auf dem das heutige Mailsystem beruht, konnte man früher E-Mails nur als reine Textdateien versenden. Genauer waren im Body nur ASCII-Code Zeichen zulässig. ASCII-Code ist der klassische US-amerikanische Standardcode, der mit 7 Bits (0-127) aufgebaut ist. Als man aber zunehmend das Bedürfnis verspürte, auch kompliziertere Zeichen sowie Bilder, Audiofiles und Filmsequenzen zu versenden, stand man vor einem Problem. Bilder und Audiofiles können beliebige Bytes (8 bit) zwischen 0 und 255 enthalten. So musste man damals diesen 8 bit Datenstrom zuerst in eine Folge von 7 bit ASCII Zeichen umwandeln, bevor man die entsprechende Datei versenden konnte. Codiert wurde mit Programmen wie uuencode oder xxencode. Um die Originaldatei zu erhalten, benötigte man uudecode oder xxdecode.
Seit 1996 gibt es den Standard MIME (Multipurpose Internet Mail Extensions) mit dem das Versenden von Umlauten und Dateien als „Attachments“ kein Problem mehr darstellt. Dieser Standard strukturiert den Body der E-Mail. Er unterteilt den Body und gibt vor jedem Abschnitt an, um welche Art von Information es sich handelt und wie sie codiert ist. Abschnitte können auch in weitere Unterabschnitte aufgeteilt werden.
Wenn man eine Mail mit einem MIME-fähigen Editor schreibt und sie nachher mit einem alten nicht MIME-fähigen Mailer betrachtet, so kann man diese Strukturierung gut erkennen. Wenn zum Beispiel im MIME Header „multipart/alternative“ steht, so heisst das, dass verschiedene Teile der gleichen Information in unterschiedlichen Darstellungsvarianten folgen. Der Client auf der Empfängerseite wählt dann denjenigen aus, den er am besten darstellen kann.
Nach dem MIME-Header „text/plain“ folgt gewöhnlicher Text mit Umlauten. Die Transferkodierung ist 8 bit. Diese Kodierung setzt voraus, dass das 8. Bit nicht verloren geht. Die MIME-Spezifikation lässt das zu, weil heute praktisch alle Mail-Verbindungen 8 bit transparent sind.
Eine sicherere Kodierung ist „quotet-printable“, welche Zeichen, die nicht in die 7 Bits passen, in ein Gleichheitszeichen und ihren Wert in hexadezimaler Darstellung umwandelt. Ein so kodierter Text würde dann auch mit einem nicht MIME-fähigen Mailer noch lesbar sein. Das Wort „wäre“ würde dann jedoch wie folgt aussehen: w=E4re.
Eine weitere mögliche Kodierung wird durch „text/html“ angekündigt. Sie besteht aus Text im HTML-Format, indem nur 7 bit ASCII-Zeichen vorkommen. Der Umlaut ä würde dann ohne MIME-fähiges Programm ungefähr so aussehen: ä und ö würde zu ö.
Nach dem MIME-Header „image/gif“ folgt ein Bild im GIF-Format. Die Transferkodierung heisst „base64“. Diese Kodierung packt jeweils drei Bytes à 8 bit zu vier Zeichen mit jeweils 6 bit (ASCII Code Zeichen 0-64). Die Datenmenge wird dadurch vergrössert, dafür beschränkt sich diese Kodierung auf Zeichen, die den Transportweg garantiert überstehen.

3. Senden und Empfangen einer E-Mail

E-Mail kann man als eine Reihe von Protokollen betrachten, die zum Austausch elektronischer Nachichten eingesetzt werden können. Man unterscheidet zwei verschiedene Bereiche. Der erste Bereich ist das Senden einer E-Mail an eine beliebige Empfängeradresse. Der zweite Bereich spezifiziert den Empfang einer E-Mail Nachricht durch den Benutzer.



3.1 Senden von E-Mail

Das Internetprotokoll, das das Senden von E-Mails beschreibt, ist das Simple Mail Transfer Protocol (SMTP). Das SMTP operiert relativ einfach. Der Absender einer E-Mail, resp. ein E-Mail Programm, öffnet zuerst eine Verbindung zum Empfänger. Beim Empfänger kann es sich dabei um den direkten Mail-Host des Adressanten oder um ein Übertragungssystem handeln, welches dann die E-Mail zum direkten Mail-Host des Adressanten weiterleitet.
Es hat sich jedoch herausgestellt, dass das SMTP in einigen Anwendungen den Anforderungen nicht genügte. Die deshalb entwickelte Erweiterung zu SMTP ist das Extendet Simple Mail Transfer Protocol (ESMTP). Die wichtigsten Erweiterungen des ESMTP lauten wie folgt:

- Nachrichten in 8-Bit Zeichensätzen: Weil das SMTP nur ASCII-Zeichen erkennt, ist es unsicher. Im ESMTP ist jedoch definiert, wie auch Nicht-ASCII-Zeichen ohne Probleme übertragen werden können
- Ankündigung der Nachrichtengrösse: Durch das ESTMP wird dem Empfänger einer E-Mail mitgeteilt, wie gross die Nachricht ist. Dieser kann dann entscheiden, ob er die Nachricht in angekündigter Grösse akzeptiert.
- Umfangreiche und binäre Nachrichten: Oft ist es sinnvoll, bei sehr umfangreichen E-Mails, die Nachricht in mehrere Teile zu zerlegen. Die Steuerung dieses Prozesses wird auch im ESMTP definiert.

3.2 Empfangen einer E-Mail

Am Anfang wurden E-Mails immer auf dem direkten Mail-Host des Empfängers gelesen. Der direkte Mail-Host ist die Maschine, auf welcher die E-Mail Nachricht via SMTP empfangen wurde. Das Lesen von E-Mails entsprach dem Lesen einer Datei, in die das SMTP-Programm alle ankommenden Nachrichten schrieb. Das funktioniert folglich nur, wenn Benutzer und SMTP-Programm die gleichen Dateien benutzen.
Bei der neueren Methode zum Empfangen von E-Mails werden Nachrichten nicht auf dem direkten Mail-Host des Empfängers gelesen. Die Nachrichten werden auf einem anderen Mail-Host (oft auch als Mail-Server bezeichnet) gespeichert. Der Empfänger kann auf diesen Mail-Server zugreifen und so die Mail empfangen. Das bietet den Vorteil, dass kein gemeinsames Dateisystem benötigt wird, da der

Zugriff auf den Mail-Server über eine Netzwerkverbindung erfolgt. Drei verschiedene Zugriffsarten sind möglich:

- Offline: Der Benutzer verbindet sich periodisch mit dem Mail-Server und holt neue Nachrichten ab. Die Nachrichten können dann auf dem Server gelöscht werden.
- Online: Im Online Betrieb wird die Nachricht vom Benutzer auf dem Mail-Server aus Entfernung bearbeitet.
- Getrennt: Die E-Mail wird vom Server zum Benutzer kopiert, nicht aber auf dem Server gelöscht.

Die wesentlichen Nachrichtenzugriffsprotokolle sind das Post Office Protocol (POP) und das funktionellere Internet Message Access Protocol (IMAP).

Post Office Protokol (POP)

Nachdem eine Nachricht gesendet worden ist, wird sie auf dem Mail-Server des Empfängers gespeichert. Auf dem Mail-Server ist Platz für den Empfänger reserviert. Das entspricht sozusagen einem Postfach. Oft nimmt der Empfänger mit seinem Mail-Server nur gelegentlich Kontakt auf, um seine Nachrichten abzufragen. Das entspricht ungefähr einem Gang zur Post (deshalb Post Office Protocol).
Die Funktionsweise von POP ist recht einfach. Der Zugriff auf die Nachrichten im Mail-Server kann erfolgen, sobald der Benutzer eine TCP-Verbindung zu einem bestimmten Port des Mail-Servers (POP-Servers) aufgebaut hat. Die Authentifizierung erfolgt über die POP-Befehle „user“ und „pass“, ist aber relativ unsicher, weil so Benutzername und Passwort unverschlüsselt über die TCP-Verbindung übertragen werden. Nach der erfolgreichen Autorisierung wechselt die POP-Verbindung in den Transaktionsstatus. Dieser Status erlaubt dem Benutzer nun freien Zugriff auf sein Postfach. Der Nachteil von POP ist, dass POP nur den oben erwähnten Offline-Betrieb unterstützt.

Internet Message Access Protocol (IMAP)

Das neuere Internet Message Acces Protocol (IMAP) ist eine leistungsfähigere Erweiterung von POP. Es hat im Vergleich zu POP folgende Vorteile aufzuweisen:

- Unterstützung verschiedener Ordner: Auf dem Mail-Server (IMAP-Server) können neben dem Ordner für eingehende Nachrichten beliebig weitere Ordner erstellt werden.
- Ordnerbearbeitung über das Netzwerk: Nachrichten können von Ordner zu Ordner verschoben werden.
- Optimierte Online-Performance: IMAP erlaubt das Abholen einzelner Teilstücke von komplizierten MIME-Nachrichten (Bilder,Videos,...)

IMAP hat jedoch auch zwei erwähnenswerte Nachteile. Das Protokoll ist zwar im Vergleich zu POP funktionell überlegen, ist aber entsprechend schwerer zu implementieren. Der zweite Nachteil ist, dass IMAP zur Zeit weniger unterstützt wird als POP.

4. Zusammenfassung und Schlussfolgerung

Mit dem Aufkommen vom Internet hat auch die Bedeutung von E-Mail enorm zugenommen. E-Mail ist daran, konventionelle Kommunikationsmittel zu verdrängen. Verantwortlich für die einfache Nutzung von E-Mail sind die vielen einfachen Internetstandards und Protokolle, die das Versenden von Nachrichten definieren resp. unterstützen. Die Kodierung einer E-Mail wird heute durch den Standard MIME geregelt. Früher musste man Sonderzeichen, die nicht zum ASCII-Code gehörten, sowie Bilder und Audio Files in ASCII-Code umwandeln, weil Mailverbindungen nicht 8 Bit transparent waren. Das Versenden und Empfangen von E-Mails basiert auf dem SMTP (ESMTP) und POP (IMAP). Diese Protokolle regeln den Transportweg einer E-Mail vom Sender über Mail-Server zum Empfänger.

Klar ist, dass heutzutage wohl niemand mehr auf E-Mail verzichten möchte und dass E-Mail wesentliche Vorteile gegenüber anderen Kommunikationsmitteln aufweist. Doch hinter dieser neuen Kommunikationstechnik sind auch Gefahren verborgen, die mehr und mehr für Gesprächsstoff sorgen werden. Das meistgenannte Problem in Bezug auf E-Mail Versand ist die Sicherheit. Die in diesem Vortrag erwähnten Standards sind alles andere als sicher, weil die Nachrichten auf ihrem Weg durchs Internet uncodiert sind. So kann sich der geübte Fachmann ohne weiteres Zugriff auf fremde, vertrauenswürdige Daten verschaffen. Lösungen werden jedoch inzwischen bereits durch neue Protokolle wie etwa das S/MIME-Protokoll (Secure Multipurpose Internet Mail Extensions) oder OpenPGP(Pretty Good Privacy) angeboten. Diese Protokolle definieren die Verschlüsselungen und digitalen Signaturen für E-Mail. (Genaueres zu diesem Thema folgt im Vortrag Nr. 10 Sichere Kommunikation von Luigi Scoca.)
Eine andere Gefahr, die noch zu erwähnen ist, beruht nicht auf technischer Natur. Die Kommunikation über das Internet ist nicht nur sehr unpersönlich, sondern kann dazu führen, dass soziale Kompetenzen weniger stark geschult werden. Tatsache ist, dass man durch konventionelle Kommunikationsmittel wie zum Beispiel Briefpost oder Telefon mehr im direkteren und verbalen Kontakt mit Menschen ist.

Bibliographische Angaben

[1] E. Wilde: *World Wide Web –Technische Grundlagen;* Springer Verlag, Berlin, Deutschland, 1999, Seiten 497 - 505.
[2] H. Bögelholz: *Wirbelwind – Elektronische Post im Internet;* c`t, Heft 8, 1999, Seiten 152 -154.
[3] G. Spiegel: *Gesicherter Umschlagplatz;* c`t, Heft 26, 1999, Seiten 160 -169.

Sichere Kommunikation –SSL und SHTTP

Luigi Scoca
23. Januar 2001

PPS Seminar: Grundlagen der Internet-Technologien

1.Einführung (Sicherheitsrisiken)

Wie sicher ist die Kommunikation (Datenaustausch) via Internet?
Die herkömmlich auf dem Internet basierenden Dienste wie z.B. E-Mail weisen zwei schwerwiegende Sicherheitsrisiken auf, die man leicht ausbeuten kann. Die Kommunikation über das Netz erfolgt nämlich im Klartext. Das heisst, die Daten werden nicht verschlüsselt. Mit einem geeigneten Programm können nun die Daten aufgezeichnet werden und die Bits anschliessend ausgewertet werden – man nennt dies *„Abhören“*. Ausserdem werden Benutzername und Passwort beim Abholen der E-Mails vom Server mittels POP3 unverschlüsselt übertragen. Diese Daten könnten von einer Drittperson abgefangen werden; diese kann sich nun durch eine falsche Identität ins Netz einloggen. Das zweite Risiko ist die fehlende Authentizität und Integrität der Datenpakete. Das bedeutet das man nicht sagen kann, ob die versandten Daten während des Transports abgeändert oder verfälscht wurden. Angreifer können somit leicht Datenpakete unter falscher Identität an andere Computer senden beziehungsweise übertragene Daten manipulieren. Einen solchen Angriff bezeichnet man als *„man in the middle attack“*.
Im Normalfall ist es nicht möglich, ein Netzwerk physisch vor Angreifern zu schützen. Doch was unternimmt man, um das Netz sicherer zu machen? Es existieren zwei Wege zur Sicherung der Kommunikation zwischen Client und Server. Je nach dem ob die Sicherheitsmassnahmen innerhalb der Transportinfrastruktur oder innerhalb der Anwendung implementiert werden sollen. Man unterscheidet folgende zwei Sicherheitsmassnahmen:

1.2 Verwenden einer sicheren Transportinfrastruktur

Geht man von einer sicheren Transportinfrastruktur aus, muss das Anwendungsprotokoll nicht verändert werden. Die Aufgabe besteht darin, eine sichere Verbindung herzustellen. Ein Vorteil dieser Variante ist, dass jede Anwendung diese Sicherheitsprotokolle benutzen kann, sofern sie die Transportinfrastruktur kennen. Ein weiterer günstiger Punkt ist der modulare Aufbau.

1.3 Verwenden eines sicheren Protokolls

Im zweiten Fall wird davon ausgegangen, dass die Transportinfrastruktur unsicher ist, und aus diesem Grund das Anwendungsprotokoll abgeändert werden kann. Um dieses exakt definieren zu können, muss das Anwendungsprotokoll über Sicherheitsmerkmale verfügen. Diese Sicherheitsmerkmale sollen eine Manipulation der Daten verhindern. Die Anforderungen bezüglich der Transportinfrastruktur sind hier wesentlich tiefer. Die Transportinfrastruktur kann im Grunde genommen unverändert bleiben. Jedoch das Hinzufügen von Sicherheitsmerkmalen am Protokoll bzw der Anwendung ist sehr schwierig und umfangreich zu gestalten. Darüber hinaus lassen sich die Sicherheitsmerkmale lediglich für eine bestimmte Anwendung

verwenden. Es müssen also für jede neue Anwendung neue Sicherheitsmerkmale definiert werden, was sich als sehr aufwendig entpuppt.

2. SSL

Definition SSL:

Secure Socket Layer; von Netscape entwickelte Public-Key-basierte Absicherung von HTTP Kommunikationskanälen: Dabei sind Client- und Serverauthentifizierung möglich.

Bei dieser Lösung wird das normale HTTP über eine sicher Transportinfrastruktur eingesetzt – dem sogenannten SSL (Secure Socket Layer). Es wird also die Variante 1 (vgl. 1.1) gewählt. Diese zusätzliche Schicht wird sich zwischen der normalerweise verwendeten TCP/IP-Schicht und HTTP als Anwendungsprotokoll liegen. Das SSL-Protokoll ist ein geschichtetes Protokoll. In jeder Schicht sind Angaben über Länge und Inhalt enthalten. SSL nimmt die Daten und teilt sie in Blöcke geeigneter Grösse ein, damit sie versandt werden können. Zusätzlich werden die Daten verschlüsselt und komprimiert. Ist dies geschehen, können die Daten verschickt werden. Am Ziel angekommen, werden sie dann wieder entschlüsselt.
Wie erkennt man nun die Verwendung der SSL-Technik? Man erkennt die Verwendung der SSL-Tecnik an den Adressen der Webseiten, statt *„http://“* steht nun *„https://“*.
Welchen Anforderungen muss SSL genügen bzw. welche Merkmale und Eigenschaften, muss diese sogenannte Zwischenschicht haben?

- Verbindungssicherheit:
  Die Verbindungssicherheit wird durch die Verschlüsselung der Daten gewährleistet.

- Authentifizierung:
  Die Authentifizierung ist eine weitere Anforderung, welche man an SSL stellt. Das heisst, der Client muss sich gegenüber dem Server identifizieren können und umgekehrt. Damit wird vermieden, dass ein dazwischengeschalteter Rechner (vgl. 1.2) die Daten abfängt.

- Zuverlässigkeit der Verbindung:
  Die versandten Daten müssen auf ihre Integrität überprüft werden, damit Manipulationen ausgeschlossen werden können.

- Erweiterbarkeit:
  SSL versucht einen Rahmen bereit zu stellen, innerhalb dessen sich neue Verfahren zur Herstellung von öffentlichen Schlüsseln und Verschlüsselung grosser Datenmengen entwickelt werden können.

∂ Relative Wirksamkeit:
Da die krypthographische Verschlüsselung der Daten sehr rechenintensiv ist, bedeutet dies eine zusätzliche Belastung des Netzes, was zu Geschwindigkeitseinbussen führt. Deshalb ist ein weiteres Ziel, die Verschlüsselung mit einem geringen Rechenaufwand zu gestalten.

∂ Interoperabilität
Unabhängige Programmierer sollten SSL einsetzende Anwendungen entwickeln können, die in der Lage sind, Verschlüsselungsparameter auszutauschen, ohne jeweils den Source-Code des anderen zu kennen.

2.1 Herstellung der Verbindung
Die Verbindungsaufnahme zwischen Client und Server kann grundsätzlich auf drei Arten erfolgen. Wobei nicht alle drei gleich sicher sind.

∂ Anonymität:
Weder Client noch Server müssen sich identifizieren. Bei dieser Variante wird nur ein Schutz vor Abhören gewährleistet. Unter Umständen könnte der Server nicht der gewünschte Zielrechner sein sondern ein dazwischengeschalteter Rechner (vgl 1.2).

∂ Server Authentifizierung:
Wie der Titel schon verrät, muss sich nur der Server vor dem Client ausweisen können.

∂ Authentifizierung beider Parteien:
Die letzte Art ist die sicherste Möglichkeit, hier müssen sich beide Parteien ausweisen können – der Client beim Server und umgekehrt.

2.2Sichere Kommunikation
Nun schauen wir uns an wie die Herstellung einer sicheren Verbindung von statten geht. Der Ablauf einer typischen, sicheren Kommunikation zwischen Client und Server sieht folgendermassen aus:
∂ Verbindungsaufnahme des Clients
∂ Authentifizierung des Server
∂ Authentifizierung des Clients
∂ Datentransfer
∂ Verbindungsabbruch

Def: Asymmetrische Verschlüsselung

Bei der Verschlüsselung werden zwei Schlüssel generiert, ein privater und ein geheimer. Dabei kann nur ein passender, privater Schlüssel Daten entschlüsseln, die mit dem öffentliche Schlüssel codiert wurden und umgekehrt.

2.3 Installieren der Serverkomponenten

Beim Installieren der Serverkomponenten wird ein Schlüsselpaar generiert, ein sogenannter Hostkey. Er besteht aus einem öffentlichen Schlüssel (HK-Pub) und dem geheimen Schlüssel (HK-Sec). Diese beiden Schlüssel werden normalerweise nie mehr verändert.

2.4 Verbindungsaufnahme

Initiert nun der Client eine SSL-Sitzung, wird ein weiteres Schlüsselpaar erzeugt; bestehend aus einem öffentlichen Server-Key (SK-Pub) und einen geheimen Server-Key (SK-Sec). Diese Schlüssel werden in bestimmten Zeitintervallen geändert(circa einer Stunde) oder wenn ein Gigabyte an Daten gesandt wurde. Dieses zweite Schlüsselpaar soll verhindern, dass die Kommunikation aufgezeichnet und nachträglich entschlüsselt wird, selbst wenn sich der Angreifer im Besitze des Host-Keys befindet.

Nach dem Austausch der Protokollversionen sendet der Server seine öffentliche Schlüssel dem Client, dass heisst SK-Pub und HK-Pub. Der Client verifiziert, ob der HK-Pub in seiner Liste aufgeführt ist. Falls dies nicht der Fall ist, erscheint eine Warnmeldung – die Verbindung kann nun unterbrochen werden. Hat der Client den HK-Pub des Servers akzeptiert, erzeugt er seinerseits einen Sitzungsschlüssel (SESK). Dieser codiert er mit dem HK-Pub und dem SK-Pub und teilt dem Server welches Kryptoverfahren er anwendet. Den übermittelten Sitzungsschlüssel SESK kann der Server nun mittels HK-Sec und SK-Sec entziffern. Nun muss sich der Client identifizieren. Man unterscheidet drei Varianten der Identifizierung:

- Reine Hostbasierte Identifizierung:
  Sie erfolgt genau wie bei rlogin und RSH mittels Hosttabellen und ist extrem unsicher.

- Passwortauthentifizierung:
  Benutzername und Kennwort des Benutzers werden dem Server übermittelt. Dieser kann anhand einer lokalen Benutzerdatenbank überprüft werden. Die Daten gehen imKlartext über das Netz sind aber mit dem Sitzungsschlüssel chiffriert.

- Authentifizierung mittels Public-Key-Verfahren:
  Der Benutzer schickt mittel eines privaten Schlüssels eine Signatur an den Server. Dieser überprüft die Gültigkeit mit einem öffentlichen Schlüssel. Nun kann eine verschlüsselte Kommunikation zwischen Client und Server stattfinden.

3. S-HTTP

S-HTTP geht von einer unsicheren Transportstruktur aus (vgl 1.2), deshalb muss das Transportprotokoll mit Sicherheitsmerkmalen ausgestattet werden. Das S-HTTP Protokoll ist ein weiteres Protokoll was in die Anwendungsschicht eingefügt wird. Wenn sich der Client nun für eine sichere Datenübertragung entscheidet, wird das S-HTTP verwendet anstelle des normalen HTTP. S-HTTP definiert ein auf HTTP basierendes Nachrichtenformat. Vereinfacht ausgedrückt ist die Idee von S-HTTP die folgende: Man nehme ein schon bestehendes HTTP-Protokoll und verändere es so, dass eine Authentiefizierung zwischen Client und Server möglich ist, wie auch eine sichere Kapselung der Daten gewährleistet ist. Ausserdem müssen die S-HTTP-Protokolle kryptographische Standardformate von Nachrichten unterstützen, um den Datenkörper verschlüsseln zu können (z.B. S/MIME).

Def: Secure HTTP ist ein sicheres nachrichtenorientiertes Kommunikations-Protokoll, welches in Verbindung mit HTTP eingesetzt wird.

4.Zusammenfassung und Blick in Zukunft:

Zusammenfassend kann man sagen, dass SSL und S-HTTP das gleiche Ziel verfolgen: Eine sichere Kommunikation. Der Unterschied zwischen den beiden besteht in der Umsetzung dieser Aufgabe. Vereinfacht ausgedrückt wird mit SSL eine sichere Transportinfrastruktur kreiert, hingegen mit S-HTTP ein sicheres Kommunikationsprotokoll zur Verfügung gestellt.
In Zukunft wird die Nachfrage an sicherer Kommunikation kontinuierlich anwachsen. Dieses wird auch bestätigt durch die neuaufkommenden Firmen, welche sich in diesem Sektor beschäftigen und Techniken einen sichere Kommunikation anbieten. Fazit SSL und S-HTTP werden kontinuierlich modifiziert, ausgebaut und weiterentwickelt werden. Eine Alternative wäre etwas ganz neues zu entwickeln, was natürlich einen immensen Aufwand bedeutet und natürlich viele innovative Ideen verlangt.

5. Bibliographische Angaben

[22] S:Leich: *Doppelt genäht*; iX, Heft 1, Seiten 146-149.
[23] E. Wilde: *World Wide Web – Technische Grundlagen*; Springer Verlag, Berlin, Deutschland, 1999, Seiten 127-135.
[24] M. Thorbrügge*: Lückenfüller;* c`t, Heft 16, Seiten 176-179
[25] A. Freier, P. Karlton, P.C. Kocher: *SSL Protocol Version 3.0*; Internet Draft, work in progress, November 18,1996

Mobilkommunikation - WAP[1]

Vortrag 11
30. Januar 2000
PPS Seminar: Grundlagen der Internet-Technologie
Alain Haja Randriamora

[1] WAP: Wireless Application Protocol

1. Einleitung

Handys[2] haben in den letzten Jahren enorm an Bedeutung gewonnen und ein Ende des Booms ist noch längst nicht in Sicht. Genauso wie das Handy heutzutage nicht mehr wegzudenken wäre, wäre auch das Internet nicht mehr wegzudenken. Was liegt also näher, als diese beiden Technologien zu kombinieren, sprich Internetseiten für das Handy zugänglich zu machen und diese auf ihrem Display darzustellen. Dies ist allerdings mit einigen technischen Problemen verbunden, dennoch die Idee wurde in Tat umgesetzt und zwar mit Hilfe des Wireless Application Protocol (WAP).

2. Das WAP-Forum

Motorola, Nokia, Ericsson und Phone.com haben gemeinsam den ersten Stein gelegt, indem sie im Dezember 1997 das WAP-Forum gegründet haben. Dies ist eine nicht gewinnorientierte Vereinigung, welcher jedes Unternehmen beitreten kann, wenn es dazu bereit ist, die beschlossenen Spezifikationen einzuhalten. Mittlerweile zählen schon über 200 Unternehmen dazu. Ziel war es, einen weltweiten Standard für den Datenverkehr in Mobilfunknetzen zu erarbeiten. Das WAP-Forum ist für die einheitliche Umsetzung von WAP in den verschiedenen Handys verantwortlich und definiert, wie die verschiedene Geräte miteinander kommunizieren. Zudem beachtet es moderne technische Fortschritte, sodass kein Hersteller die Überhand gewinnen kann. Alle 6 Monate werden die aktuellen offiziellen Spezifikationen vorgestellt.

3. Was ist WAP?

WAP ist ein Protokoll, das die Übertragung und Darstellung von Internetdokumenten auf Handys definiert. Es regelt die Kommunikation zwischen einem WAP-Browser, welcher in jedem WAP fähigen Gerät eingebaut ist, und dem Webserver. Zwischen dem Handy und dem Server steht noch ein Gateway, welcher die Ausgabe des Webservers in das WAP-Format umsetzt und die Daten ans Handy weiter sendet. WAP ist also wie SMS ebenfalls ein Protokoll, das die Übermittlung von Daten von und zum Handy definiert, der Unterschied besteht darin, dass bei SMS Daten immer nur in eine Richtung verschickt werden und die Datenpakete auf 160 Zeichen beschränkt sind. WAP hingegen lässt beidseitige Kommunikation und grössere Datenpakete zu. Die ersten WAP-Handys kamen Ende 1999 auf den Markt, der offizielle WAP-Start war aber erst im Februar 2000.

4. Welche Probleme bestehen?

Erste Versuche zeigten schnell, dass die Umsetzung von WAP nicht ohne weiteres möglich ist. Ein Handy bleibt eben ein Handy und somit bleiben auch die Nachteile gegenüber einem PC. Mit folgenden Problemen musste und muss man sich auseinandersetzen:

- Geringe Übertragungsrate (9600 bps)
- kleine Speicher, auf einem Handy können Seiten einer maximalen Grösse von ungefähr 1400 Bytes dargestellt werden.
- unkomfortable Eingabemöglichkeiten, da meist keine echte Tastatur erhalten.

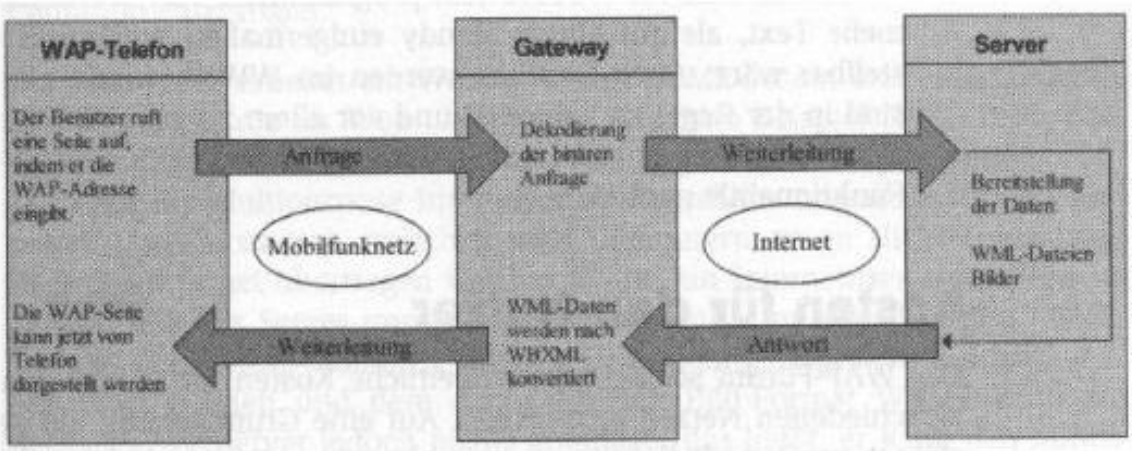
[2] In diesem Dokument ist stets nur von Handys die Rede, gemeint sind aber auch alle andern drahtlosen Mobilgeräte wie zum Beispiel PDAs (Personal Digital Assistent) etc.

- kleine Display mit geringer Auflösung. In der Regel keine Farbdisplays (weil Farbtiefe bloss ein Bit). Grafiken sollte man also nur sehr beschränkt einsetzen, nur auch schon wegen den kleinen Speichern.
- die grosse Anzahl verschiedener Geräte, nicht alle unterstützen alle Textformatierungen
- Geringere Stabilität der Verbindung mit Warte und Verzögerungszeiten ist also stets zu rechen.

5. Vom Handy zum Gateway zum Webserver und wieder zurück!

WAP-Seiten werden auf üblichen Internetservern gespeichert, da natürlich nicht jeder Server eine Verbindung zum Mobilfunknetz hat braucht es Vermittler, sogenannte Gateways, welche einerseits Verbindung über das Internet mit den Servern aufnehmen und andererseits für Mobilgeräte über eine Mobilnummer zu erreichen sind. In jedem WAP fähigen Gerät ist ein Modem eingebaut, das ein Gateway anwählen kann und mitteilt, welche WAP-Seiten der Nutzer gern sehen möchte. Das Gateway besorgt dann die Daten. Die Übertragungsgeschwindigkeit ist dabei sehr gering (9600 bps). Um Wartezeiten kurz zu halten, werden die grossen [3]WML-Dateien vor der Übertragung vom Gateway in ein Binärformat umgewandelt (WBXML: WAP Binary XML). Zudem werden Meta-Informationen und redundante Abschnitte bei der Konvertierung entfernt. Normalerweise wählt man als Benutzer direkt WAP-Seiten an, welche ja in WML programmiert sind. Man kann aber auch ganz normale Webseiten aufrufen, da Handys aber keine HTML-Dateien darstellen können, werden diese vom Gateway nach WML konvertiert, theoretisch kann man jede HTML-Seite nach WML konvertieren, praktisch treten aber unausweichliche Schwierigkeiten auf, nämlich:

- HTML ist nicht für stark eingeschränkte Displays vorgesehen, bei der Konvertierung werden daher viele Funktionen verloren gehen. Farben werden zum Beispiel vollständig ignoriert.
- Webseiten sind für Webbrowser programmiert und erhalten meist viel Text und Grafiken, welche auf einem Handy niemals zufriedenstellend darstellbar wären. Ein Verzicht auf Grafiken bedeutet oft auch, dass die Seite gar nicht mehr funktioniert oder zuminst nicht mehr so wie sie sollte. Meist sind HTML-Seiten schlicht viel zu gross fürs WAP!

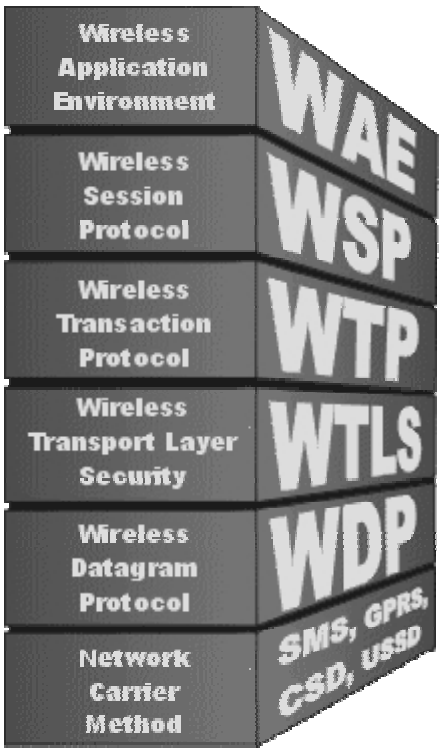


Gateways als Vermittler zwischen Handy und Server

[3] WML: Wireless Markup Language: siehe Kapitel 7

6. Das Schichtenmodell

Aber was passiert eigentlich genau, wenn ein Handy WAP-Angebote aus dem Netz lädt. Die WAP-Architektur basiert auf einem schichtenförmigen Modell, wie man es auch vom TCP/IP kennt. WAP wird in 5 Schichten beschrieben und in jeder Schicht kommen Anwendungen und Protokolle gleichermassen zum Einsatz.



Das WAP-Stack stellt die Protokolle dar, die den gesamten Prozess der schnurlosen Übertragung überdecken.

Anwendungschicht

Hier findet man das Wireless Application Environment (WAE), das als Anwendungsumgebung auf WWW- und Telefonietechnologien basiert und als Ausführungsumgebung von WAP-Anwendungen dient. Es unterstützt WML, WML-Script und WTA (Wireless Telephony Applications)

Sitzungschicht

Hier sorgt das WSP (Wireless Session Protocol) für die Bereitstellung von zwei Diensten. Zum einen um einen verbindungsorientierten Service (d.h. Daten werden zwischen Handy und Netzwerk hin und her geschickt) zum andern um einen verbindugslosen Service (Daten werden nur vom Netzwerk zum Handy geschickt).

Transaktionsschicht

Das WTP (Wireless Transaction Protocol) ist sozusagen der Verkehrspolizist. Es regelt die Transaktionen. Das WSP und WTP entsprechen dem HTTP im TCP/IP Protokoll

Sicherungsschicht

Sie dient zur Sicherung der Datenintegrität, Privatsphäre und Authentifizierung. Zudem bietet sie Schutz vor Denial-of-Service-Attacken.

Transportschicht

Das WDP (Wireless Datagram Protocol) ist für die Kommunikation zwischen dem Bearer (Schnittstellen zwischen WAP und physikalischen Netzen wie GSM oder TCP/IP) und den darüber liegenden Schichten zuständig.

Jede Schicht des WAP-Stacks hat eine genau definierte Schnittstelle zu der darüberliegenden Schicht. Diese Schichtarchitektur ermöglicht es Anbietern, Anwendungen und Dienste für die entsprechende Schicht anzubieten, indem sie die vom WAP-Stack unterstützen Funktionen nutzen. Man kann so sogar Dienste anbieten, die nicht durch das WAP spezifiziert sind. Die hierarchischen, unabhängigen Schichtenanordnung hat zudem den Vorteil, dass das System erweiterbar, flexibel und skalierbar ist.

7. WML, die Sprache für WAP-Seiten

Damit der Browser im Handy weiss, wie er die empfangenen Daten darstellen soll, wird für WAP eine eigene Sprache benötigt, nämlich WML, die von XML abgeleitet wurde, perfekt an WAP angepasst ist und der Sprache HTML sehr ähnelt. WAP hat auch ihr eigenes Bildformat: WBMP (Wireless Bitmap) bisher sind mit diesem Format jedoch nur schwarzweiss Bilder möglich. Beide Sprachen haben denselben Zweck, nämlich Inhalte in Dokumenten aufzubereiten, die über das Internet übertragen werden können. Beide Sprachen sind plattformunabhängig. Es gibt natürlich auch Unterschiede:

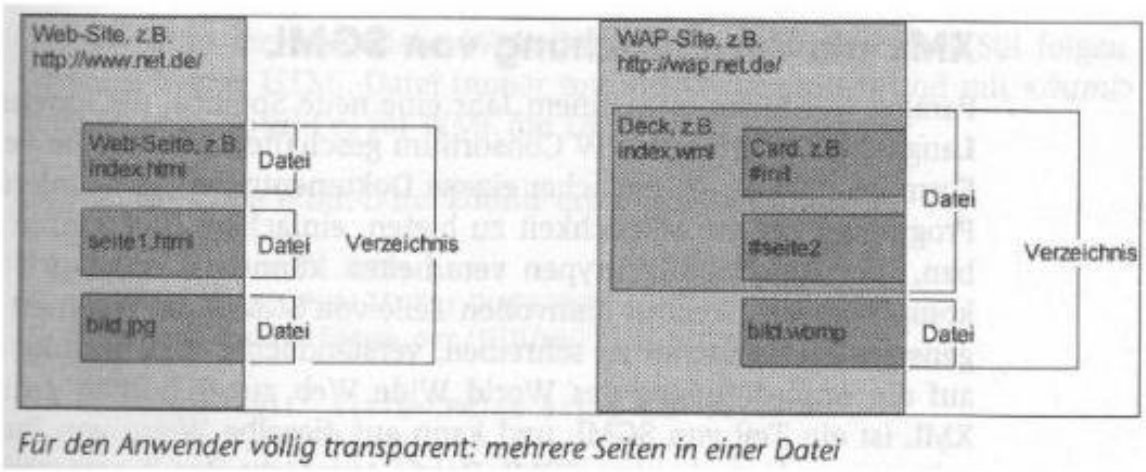
- WML verlangt eine wesentlich strengere Syntax als HTML. Der wohl wichtigste: Beim Programmieren in WML muss man auf Gross-/Kleinschreibung achten.
- In WML müssen alle Tags (Kommandos) mit einem Endtag abgeschlossen sein und klein geschrieben sein! <tag>…</endtag> In HTML gibt es zum Beispiel kein Endtag für den Zeilenumbruch (
)
- Alle Attributwerte müssen in Anführungszeichen eingeschlossen werden. Beispiel <img width = "200">
- Mit Sonderzeichen zum Beispiel den Umlauten ä ö ü muss man in WML speziell Acht geben, denn man kann nicht davon ausgehen, dass jedes Handy diese automatisch richtig interpretieren kann.

Wer HTML programmieren kann, der wird keine Probleme haben mit WML. Bloss die obigen Punkte müssen beachtet werden. Fast alle Befehle sind identisch, WML hat natürlich viel weniger Befehle, da die kleinen Handydisplays bedeutend weniger Spielraum zulassen. WML-Seiten sehen sehr schlicht aus, keine x-verschiedene Schriftarten und Schriftgrössen, bloss Formatierungsmittel wie fett und kursiv können angegeben werden. WAP Seiten sind schlank, schnell, textorientiert. Aufpassen muss man vor allem mit Bilder beim Programmieren von WAP-Seiten. Die allerwenigsten Handys haben ein Farbdisplay, ja längst nicht alle können überhaupt Grafiken darstellen, also darf ein Bild innerhalb einer WAP-Seite niemals eine

zentrale Rolle spielen und jede WAP-Seite sollte auch ohne Grafik ihren Zweck erfüllen.

8. Die Card-Struktur

Im WWW werden alle Webseiten die zu einem Projekt gehören, als sogenannte Site zusammengefasst. Jede Site besteht aus verschiedenen Seiten und jede Seite ist in einer eigenen Datei gespeichert.
Die WAP-Struktur ist etwas anders. Um lange Ladezeiten innerhalb einer Site zu vermeiden, werden mehrere Seiten in einer einzigen Datei gespeichert (Die Datei wird WML-Deck genannt) Ein Deck ist in mehrere Cards (eine Card entspricht einer Seite) aufgeteilt.



Für den Anwender völlig transparent: mehrere Seiten in einer Datei

9. Ein konkretes Beispiel

```
1. <?xml version="1.0"?>
2. <!DOCTYPE wml PUBLIC ".//WAPFORUM//DTD WML 1.1//EN"
   http://wwww.wapforum.org/DTD/wml_1.1.xml>
3. <wml>
4. <card id="start" title="WAP-Beispiel">
5. Das ist ein Test für die Darstellung von Texten auf WML
   Seiten.
6. </card>
7. </wml>
```

Erläuterungen:

1. Standartzeile die für alle XML-Dokumenten verlangt wird.
2. Das ist die XML Dokument Type Definition, welche benötigt wird, da externe Dokumenttypen verwendet werden.
3. Jedes WML-Deck wird durch den wml-Tag definiert und beendet (Zeile 7)
4. Jedes WML-Card wird durch den card-Tag definiert und beendet (Zeile 6). Dieses Demonstrationsbeispiel beinhaltet nur eine Card, man könnte natürlich dieses Deck mit weiteren Cards ergänzen.
5. Dies ist der Text, welcher schlussendlich auf dem Handy zu sehen ist und "beliebig" erweitert und formatiert werden kann.

10. Aussichten

Wie man sieht, Internet auf dem Handy wurde realisiert. Aber die hohen Erwartungen konnten bisher nicht erfüllt werden. Die Euphorie war Anfangs gross, doch der wahre Durchbruch blieb aus. SMS ist nach wie vor viel beliebter als WAP und kann fast genauso viel. Schlagzeilen wie: "WAP das gefloppte Internet für das Handy" hörte man oft. Auch an sonstigen Kritiken mangelt es nicht: Zu teuer, zu langsam, zu kompliziert, etc... Nun hofft man auf die Zukunft. In naher Zukunft auf das datenorientierte GPRS (Generalized Packet Radio Service). Theoretisch ist damit ein Übertragungsrate von 171'000 bps möglich, praktisch wird sich der Wert wohl zwischen 30'000 und 40'000 bps einpendeln, weil sich die Benutzer die Bandbreite teilen müssen. Man bezahlt aber nur die Menge der übertragenen Daten, die Zeitdauer der Verbindung spielt für die Kosten keine Rolle mehr - immerhin, so kann man ständig mit dem Netz verbunden bleiben.
Eine völlig neue Dimension wird bald UMTS (Universal Mobile Telecommunications System) eröffnen, mit welchem bis 2 Millionen bps erreicht werden können.
Auch die Schattenseiten zeichnen sich jetzt schon ab. Die Werbewirtschaft reibt sich schon die Hände, lästige Werbung auf dem Handy wird wohl bald immer mehr ein Thema werden und wo Daten übertragen werden, sind auch Viren nicht fern….

Literaturverzeichnis

Christian Immler, Markos Kreinacke, Andre Spallek: Das grosse Buch WAP: Düsseldorf, 2000, Data Becker

L. Röwenkamp: Handy HTML; iX, Heft 2, 2000, Seiten 52-57

Wireless Application Forum: WAP: Wireless Application Protocol; White Paper, October 1999

Folgende Internetseiten:

http://www.howstuffworks.com/index.htm
http://www.wapmag.de/index.html
http://handy-tabelle.de/wap/wap002.html
http://www.at-web.de/wap/wml.htm