Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Burkhard Stiller, Jan Gerke (Edt.)

PPS-Seminar: Grundlagen der Internet-Technologie 3

TIK-Report
Nr. 113, Juni 2001

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Burkhard Stiller, Jan Gerke (Edt.):
PPS-Seminar: Grundlagen der Internet-Technologie 3
Juni 2001
Version 1
TIK-Report Nr. 113

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 3

Einleitung

In der dritten Auflage dieses PPS-Seminares wurden ebenfalls wieder Studierende des Departements für Elektrotechnik, die die Grundlagen und erste wichtige Begriffe eines weitverbreiteten Netzwerkes – dem Internet – erlernen möchten 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, was ist Mobile IP? 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:	Raphael Finger:	Grundlagen des Internet
Vortrag 2:	Thomas Haag:	Netzwerktechnologien für das Internet
Vortrag 3:	Otto Huber:	IP, Adressierung und Routing im Internet
Vortrag 4:	Yannick Thebault:	IPng – Die nächste Generation des Internet Protokolls
Vortrag 5:	Matias Fernandez:	TCP/UDP
Vortrag 6:	Philippe Schaller:	Das HTTP-Protokoll des Web
Vortrag 7:	Nicolas Studhalter:	Die Beschreibungssprache HTML
Vortrag 8:	Thomas Mühlemann:	Die Datenstrukturierungssprache XML
Vortrag 9:	Andreas Meier:	Multimedia-Unterstützung im Web – SMIL
Vortrag 10:	Michael Casty:	Sichere Kommunikation – SSL, SHTTP
Vortrag 11:	Olivier Gillieron:	IP-Telefonie
Vortrag 12:	Martin Walder:	Elektronische Post im Internet
Vortrag 13:	Christian Morf:	Mobile IP
Vortrag 14:	Adrienne Heinrich:	Drahtlose Kommunikation – WAP

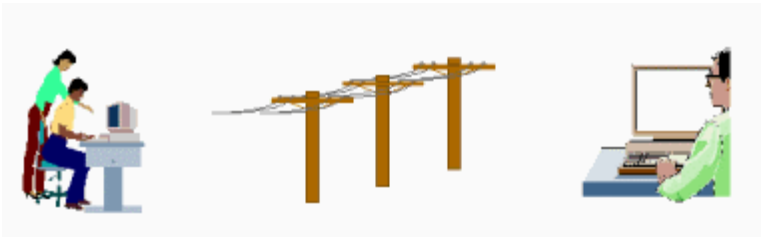
Grundlagen des Internet

Vortrag 1

Finger Raphael

30. April 2001

PPS-Seminar: Grundlagen der Internet-Technologie



1 Einführung

1.1 Internet – *die* Kommunikationsplattform der Zukunft

Im 1984 veröffentlichten Roman „Neuromancer“ verband der Science-Fiction-Autor W. Gibson das Gehirn von Hackern mit Computerzellen und beschrieb die Situation mit dem Begriff „Cyberspace“. Soweit ist es noch nicht, die Verbindung von Robotern oder Menschen mit speziellen „Datenhandschuhen“ oder „Cybersexanzügen“ über das Internet ist aber realisiert.

Den Begriff „Surfen im Internet oder Cyberspace“ kennt jedes Kind und auch für ältere Leute ist das Internet kein Fremdwort mehr. Kurzum, das Internet gehört zu unserem Alltag wie Zeitung, Radio und Fernsehen. Entsprechend ist auch seine wirtschaftliche und politische Bedeutung. Alle sind heutzutage „online“, das heisst mit dem Internet verbunden und können so orts- und zeitunabhängig beliebige Inhalte oder Informationen austauschen. Diese Raum-, Inhalts- und Zeitsouveränität unterscheidet das Internet von herkömmlichen Kommunikationsplattformen wie die Telefonie oder die Briefpost.

1.2 Was ist eigentlich das Internet ?

Das Wort „Internet“ setzt sich aus zwei Teilen zusammen, nämlich aus „inter“ (lateinisch für „zwischen“) und „net“, der Abkürzung für „networking“ (englisch für „vernetzen“). Im Computerbereich bedeutet „Internet“ also die Vernetzung zwischen Computernetzen. Das Internet ist demnach ein Computernetzwerk.

Das Internet darf nicht mit dem World Wide Web (WWW) gleichgesetzt werden. Es ist statt dessen ein Oberbegriff für viele einzelne Funktionen (Dienste). Eine ist das WWW, das vor allem der passiven Informationsabfrage dient. Ein weiterer beliebter Dienst des Internets ist das Postsystem „e-mail“. Der Benutzer schreibt seinen Brief dabei nicht mit Papier und Bleistift, sondern mit der Tastatur. Kein Briefträger übermittelt den Brief, sondern die Leitungen des Internets. Doch gibt es auch andere Postsysteme. Wer zum Beispiel Post (also e-mail) zu einem Thema erhalten möchte, kann sich in „mailing-lists“ eintragen. Ähnlich funktioniert das „news-System“, auch genannt „usenet“. Wichtig sind zudem das „file transfer protocol“ (kurz: „ftp“), durch das auch Dateien übermittelt werden können sowie das „telecommunications network“ (kurz „telnet“), das es ermöglicht, Rechner remote (von fern) zu bedienen.

Nun drängt sich die Frage auf, wem das Internet gehört. Das Internet gehört niemandem und doch allen. Das Netz der Netze ist dezentral organisiert. Das Internet baut auf das Engagement professioneller Unternehmen, welche die grossen Teilnetze betreiben, sowie auch auf die vielen Tausende Informationsanbieter. Die Hauptarbeit des Internet lastet auf den Schultern vieler Einzelpersonen, die sich an Universitäten, Unternehmen oder in den eigenen vier Wänden befinden und das Internet mit Informationen versorgen, vielfach auch unentgeltlich.

2 Geschichte und Technologie

2.1 Entstehung des ARPA-Netzes

Mitte der sechziger Jahre, auf der Höhe des kalten Krieges, wünschte sich das US-Verteidigungsministerium ein Kommando- und Steuernetz, das einen Atomkrieg zu überleben imstande sein sollte. Die konventionellen, leitungsvermittelten Telefonnetze galten als zu verletzlich, da der Ausfall einer Leitung oder eines Vermittlers mit Sicherheit alle Gespräche, die über sie laufen würden, beenden und sogar das Netz trennen könnte. Um dieses Problem zu lösen, wandte sich das Ministerium an seinen Forschungsbereich ARPA (der später in DARPA umbenannt wurde), der *(Defense) Advanced Research Projects Agency.*

ARPA wurde als Reaktion auf den Start des Sputniks durch die Sowjetunion im Jahre 1957 mit der Aufgabe ins Leben gerufen, ausgefeilte Technologien zu entwickeln, die für das Militär von Nutzen sein würden. ARPA hatte weder Wissenschaftler noch Labors. Es bestand aus nichts als einem Büro und einem (nach Pentagon-Massstäben) kleinen Budget.

ARPA arbeitete eng mit Universitäten zusammen, welche in ihrem Auftrag die Forschung der neuartigen Paketvermittlung vorantrieben. Nach einigen Diskussionen mit verschiedenen Fachleuten entschied ARPA, dass das Netz für das US-Verteidigungsministerium ein paketvermitteltes, aus mehreren Teilnetzen und Hostrechnern bestehendes Netz sein sollte.

ARPA schrieb die Entwicklung des Teilnetzes aus. Nach der Auswertung und zahlreichen Besprechungen mit Netzforschern (hauptsächlich graduierte Stundenten) wurde im Dezember 1969 ein experimentelles Netz mit Knoten an vier Universitäten in Betrieb genommen. Das Netz wuchs schnell und überspannte bald die Vereinigten Staaten von Amerika (Abb. 1a, 1b, 1c).

Dieser Ausbau zeigte auch auf, dass die vorhandenen ARPA-Netz-Protokolle nicht für den Betrieb über mehrere Netze geeignet waren. Diese Beobachtungen führten zu weiteren Forschungsarbeiten über Protokolle, die schliesslich in der Entwicklung des TCP/IP-Modells und der relevanten Protokolle resultierte (Cerf und Kahn, 1974). TCP/IP wurde speziell zur Abarbeitung von Übertragungen über verbundene Netze entwickelt, was immer wichtiger wurde, da mehr Netze an das ARPA-Netz angeschlossen wurden.

Die Netze, Maschinen und Benutzer, die an das ARPA-Netz angeschlossen wurden, nahmen rapid zu, nachdem TCP/IP am 1. Januar 1983 das einzige offizielle Protokoll wurde. Viele regionale Netze kamen dazu, und Verbindungen wurden mit Netzen in Kanada, Europa und dem pazifischen Raum hergestellt. Etwa mitte der achziger Jahre betrachteten die Beteiligten die Sammlung von Netzen als *einen* Netzverbund, später als *das* Internet.

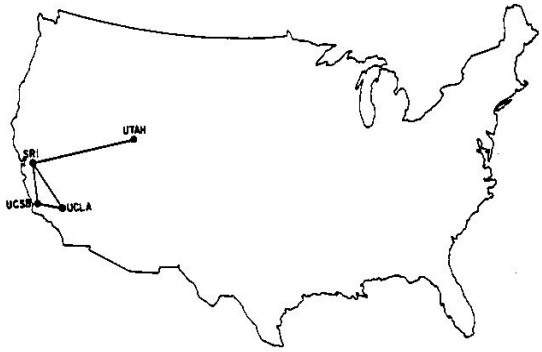


Abb. 1a: ARPA-Netz, Dezember 1969

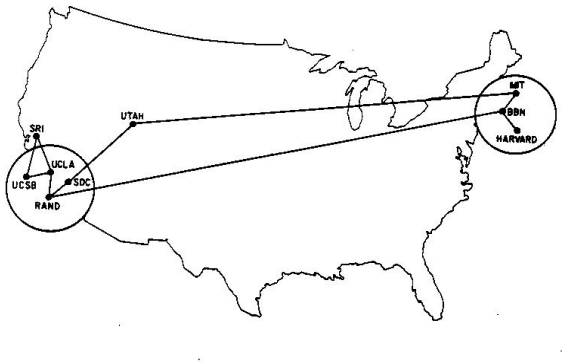


Abb. 1b: ARPA-Netz, Juni 1970

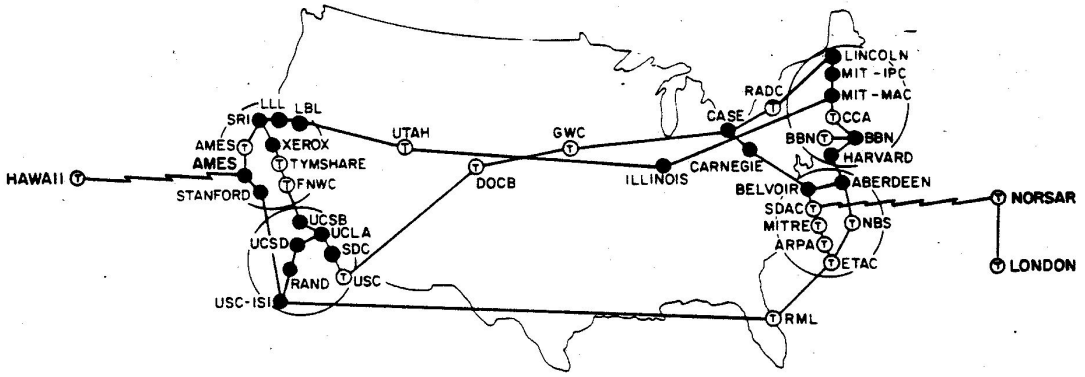


Abb. 1c: ARPA-Netz, September 1973

2.2 Netzwerk-Schichtenmodell

Unabhängig von der Architektur der zugrundeliegenden Rechner ist die Netzwerkkommunikation ein sehr komplexes und abstraktes Thema. Um dieses Thema besser zu strukturieren und damit leichter auf die erhöhten Anforderungen des Marktes reagieren zu können, entwickelte die International Organization for Standardization (ISO) 1977 ein sogenanntes Referenzmodell für „Open System Interconnection“ (OSI), das ISO-OSI-Referenzmodell (Abb. 2). Hierbei handelt es sich um ein Schichtenmodell, wobei die transportorientierten Funktionen in vier (Schicht 1 bis 4) und die datenverarbeitungsorientierten Funktionen in drei Schichten (Schicht 5 bis 7) unterteilt sind.

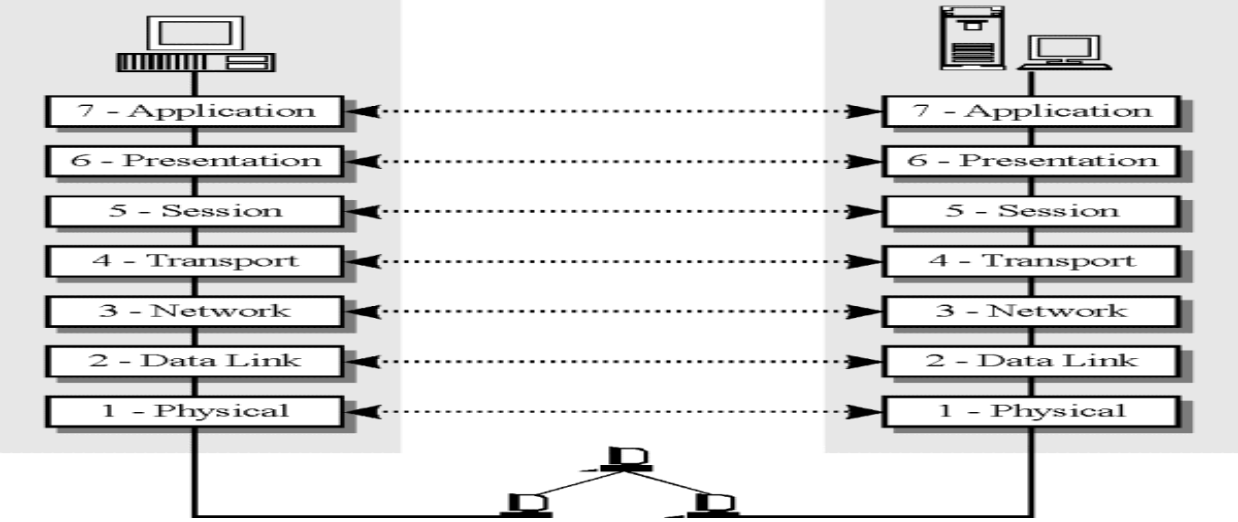


Abb. 2: ISO-OSI-Referenzmodell

Die grundlegende Idee hinter einem Schichtenmodell ist, dass jede beteiligte Schicht einer darüberliegenden Schicht bestimmte Dienste anbietet. Damit schirmt sie die höheren Schichten von Details ab, wie die betreffenden Dienste realisiert sind. Dadurch ist es möglich, dass eine Schicht n (n = 1 – 7) des einen Computers mit der selben Schicht n eines anderen Computers kommuniziert. Die Regeln und Konventionen dieser Kommunikation werden als Protokolle der Schicht n bezeichnet.

In der Realität kommunizieren die Schichten nicht direkt miteinander. Jede Schicht reicht ihre Daten und zusätzliche Kontrollinformationen an die direkt darunterliegende Schicht weiter, bis die tiefste Schicht erreicht ist. In dieser Schicht liegt das physikalische Medium, durch das die echte Kommunikation stattfindet (Signalübertragung via Licht oder Wellen in einem Leiter).

Zwischen einem Paar übereinanderliegenden Schichten besteht eine definierte Schnittstelle (Interface). Das Interface bestimmt die Operationen und Dienste, die die untere der oberen Schicht anbietet. Die obere Schicht greift über sog. Service Access Points (SAP) auf die Dienste der direkt darunter liegenden Schicht. Ein Satz von Schichten und Schnittstellen wird Netzwerkarchitektur genannt. Eine Liste von Protokollen, die von einem bestimmten System genutzt werden – ein Protokoll pro Schicht – wird Protocol Stack genannt (typisches Beispiel: Internet Protocol Stack, Abb. 3).

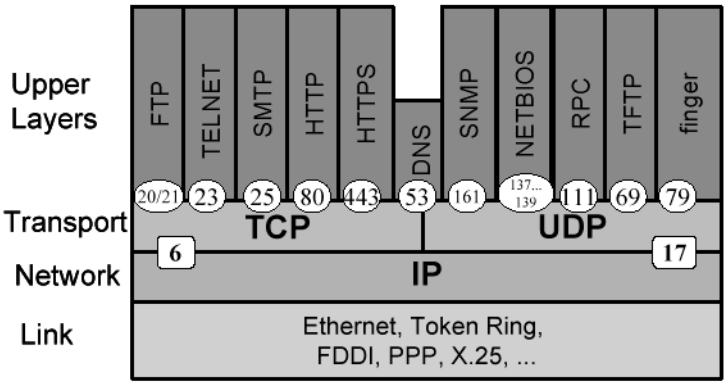


Abb. 3: Internet Protocol Stack

Typischerweise addiert jedes Protokoll bestimmte Kontrollinformationen (Header) zu den Nutzdaten der darüberliegenden Schicht, wenn sie von oben nach unten durch die Schichten gereicht werden. Diese sind für den Gegenpart beim Empfänger gedacht. Diese zusätzlichen Header werden beim Empfänger dann auf dem Weg zur obersten Schicht wieder entfernt.

Schichten können zwei Arten von Diensten nach oben bereitstellen. Die eine Art ist verbindungsorientiert und am ehesten mit einem Telefonsystem zu vergleichen. Man wählt den Partner an, kommuniziert mit ihm und trennt die Verbindung wieder. Die zweite Art ist verbindungslos und orientiert sich am Postsystem. Jede Nachricht wird mit einer vollständigen Adresse versehen und wird durch das System zum Empfänger geleitet. Hierbei kann es im Gegensatz zu verbindungsorientierten Diensten vorkommen, dass sich die Reihenfolge von verschickten Nachrichten durch unterschiedliche Verzögerungszeiten im System verändert.

Jeder Dienst wird dabei durch die sogenannte Quality-of-Service (QoS) charakterisiert. Diese Dienstgüte bezieht sich auf die Zuverlässigkeit der Übertragung sowie weiterer Merkmale wie Sicherheit oder Effizienz bezüglich der Geschwindigkeit.

2.3 Komponenten des Internet

Institute, Behörden und Firmen vernetzen in immer stärkerem Masse ihre verwendeten Rechnerplattformen. Dies dient im wesentlichen zum leichteren Austausch wichtiger Daten und Informationen, der Nutzung von netzweiten Ressourcen (Drucker, Massenspeicher, etc) sowie der einfacheren Wartbarkeit von einer entfernten Konsole aus oder gar zur multimedialen Kommunikation. Die physikalische Ausdehnung dieser lokalen Netze (LAN = Local Area Network) beschränkt sich in der Regel auf ein oder mehrere Gebäude in relativer Nähe (< 10 km). Die Grösse von LANs ist aus technischen Gründen (z.B. Signaldämpung) beschränkt.

Die Übertragungstechnologie gängiger LANs besteht zumeist aus einem Kabel, das alle Maschinen verbindet. Als Topologie ist daher entweder ein Bus oder ein Ring vorgesehen (Abb. 4). Neuere Technologien fordern ein eigenes Kabel für jeden angeschlossenen Rechner, was zu einer Sterntopologie mit einem zentralen Sternverteiler (Hub) führt. Noch aufwendiger sind vermaschte Topologien.

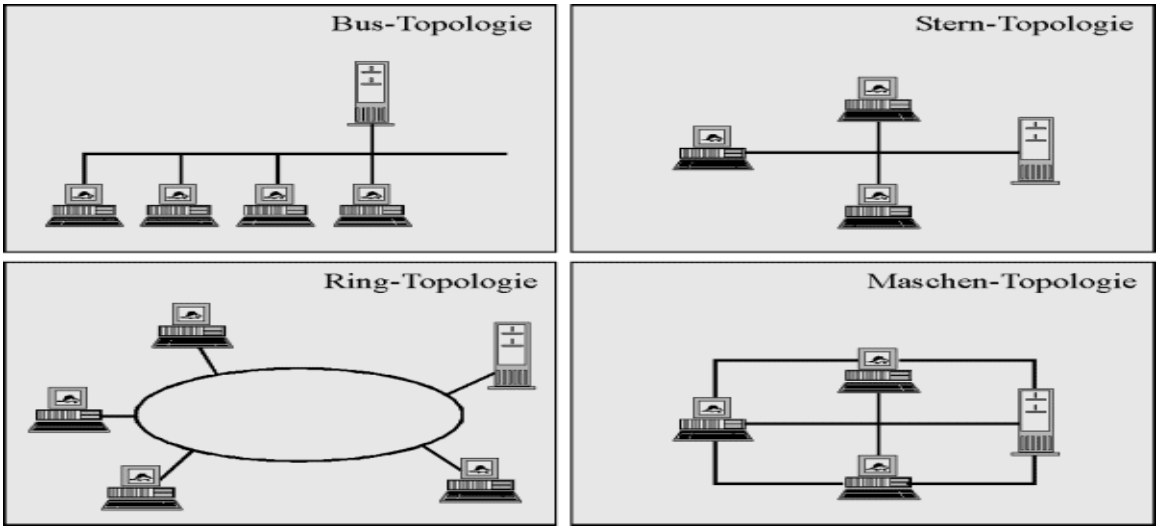


Abb. 4: Netzwerktopologien im LAN

Die verbreiteste LAN-Technologie ist Ethernet (Bus: 10/100/1000 Mbit/s). Hier kann jeder Rechner zu jeder Zeit auf das Netzwerk senden. Wesentlich beim Ethernet ist die Art zur Vermeidung von Kollisionen über ein spezielles Protokoll (CSMA-CD). Es sorgt dafür, dass bei einer Kollision beim Senden von Daten von zwei Rechnern die Transmission sofort unterbrochen wird. Diese Daten werden verworfen und nach Zeiten, die von Zufallsgeneratoren bestimmt werden, beginnen die Rechner wieder aufs neue, ihre Daten zu senden.

Um den Datenverkehr zwischen Firmen oder Filialen innerhalb von Städtegrenzen zu ermöglichen, wurden MANs (Metropolitan Area Networks) als die vergrößerte Ausgabe von LANs installiert. Ihre wichtigsten Vertreter sind FDDI (Fibre Distributed Data Interface) mit Transferleistungen von 100 Mbit/s und DQDB (Distributed Queue Dual Bus) mit skalierbaren Übertragungsraten von 34, 45 oder 140 Mbit/s.
Eine große Bedeutung auf dem Netzmarkt besitzen weltumspannende WANs (Wide Area Networks, Abb. 5). Sie basieren in der Regel auf physikalischen Leitungen, die von den nationalen Telekom-Unternehmen betrieben werden. Ein prominenter Vertreter einer physikalischen Netztechnologie ist ISDN (Integrated Service Digital Network, 128 kBit/s bis 2 Mbit/s). Eine andere Technologie für WANs, die jedoch auch für LANs und MANs eingesetzt werden kann, ist ATM (Asynchronous Transfer Mode), die Datentransferraten von 25, 50, 155 oder 622 Mbit/s erlaubt und speziell für zeit- und synchronisationskritische multimediale Datentypen geeignet ist.

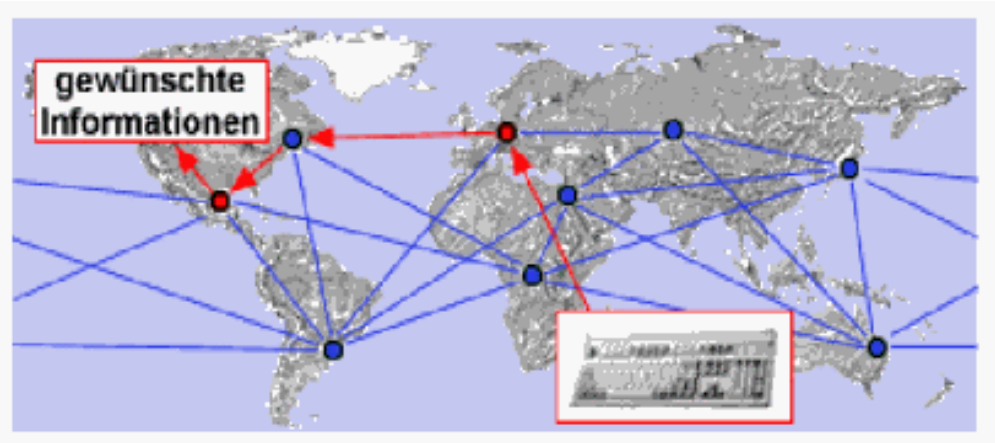


Abb. 5: Verbindung mehrer Netze zu einem WAN

Elektrische Signale können auf physikalischen Leitungen nur eine begrenzte Distanz zurücklegen, ohne dabei an Leistung zu verlieren. In LANs werden daher verschiedene Geräte genutzt, um die Signale zu regenerieren und mit anderen LANs oder WANs zu kommunizieren:

Repeaters: Sie regenerieren elektrische Signale. Damit erlauben sie die Verbindung zweier Kabelabschnitte zur Verlängerung des Netzstrangs auf Schicht 1.

Bridges: Sie erlauben die Verbindung zweier LANs auf Schicht 2. Jede Station im jeweiligen LAN kann damit auf Ressourcen des anderen LANs zugreifen. Bridges erlauben auch die Kombination verschiedener Netzkabeltypen.

Hubs: Sie arbeiten ähnlich wie Bridges, sind jedoch für sternförmige Netzwerktopologien gedacht.

Routers: Sie können je nach Adresse eines Datenpakets dessen Weiterleitung oder die Zurückweisung bewirken. Dies kann helfen, unnötigen Netzverkehr in LANs drastisch zu senken, da Datenpakete nur durchgelassen werden, wenn die Zieladresse jenseits des Routers liegt.

Die Verbindung sehr vieler LANs, MANs und WANs auf der OSI-Schicht 3 ergibt dann das Internetwork oder einfacher das Internet, welches durch Router gekoppelt sind.

3 Konnektivität zum Internet

3.1 Privatpersonen

Um als Privatperson „online“ zu sein, verbindet man seinen Computer mit Hilfe eines Modems über das öffentliche Telefonnetz (analog oder via ISDN), über ein Kabelnetz oder gar über das Stromnetz mit einem Internet Service-Provider (ISP). In der Regel muss dazu eine Einwählnummer, ein Benutzername und ein (geheimes) Passwort des ISP bekannt sein. Für die Dauer der Verbindung ist nun der private Computer Teil des Internet und kann so alle anderen Computer des Internet erreichen. Der private Computer kann aber auch (theoretisch) von allen anderen Computern des Internet angewählt werden.

3.2 Firmen

Firmen verbinden ihre eigenen Computernetze über Mietleitungen (Standleitungen) mit einem ISP. Sie sind so dauernd an das Internet angeschlossen (Abb. 6).
Die Internet-Technologien lassen sich in den Firmen natürlich nutzbringend einsetzen. Entsprechend wurde der Name Intranet kreiert. Während das Internet eine völlig uneingeschränkte Kommunikation erlaubt, wird das Intranet vom öffentlichen Bereich durch spezielle Sicherheitsgeräte – sog. Firewalls – abgetrennt: Der Zugriff vom Internet auf interne Firmenrechner wird verunmöglicht bzw. genau kontrolliert; der Zugang von der Firma zum Internet ist kaum eingeschränkt.
Firmenrechner, welche vom Internet her auch weitgehend uneingeschränkt erreichbar sein sollen, werden ausserhalb des Firewalls plaziert (Web- und Mail-Server). Damit diese nicht für andere Zwecke missbraucht werden, wird häufig zum Internet hin ein zweiter Firewall installiert, welcher den Zugriff auf die öffentlichen Server genau kontrolliert. Der Bereich zwischen den beiden Firewalls wird demilitarisierte Zone (DMZ) genannt (Abb. 7).

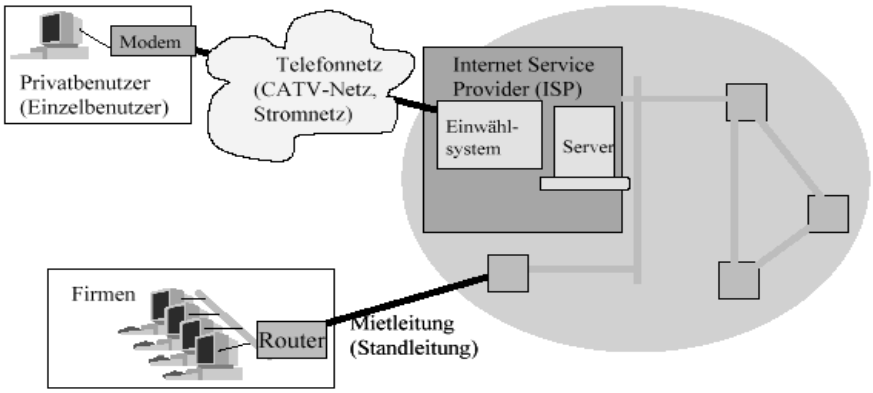


Abb. 6: Anschluss von Privaten und Firmen an das Internet

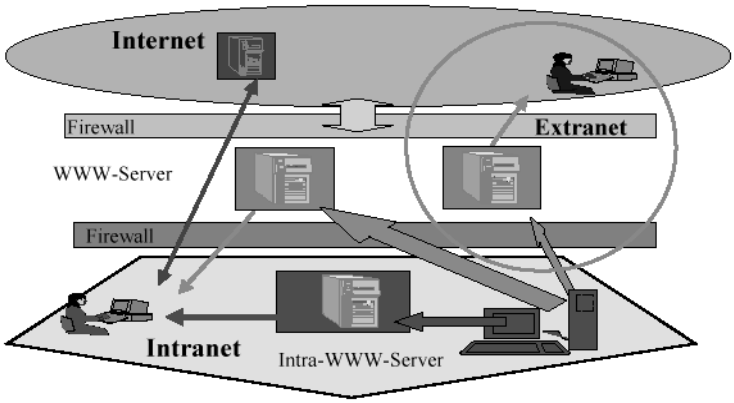


Abb. 7: Internet/Intranet

4 Schlussfolgerungen

Inwiefern das Internet neben der Orts-, Zeit- und Inhaltssouveränität auch einen weltweiten und völlig transparenten Marktplatz schafft, wird die Zukunft zeigen. Es ist aber unwahrscheinlich, dass eine solche Transparenz des Marktes ohne Einfluss auf die Margen sein wird. Die Internet-Entwicklung bringt auch neue Möglichkeiten für an sich bekannte Handelsformen. Hier sind beispielsweise Auktionen zu erwähnen, bei welchen die Anbieter oder die Käufer Produktpreise via Versteigerungen festlegen. Eine andere interessante Entwicklung ist die Bildung von Kaufgemeinschaften, welche durch den gemeinsamen Kauf von Produkten besondere Rabatte aushandeln.
Zu erwähnen ist auch die Entwicklung im E-Government-Bereich, welche es ermöglichen würde, Wahlen und viele andere administrative Tätigkeiten (z.B. Steuererklärung, Anfragen an Gemeinden) auf elektronischem Wege durchzuführen.
Bemerkenswert sind auch die Fortschritte beim E-Commerce und E-Business.

Die Frage, „was kommt nach dem Internet“, beantwortet man wohl am besten mit „das Internet“. Allerdings ist abzusehen, dass noch vor 2005 nicht nur einzelne Freaks mobil surfen und weltweit kommunizieren werden. Die Fix- und Mobilkommunikation wird mehr und mehr zusammenwachsen. Damit dürfte sich das Internet über Heim und Geschäft, hinaus auch auf das Auto oder generell auf den mobilen Menschen ausdehnen.

5 Literaturverzeichnis

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

Grundlagen der Internet-Technologie

2.Netzwerktechnologien für das Internet

PPS Seminar SS 01

Verfasser:
Thomas Haag
Betreuer: Jan Gerke

30.April 2001

Netzwerktechnologien für das Internet

1. Einführung

Um zwischen zwei Computern Daten auszutauschen, muss man diese verbinden. Je mehr Computer man verbinden will, je schneller man Daten austauschen will, je grösser die zu überwindenden Distanzen sind, desto schwieriger wird, es dies fehlerfrei zu realisieren. Ein einziges falsches Bit genügt, um ein gesamtes Programm unbrauchbar zu machen. Je nach Verwendungsbereich sind verschiedene Kriterien wichtig, deshalb sind verschiedene Technologien entwickelt worden, um jeden Verwendungszweck abzudecken. Es herrscht jedoch ein stetiges Ringen um Protokollstandards und Marktanteile.
In den folgenden Abschnitten werden verschiedene Netzwerklösungen vorgestellt, die technischen Grundlagen erläutert, Vor- und Nachteile hervorgehoben. Unter Punkt vier werden die verschiedenen Technologien miteinander verglichen.

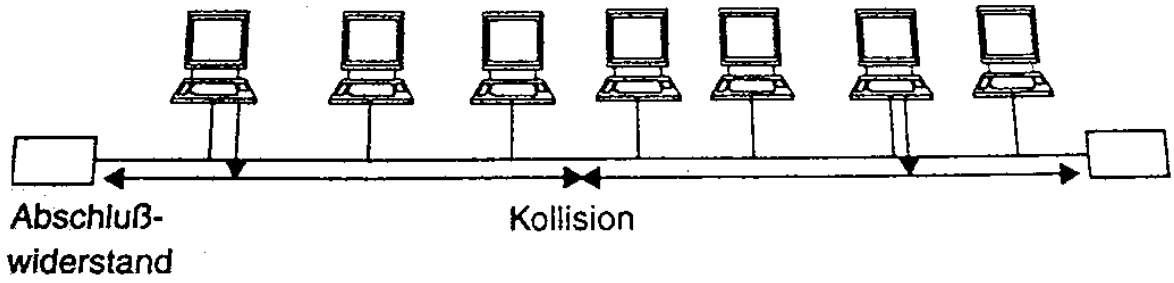
2. Lokale Netzwerke (LAN, „Lokal Area Network“)

Lokale Netzwerke sind in ihrer Grösse eingeschränkt, was zählt ist die Länge der Verbindung zwischen den zwei am weitesten entfernten Computern. Je nach Technologie und Geschwindigkeit der Übertragung variiert die maximale Kabellänge zwischen wenigen Hundert bis einigen Tausend Metern.

2.1 Ethernet

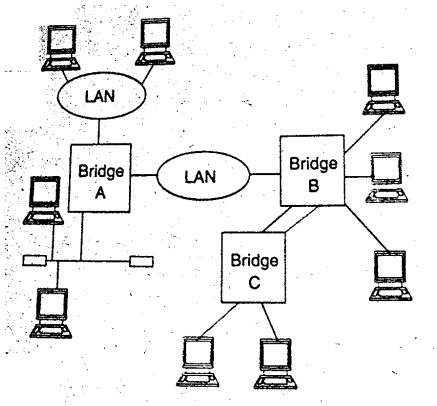
Das am weitesten verbreitete System ist Ethernet. Im Ethernet- Netzwerk teilen sich alle Computer dasselbe Kabel. Es kann jeweils nur ein Computer auf ein mal senden, dazu benötigt man einige Tricks. Bevor ein Computer senden will schaut er ob die Leitung frei ist, nur wenn gerade kein anderer etwas sendet, beginnt er mit der Übertragung. Jetzt kann es trotzdem noch sein, dass zwei Computer beinahe gleichzeitig zu senden beginnen, dies wird jedoch von beiden Computern erkannt und sie hören beide mit der Übertragung auf (Collision). Beide warten nun eine zufällige Zeit lang, um nicht schon wieder gleichzeitig zu senden und schicken danach das „kollidierte“ Paket nochmals.
Es ist notwendig, dass jede Kollision erkannt wird, sonst entstehen Übertragungsfehler, daraus lässt sich die maximale Grösse des Netzwerkes bestimmen. Bevor ein Computer das gesamte Paket gesendet hat, muss der Anfang des Paketes jeden angeschlossenen Computer erreicht haben.

Wenn man dies jetzt für die minimale Paketgrösse errechnet, ergibt das bei herkömmlichen Ethernet (10 Mbit/s) etwa 2,5Km, beim zehnmal schnelleren FAST Ethernet (100 Mbit/s) sind nur noch wenige 100 m möglich.
Beim neuen Gigabit- Ethernet (1000 Mbit/s) ist die minimale Paketlänge erhöht worden, um ein weiteres Beschränken der Grösse zu verhindern.



2.2 Bridges (Switches)

Wenn man nun mehrere LAN's zusammenschliessen möchte, verwendet man dazu Bridges, sie werden zwischen die Netzwerke geschaltet und stellen dort eine Verbindung her. Bridges wirken wie eine Postzentrale, die Adresse des Daten- Paketes wird gelesen und an das entsprechende Netz weitergeleitet. Bridges mit vielen Anschlüssen werden auch Switches genannt. (Unterschied zu einem HUB, der HUB sortiert nicht, er sendet an alle ohne auszuwählen).
Es gibt im Wesentlichen zwei verschiedene Techniken für Bridges, das **Source Route Bridging** und das **Transparent Bridging.** Beim Transparent Bridging „merken" sich die Bridges selbst, wohin sie welche Adressen schicken müssen und welche sie einfach ignorieren können. Das Source Route Bridging hingegen hat den Vorteil, dass sich Bridges keine Adressen merken müssen. Der Computer selbst sendet zu Beginn ein *Discovery*-Paket (eine Art Ping) an alle Computer, der gesuchte Computer sendet eine Antwort zurück, die die Information mit dem zurückgelegten Weg enthält. Der Sender kann nun explizit über den gespeicherten Pfad senden.



2.3 Token Ring

Die Token Ring Technik ist eine andere Art Computer zusammenzuschliessen. Es gilt auch wieder, Kollisionen zu vermeiden. Um dies zu erreichen, wird von den Computern, die am Netz angeschlossen sind, eine Art Marke (Token) im Kreis herumgegeben, nur der Computer der gerade die Marke besitzt, hat das recht Daten zu senden, jedoch nur bis die Token-holding-Time abgelaufen ist (limitierte Paketgrösse), dann muss er die Marke weitergeben. Vorteil dieses Systems, es entstehen keine Kollisionen (= Zeitverlust durch Neusenden), auch bei stark belastetem Netz, hat jeder Computer die Chance etwas zu senden. Man hat die Garantie, dass man nach einer festgelegten Zeit wieder Gelegenheit hat zu senden (Token-holding-Time * anz. Computer = Maximale Wartezeit). Nachteil, wenn eine Verbindung ausfällt, ist der Token „Kreis“ unterbrochen, das Netzwerk ist tot.

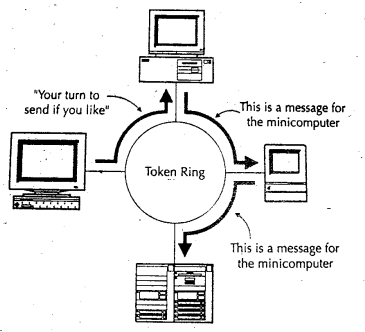
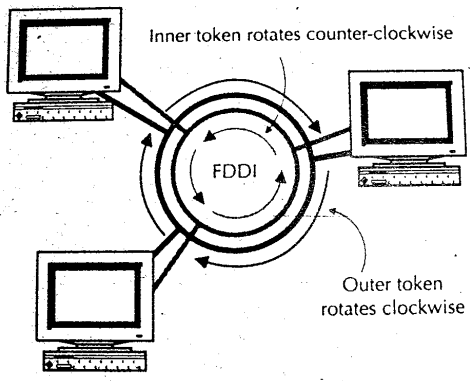


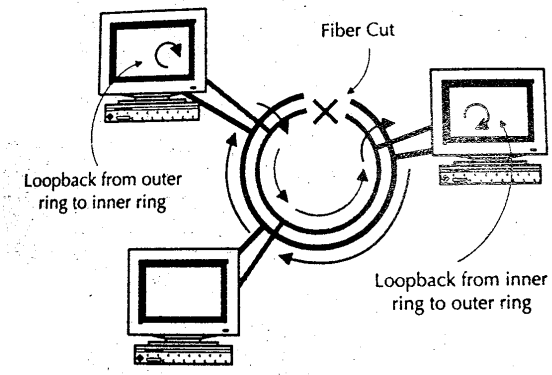
Figure 3.12 Sending data across a Token Ring LAN.

2.4 FDDI (Fiber Distributed Data Interface)

Eine Weiterentwicklung des Token Rings ist das FDDI, es beruht auf derselben Ringstruktur aber auf Glasfasertechnologie, was einen Geschwindigkeitsvorteil bringt. Als zusätzlichen Bonus wurde einer der grossen Schwächen ausgemerzt, neu besteht das Netzwerk aus zwei gegeneinanderlaufenden Ringen, die bei Ausfall einer Verbindung „selbstheilend“ wirken (siehe Bild).



FDDI uses two separate rings.

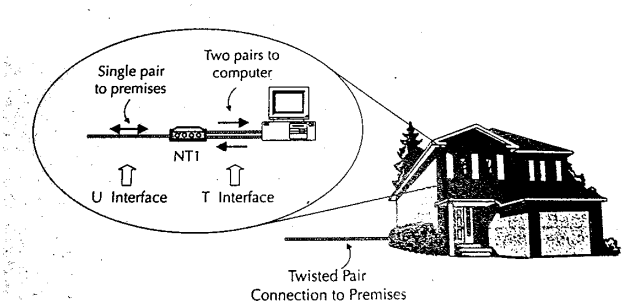


Automatically repairing an FDDI fiber cut.

3. Weitverkehrsnetzwerke (WAN, „Wide Area Network“)

Wie der Name schon sagt geht es bei WAN’s darum, grosse Distanzen zu überwinden. Sie verbinden Städte, Länder und Kontinente. Um dies zu erreichen, wird eine andere Technologie verwendet, es wird vor dem Daten-Transfer eine „feste“ Verbindung erstellt, die exklusiv für den Sender und Empfänger reserviert ist (keine Kollisionen möglich). Nach der Übertragung wird die Verbindung wieder abgebaut. Diese Übertragungsart nennt man verbindungsorientiert, im Gegensatz zu den verbindungslosen Techniken wie Ethernet oder Token Ring.

3.1 ISDN (Integratet Services Digital Network)



Interfacing to ISDN.

ISDN wie wir es als Hausanschluss kennen wird Basic Rate Interface genannt (BRI), mit einer fixen Transferrate von 128kbit/s, daneben gibt es noch das Primary Rate Interface (PRI) mit 1,92 Mbit/s (1,472 Mbit/s in der USA).
Ein BRI besteht aus zwei B channels mit je 64kbit/s zur Datenübertragung und einem D channel mit 16kbit/s zur Steuerung. Beim Übermitteln von Daten wird zuerst über den Steuerungskanal eine Verbindung erstellt, die solange bestehen bleibt, bis sie wieder mit einem RELEASE Befehl aufgehoben wird.
Für den Heimgebrauch wird das ISDN-Signal im NT1 (Network Termination 1) auf das zweipolige Telefonkabel kodiert.

3.2 X.25 Standard und Frame Relay

Das X.25 Protokoll bietet im Vergleich zu ISDN, die Möglichkeit mehrere Kanäle auf einer physikalischen Leitung zu senden, dadurch können Bandbreiten besser ausgenutzt werden. Der typische Arbeitsbereich von X.25 liegt zwischen 200kbit/s und 2Mbit/s. Es werden zusätzlich zu den Daten auch noch Fehler- und Flusskontrollfunktionen übertragen. Dadurch wird aber bei jeder Paketübertragung noch viel „Ballast“ mitgeliefert. Eine Weiterentwicklung von X.25 ist das Frame Relay, es kann mit höheren Datenraten umgehen. Zudem wurde einiges an „Ballast“ weggekürzt, die Fehlerkorrektur wurde entfernt und ist nun Sache der Software. Dies wurde möglich durch die bessere Übertragungsqualität der Leitungen.

3.3 ATM (Asynchronous Transfer Mode)

ATM ist ähnlich wie ISDN aufgebaut, ebenfalls Verbindungsorientiert, dies bringt unter anderem den Vorteil, dass die versendeten Pakete in der richtigen Reihenfolge ankommen und nicht zuerst sortiert werden müssen. Es ist jedoch möglich auf einem physischen Kabel mehrere virtuelle Kanäle einzurichten. Man kann sogar eine feste Bandbreite reservieren, was vor allem für Echtzeitanwendungen wichtig ist (Telephonie, Video-Streaming).
In der ATM- Adaptionsschicht können diese Parameter gesetzt werden:

- Bandbreite fest/variabel
- verbindungsorientiert/verbindungslos
- isochron/asynchron

Die Verbindungsarten wurden in vier Klassen unterteilt, A-D (siehe Tabelle).

ATM wird in Weitverbindungsnetzwerken eingesetzt, durch die hohe Flexibilität, wurde es aber auch schon in einigen lokalen Netzwerken eingebaut, wo es Datenraten bis 622Mbit/s erlaubt. Die hohe Flexibilität stellt jedoch auch hohe Anforderungen an die Hardware, was sich im Preis niederschlägt.

	Klasse A	Klasse B	Klasse C	Klasse D
Synchronität	isochron		asynchron	
Bitrate	konstant	variabel		
Verbindungs-modus	verbindungs-orientiert			verbindungs-los
Anwendun-gen	Emulation synchroner Dienste (ISDN)	Video mit va-riabler Bitrate (MPEG...)	verbindungs-orientierte Datenkommu-nikation	verbindungs-lose Daten-kommunika-tion
AAL	AAL1	AAL2	AAL3/4 und AAL5	

3. Übersicht

	LAN			WAN			
	Ethernet	**Token R**	**FDDI**	**X.25**	**Frame R**	**ISDN**	**ATM**
Verborien.1.)	Nein	Nein	Nein	Ja	Ja	Ja	Möglich
fixe Bitrate	Nein	Nein 2.)	Nein 2.)	Ja	Ja	Ja	Möglich
Bitrate 3.)	10-1000	4 od. 16	100	2	Über 2	,128-1,92	622
Paketv. 4.)	Ja	Ja	Ja	Ja	Ja	Nein	Nein

1.) Verbindungsorientiert, Gegenteil ist Verbindungslos
2.) man hat die Garantie nach einer festen Zeit wieder senden zu dürfen
3.)maximale Übertragungsrate in Mbit/s
4.)paketvermittelnd: Daten werden in Pakete unterteilt und adressiert. Sie finden mit Hilfe der Adresse den richtigen Empfänger, Gegenteil ist leitungsvermittelnd

5. Zusammenfassung

Eine Übertragungstechnologie ist stets ein Kompromiss, es werden zu viele verschiede Anforderungen gestellt, um diese alle optimal zu erfüllen. Zur Dateiübertragung ist es z.B. egal ob der Schluss einer Datei zuerst ankommt, oder wie konstant die Bandbreite ist, die durchschnittliche Datenrate ist entscheidend. Beim Telefonieren über Internet ist es jedoch recht mühsam, wenn immer wieder Übertragungsunterbrüche entstehen, ein möglichst gleichmässiger Fluss ist entscheidend. Flexible Technologien sind gefragt.
Die Entwicklung ist jedoch rasant (exponentiell), vor einigen Jahrzehnten konnte man einige Textzeilen, heute schon bewegte Bilder übertragen. Ein Ende ist noch nicht in Sicht...

Literaturangaben:

S. Thomas: IPng and the TCP/IP Protocols;John Wiley & Sons, Inc. New York, USA, 1996,Seiten 43-76

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

PPS Seminar
Grundlagen der Internet-Technologie

Internet Protokoll, Adressierung und Routing im Internet

Otto Huber

1. Das siebenschichtige ISO/OSI Referenzmodell

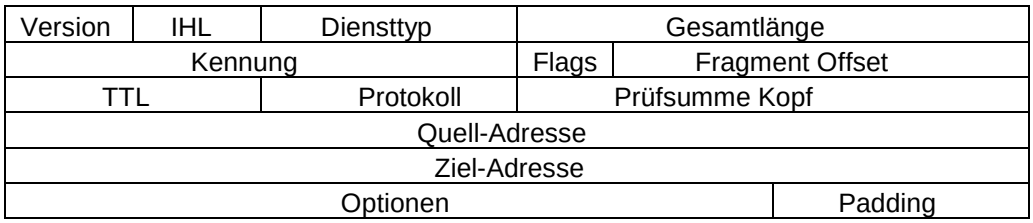
(International Standard Organization/ Open System Interconnection)

Nummer	**Allgemeine Funktion**	**Protokoll Beispiele**
(7) application layer / Anwendungsschicht	Die Anwendung, wie sie sich dem Benutzer präsentiert	Telnet, ftp, WWW, mail, news, rlogin
(6) presentation layer / Darstellungsschicht	Hier finden Konvertierungen von Daten statt, sofern dies notwendig ist (z.B. Umwandlung von ASCII in EBCDIC-Zeichensatz)	
(5) Session layer / Sitzungsschicht	Steuerung des Verbindungsaufbaus. Schnittstelle des Programmierers zum Netzwerk. Ist eine Verbindung zu einem anderen Rechner hergestellt, verwaltet diese Schicht die Verbindung.	
(4) transport layer / Transportschicht	Die Transportschicht überträgt die Daten vom Quellrechner zum Zielrechner. Aufgaben der Transportschicht ist das folgerichtige Übertragen eines Datenstromes, das Garantieren von Eigenschaften, die beim Verbindungsaufbau angefordert wurden (verbindungsorientierter Dienst mit geordnetem Verbindungsabbau, verbindungsloser Dienst (Datagramm))	TCP, UDP
(3) network layer / Netzwerkschicht	Senden und Empfangen von Paketen im Netz. Wichtigste Aufgabe dieser Schicht ist dabei das "Routen" der Datenpakete im Netz. Die Routen können statisch sein oder auch zur Laufzeit bestimmt werden. In dieser Schicht werden auch Gebühreninformationen zur Abrechnung gesammelt.	IP
(2) data link layer / Verbindungsschicht	Übertragungssicherung der physikalischen Schicht. Dies wird durch paketweises Übertragen der Daten erreicht. Emfangsbestätigungen des Empfängers müssen verarbeitet werden. Sollten Pakete fehlerhaft übertragen werden, so wird von dieser Schicht das Paket erneut übertragen.	Ethernet, ISDN, Datex-P (X.25), PPP V.42, V.42bis
(1) physical layer / Physikalische Schicht	Übertragung der Bits auf einem Übertragungsmedium. Hier werden die physikalischen Eigenschaften des Übertragungsmediums festgelegt (Kupfer/Lichtwellen-leiter, Übertragungsfrequenz, welche Spannung bedeutet 0, welche 1, parallele oder serielle Übertragung....	Lichtwellenleiter, Koaxkabel, Twisted Pair, Telefon, V.24

2. IP (Internet Protocol)

TCP und IP wurden vom DOD (Department of Defense), dem amerikanischen Verteidigungsministerium entwickelt. Das Ziel dieser zwei Protokolle war es, verschiedene Netzwerke von verschiedenen Anbietern zu dem Netzwerk der Netzwerke (dem “Internet“) zu verbinden. Somit bildet TCP/IP ist den kleinsten gemeinsamen Nenner des gesamten Datenverkehrs im Internet. Ungeachtet dessen, ob sie www-Seiten aufrufen, E-Mails versenden, mit FTP Dateien downloaden oder mit Telnet auf einem entfernten Rechner arbeiten: Stets werden die Daten auf gleiche Weise adressiert und transportiert.
Das TCP (Transmission Control Protocol) definiert auf der der Transportschicht (Transport Layer) des ISO/OSI Referenzmodells ein Host-zu-Host Protokoll im Rahmen der Internet Protokoll Familie. Das IP (Internet Protocol) bildet in dieser Protokoll-Familie eine Realisierung der Schicht 3, der Vermittlungsschicht (Network Layer). Die übliche gemeinsame Nennung von TCP und IP als ein gemeinsames Protokoll ist, genau betrachtet, eine gewisse Nachlässigkeit, da im Grunde genommen TCP und IP voneinander unabhängige Realisierungen der Schichten 4 und 3 nach dem ISO/OSI-Referenzmodell sind.
Das Internet Protokoll verfügt über keine Funktionen zur Flussregelung oder Fehlerbehebung, es ist ein verbindungsloses Konzept, das jedes Paket unabhängig weiterleitet, was eine schnelle Reaktion auf Ausfälle oder Überlastungen von Teilnetzen ermöglicht.

Die Reihenfolge der Pakete wird nicht unbedingt eingehalten, es ist gut möglich, dass später gesendete Pakete schneller ans Ziel geraten, wo sie dann auf Grund der Kennung im Header wieder in die richtige Reihenfolge gebracht werden müssen.
Das TCP und das IP an sich beinhalten keine Sicherung und Verschlüsselung der Daten, diese müssen auf den höheren Layern implementiert werden. Sicherheiten auf dem Network layer bietet zum Beispiel das IPSec. Das IPSec ist eine, im Internet Protokoll integrierte Protokollerweiterung. Dieses Protokoll wurde von der Internet Engeneering Task Force (IETF) entwickelt. Das IPSec bietet Sicherheit für die Übertragung von heiklen Informationen auf ungeschützten Netzwerken. Der IPSec-Sender kann vertrauliche Daten verschlüsseln bevor er sie über ein Netzwerk sendet. Das IPSec identifiziert den Sender der Daten und überprüft, ob seine Daten nicht verändert wurden. Der IPSec Empfänger kann erkennen ob Daten mehrfach gesendet wurden und kann diese gegebenenfalls zurückweisen.



Format einer IP-Dateneinheit

Der feste Kopfteil (Header) umfasst alle diejenigen Felder, die auf jeden Fall erforderlich sind. Er hat die Länge von fünf 32-Bit Worten.

Feld	Aufgabe	Länge
Version	Versionsnummer des Protokolls	4
Kopflänge (IHL)	Länge des IP-Kopfs in 32-Bit-Worten	4
Diensttyp	Angabe des Diensttyps	8
Gesamtlänge	Gesamtlänge in Byte inklusive Kopf und Daten	16
Kennung	Identifikationswert von Dateneinheiten	16
Flags	z.B. Anzeige, ob Fragmentierung zugelassen	3
Fragment-Offset	Anfangsposition in der gesamten Dateneinheit	13
Lebenszeit	Maximaler Hop-Count der Dateneinheit	8
Protokoll	Kennung des Schicht-4-Protokolls (z.B. 7 für TCP)	8
Prüfsumme	Prüfsumme über den Kopf einen IP-Dateneinheit	16
Quelladresse	Internet-Adresse des Quellsystems	32
Zieladresse	Internet-Adresse des Zielsystems	32
Optionen	keine, eine oder mehrere Optionen variabler Länge	k*32

2.1 Die Felder einer IP-Dateneinheit

Im *Diensttyp* enthalten sind drei Prioritätsbit, ein D-Bit, Verzögerung fordert, ein T-Bit, mit der Forderung nach einer gewünschten Durchsatzrate und ein R-Bit, welches einen Wunsch nach Zuverlässigkeit anbringt. Zudem sind noch zwei weitere reserviert. Garantie dieser Wünsche wird absolut nicht geboten, die Interpretation der einzelnen Prioritätsbits ist sogar noch vom Hersteller des Router abhängig und nicht exakt genormt.
Die drei *Flag-Bits* geben an, ob eine Fragmentierung einer Dateneinheit zugelassen ist. Eine IP-Dateneinheit hat theoretisch die maximale Grösse von 2^{16} Bytes (ein Byte pro Bit im Feld der Gesamtlänge), doch in einem Ethernet lässt das IP-Paket eine Maximallänge von 15024

Bytes zu. Daher müssen die TCP-Pakete, die diese Länge überschreiten in mehrere IP-Pakete geteilt werden. Unter diesem Teilungsvorgang leidet die Performace eines Netzwerkes.
Die *Prüfsumme* bietet die einzige Fehlerkontrolle im IP-Datenpaket, doch damit wird nur der Kopf auf Fehler überprüft. Im Gegensatz zum heute geläufigen IPv4 (Internet Protocol version 4) soll der Nachfolger (IPv6) gewisse Sicherungsmöglichkeiten der Daten beinhalten.
Das *Optionen-Wort* enthält ein acht Bit langes Kodierungsfeld und Platz für die Angabe von Optionen, wie z.B. Record Routing (Aufzeichnung der Route) oder Source Routing (genau vorgeschriebene Route einhalten). Der optionale Kopfteil endet immer auf eine 32 Bit Grenze, daher kann er ein Padding als „Lückenfüller" enthalten. Dieser Teil im Header lässt zwar die Effizienz beim Verarbeiten von IP-Dateneinheiten etwas leiden, dafür sind sie flexibel für Erweiterungen.

3. Adressierung im Internet

Wie weiss ein Computer welchen Weg er im Netz zu wählen hat, um möglichst gezielt an den gewünschten Empfänger zu gelangen?

3.1 IP-Adressierung

In einem Netzwerk hat jede Maschine eine eigene Adresse, über welche sie erreichbar ist und mit deren Identität sie sich im Netz einwählt. Diese Adresse ist auf jeden Fall ein Bit-Code, der im Falle des Internet Protokolls 32 Bit lang ist. Es werden jeweils acht Bit zu einem Byte zusammengefasst und zur Vereinfachung im Dezimalsystem für den Menschen geschrieben.
Eine typische IP-Adresse sieht so aus: 10100000010101011001001001000010 oder für uns etwas leichter lesbar: 160.85.146.66 - vier Zahlen also, getrennt durch Punkte. Die Punkte haben die Aufgabe, über- und untergeordnete Netze anzusprechen. So wie zu einer Telefonnummer im weltweiten Telefonnetz eine Landeskennzahl, eine Ortsnetzkennzahl, eine Teilnehmerrufnummer und manchmal auch noch eine Durchwahlnummer gehört, gibt es auch im Internet eine Vorwahl - die Netzwerknummer, und eine Durchwahl - die Hostnummer.
Der erste Teil einer IP-Adresse ist die Netzwerknummer, der zweite Teil die Hostnummer. Wo die Grenze zwischen Netzwerknummer und Hostnummer liegt, bestimmt ein Klassifizierungsschema für Netztypen. Die folgende Tabelle verdeutlicht dieses Schema. In den Spalten für die IP-Adressierung und einem typischen Beispiel ist die Netzwerknummer (der Vorwahlteil) fett dargestellt. Der Rest der IP-Adresse ist die Hostnummer eines Rechners innerhalb dieses Netzes.

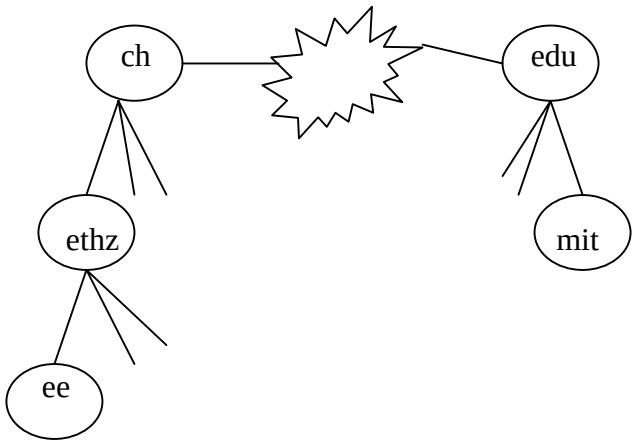
Netztyp	IP-Adressierung	Typische IP-Adresse
Klasse-A-Netz	**xxx**.xxx.xxx.xxx	**103**.234.123.87
Klasse-B-Netz	**xxx.xxx**.xxx.xxx	**160.85**.146.113
Klasse-C-Netz	**xxx.xxx.xxx**.xxx	**194.191.253**.130

Die oberste Hierarchiestufe bilden die sogenannten Klasse-A-Netze. Nur die erste Zahl einer IP-Adresse ist darin die Netzwerknummer, alle anderen Zahlen sind Hostnummern innerhalb des Netzwerks. Das erste Bit beim Bit-Code eines Klasse-A-Netzes ist eine 0, die folgenden 7 Bits geben die Netzwerknummer an. Bei Netzwerknummern solcher Netze sind daher Zahlen zwischen 1 und 126 möglich, d.h. es kann weltweit nur 126 Klasse-A-Netze geben. Das amerikanische Militärnetz oder das Firmennetz der IBM sind beispielsweise solche ein Klasse-A-Netz. Innerhalb eines Klasse-A-Netzes kann der entsprechende Netzbetreiber die zweite, dritte und vierte Zahl der einzelnen IP-Adressen seiner Netzteilnehmer frei vergeben. Da alle drei Zahlen Werte von 0 bis 255 haben können, kann ein Klasse-A-Netzbetreiber also bis zu 16,7 Millionen IP-Adressen an Host-Rechner innerhalb seines Netzes vergeben.

Die zweithöchste Hierarchiestufe sind die Klasse-B-Netze. Die Netzwerknummer solcher Netze erstreckt sich über die beiden ersten Zahlen der IP-Adresse. Die ersten zwei Bit beim Bit-Code eines Klasse-B-Netzes sind 10. Bei der ersten Zahl können IP-Adressen von Klasse-B-Netze also Werte zwischen 128 und 192 haben. Bei der zweiten sind Zahl Werte zwischen 0 und 255 erlaubt. Dadurch sind etwa 16 000 solcher Netze möglich. Da die Zahlen drei und vier in solchen Netzen ebenfalls Werte zwischen 0 und 255 haben dürfen, können an jedem Klasse-B-Netz bis zu ca. 65 000 Hostrechner angeschlossen werden. Klasse-B-Netze werden vor allem an große Firmen, Universitäten und Online-Dienste vergeben.
Die unterste Hierarchiestufe stellen die Klasse-C-Netze dar. Die ersten drei Bits des Bit-Codes eines Klasse-C-Netzes sind: 110. Die erste Zahl einer IP-Adresse eines Klasse-C-Netzes liegt somit zwischen 192 und 223. Die Zahlen zwei und drei gehören ebenfalls noch zur Netzwerknummer. Über zwei Millionen solcher Netze sind dadurch adressierbar. Vor allem an kleine und mittlere Unternehmen mit direkter Internet-Verbindung, auch an kleinere Internet-Provider, werden solche Adressen vergeben. Da nur noch eine Zahl mit Werten zwischen 0 und 255 übrig bleibt, können in einem C-Netz maximal 255 Host-Rechner angeschlossen werden.

3.2 Domain Name System

Das Domain Name System gibt an, welche Domänen einer Spezifischen Domäne über- oder untergeordnet sind. Dies lässt sich am einfachsten anhand eines Baumes erklären: Ganz rechts steht die Wurzel eines Baumes und links davon befinden sich die Unterbäume.



Beispiel: Wir befinden uns im Rechenzentrum der Elektrotechnik (www.ee.ethz.ch) und möchte die Seite web.mit.edu anschauen. In diesem Beispiel werden wir zuerst mit der Domäne ethz verbunden und da diese die gewünschte Domäne nicht kennt, leitet sie die Anfrage an die Domäne ch weiter. Diese kennt die Seite wahrscheinlich immer noch nicht direkt, sie kann dafür die Anfrage an die Domäne edu senden, welche die Seite web.mit.edu kennt und unserer Maschine auf dem selben Weg die IP-Adresse dieser zukommen lässt, damit die Seite anschliessend aufgerufen werden kann. Die Anfrage ist zeitlich limitiert, damit sie nicht beliebig lange im Netz umherirrt, falls eine Domäne gewählt wurde, die nicht existiert.

3.3 Subnetting

Um Routing-Tabellen etwas übersichtlicher zu gestalten und die langsam knapp werdende Anzahl (2^{32}) von IP-Adressen etwas effizienter auszunutzen, wurden die Subnetze

eingerichtet. Sie dienen dazu, dass ein grosses Netzwerk, wie zum Beispiel dasjenige der IBM, in kleinere Unternetze geteilt werden. Bemerkung am Rande: Das Problem der knapp werdenden Anzahl von IP-Adressen wird mit dem IPv6 vorläufig behoben sein, da dieses 2^{128} IP-Adressen zur Verfügung stellen wird. Dies sind ca. $4.5 * 10^{28}$ Adressen pro Mensch auf der Erde oder ca. 1500 Adressen pro Quadratmeter Erdoberfläche.

3.4 DHCP (Dynamic Host Configuration Protocol)

Das DHCP überall dort gebraucht, wo IP-Adressen dynamisch vergeben werden müssen. Dies ist insbesondere im Zeitalter des Mobile-Computing sehr wichtig. Wenn also jemand seinen Laptop einem Subnetz anschliesst, muss dieser eine IP-Adresse bekommen um sich in diesem Netz zu identifizieren. Diese Adresse erhält er vom DHCP, welches die Kompatibilität und die Zugriffsrechte regelt.
Auch wenn sich ein Computer, der eine Dial-up Verbindung zu seinem ISP hat, bei diesem einwählt, so muss er solange er im Internet ist eine eigene Identität und somit eine IP-Adresse haben. Diese normalerweise vom ISP dynamisch zugeteilt und ein Computer hat nicht jedes Mal dieselbe IP-Adresse wenn er sich beim ISP einwählt.

4. Routing im Internet

In kleinen Netzwerken, wie zum Beispiel in unserem kleinen Hausnetzwerk ist es relativ einfach unter den Rechnern zu kommunizieren. Mein Computer weiss einfach, dass wenn er per TCP/IP drucken will seinen Druckauftrag an die interne Adresse 192.168.0.5 senden muss. Also sendet er diese Datenpakete an den Hub und dieser wiederum sendet diese in unser ganzes Netzwerk hinaus. Diese Pakete gelangen anschliessend an alle Maschinen in unserem Netzwerk, sie werden aber nur vom Printserver aufgenommen, da dieser die entsprechende Adresse besitzt. Habe ich aber eine Anfrage an die Adresse 160.85.146.66, welche in unserem Netzwerk nicht vertreten ist, so weiss unser Router als Gateway zur Aussenwelt, dass er sich darum zu kümmern hat. Darauf sendet er seine Anfrage an die Adresse 194.191.253.130 oder 194.191.253.133, die nächst höhere Instanz nämlich zu den Routern unseres Providers. Dieser wiederum leitet diese auf Grund von Routingtabellen weiter. Jeder Router sendet in regelmässigen kurzen Abständen Signale an die umliegenden Router, um festzustellen, ob sich neue Wege für seine Pakete eröffnet haben, oder ob allenfalls einer ausgefallen ist.
Deshalb gibt es im Internet folgende Routing Methoden:
Im Internet als dem Netz der Netze ist es zunächst nur innerhalb des eigenen Sub-Netzes möglich, Daten direkt von einer IP-Adresse zu einer anderen zu schicken. In allen anderen Fällen, wenn die Daten an eine andere Netzwerknummer geschickt werden sollen, treten Rechner auf den Plan, die den Verkehr zwischen den Netzen regeln. Solche Rechner werden als Gateways bezeichnet. Diese Rechner leiten Daten von Hostrechnern aus dem eigenen Sub-Netz an Gateways in anderen Sub-Netzen weiter und ankommende Daten von Gateways anderer Sub-Netze an die darin adressierten Host-Rechner im eigenen Sub-Netz. Ohne Gateways gäbe es gar kein Internet.

Das Weiterleiten der Daten zwischen Sub-Netzen wird als Routing bezeichnet. Die Beschreibung der möglichen Routen vom eigenen Netzwerk zu anderen Netzwerken sind in Routing-Tabellen auf den Gateway-Rechnern festgehalten. In unserem Beispiel von zu Hause beschränkt sich diese Routing-Tabelle auf die beiden IP-Adressen unseres ISPs.

Zu den Aufgaben eines Gateways gehört auch, eine Alternativ-Route zu finden, wenn die übliche Route nicht funktioniert, etwa, weil bei der entsprechenden Leitung eine Störung oder

ein Datenstau aufgetreten ist. Gateways senden sich ständig Testpakete zu, um das Funktionieren der Verbindung zu testen und für Datentransfers "verkehrsarme" Wege zu finden.

Wenn also im Internet ein Datentransfer stattfindet, ist keinesfalls von vorneherein klar, welchen Weg die Daten nehmen. Sogar einzelne Pakete einer einzigen Sendung können völlig unterschiedliche Wege nehmen. Wenn Sie beispielsweise von der Schweiz aus eine WWW-Seite aufrufen, die auf einem Rechner in den USA liegt, kann es sein, dass die Hälfte der Seite über den Atlantik kommt und die andere über den Pazifik, bevor Ihr WWW-Browser sie anzeigen kann. Weder Sie noch Ihr Browser bemerken dies, da dies die Sorge der tieferen Layers des OSI/ISO-Referenzmodells ist.

5. Bibliografische Referenzen

1. T. Braun: Ipng – Neue Internet-Dienste und virtuelle Netze
2. S. Thomas: Ipng and the TCP/IP Protocols
3. M. Zitterbart, T. Braun: Hochleistungskommunikation
4. A. Tannenbaum: Computernetzwerke
5. http://zeus.fh-brandenburg.de
6. http://web.mit.edu
7. http://www-lsv.informatik.rwth-aachen.de
8. http://www.rz.rwth-aachen.de
9. http://www.cisco.com
10. http://www.3com.com
11. http://www.glossar.de
12. http://www.telematik.uni-karlsruhe.de
13. Bertelsmann Lexikon

PPS – Grundlagen der Internet Technologie

Das Internet Protokoll der Version 6

by Yannick Thebault

29.04.2001

Internet Protokol Version 6 – next generation

Einleitung

Das Internet Protokol Version 6 (IPv6) wird in Zukunft das heutige IP Verion 4 dabei ablösen, Anlass dazu ist die steigende Anzahl benötigter IP Adressen, dabei werden jedoch noch eine ganze Reihe anderer Features in das neue Protokoll implementiert. Bereits 1992 wurde der Entschluss gefasst ein neues Protokoll zu entwickeln und kurz darauf machte sich die IETF - Internet Engineering Task Force an die Arbeit. Diese Zusammenfassung soll einen kurzen Überblick über das wieso und warum sowie die technischen Erneuerungen verschaffen.

Das Internet Protokoll im Allgemeinen

Das IP übernimmt eine ganz Zentrale Rolle im Internet, es implementiert einen Standart um Daten über verschiedene Netzwerke zu transportieren, die zu transportierenden Daten werden dabei in kleine Stücke zerlegt und verpackt bevor sie auf die Reise geschickt werden, am Zielort werden die Einzelstücke wieder zu einer Einheit zusammengesetzt. Vorteilhaft dabei, ist die Tatsache, dass sich IP wenig kümmern muss auf welchem Transportweg es sich gerade befindet, also egal ob die Daten durch die Luft übertragen werden oder durch eine Glassfaser weiterbefördert. Auf das IP setzten sich nun Protokolle wie TCP und UDP die einen Standart definieren auf dem wiederum die verschiedenen Dienste wie FTP, HTTP, IRC usw. aufbauen. IP definiert dabei die Adressierung und die Optionen des Paketes, zum Beispiel: Wo soll das Paket hin. Um was für ein Paket handelt es sich.

Die Adressierung: IPv4 definiert sogenannte IP-Adressen, diese müssen in einem abgeschlossenem Netzwerk *einmalig* sein, d.h. Im Internet darf jede IP-Adresse nur einmal benutzt werden, jeder der in einem solchen Netzwerk kommuniezieren will braucht eine IP-Adresse. Die Adresse besteht aus 4 x 8bits, also Gesamt 32bits, welche theoretisch maximal 4 Milliarden Adressen zur Verfügung stellen. Dies sind aber entschieden zuwenig, wenn man den Visionären glauben schenkt, die in Zukunft jeder Kaffemaschine oder

Wäschetrommel eine solche zuweisen möchten, um sie ans Internet anzuschliessen. IPv6 definiert deren Adressen 3.4E38, da die Adresse 128 bittig ist, also stehen in Zukunft beinahe unbegrenzte Mengen an Adressen zur Verfügung.
IPv6 trägt die Versionsnummer 6, weil IPv5 bereits durch das Stream Protocol Version 2 belegt ist, auf das hier nicht weiter eingegangen wird.

Das IP Paket

Wie schon erwähnt unterteilt das IP die Daten in kleine Pakete und schickt sie auf die Reise. Ein IP Paket besteht im wesentlichen aus einem header (evtl noch extension headers) und der eigentlichen Information, somit stellt der header die eigentliche Verpackung des Paketes dar. Typischerweise ist ein Paket zwischen 500 und 1500 Bytes gross, IPv6 unterstützt jedoch Pakete bis zu 4 GByte, der Header benötigt dabei 40 Bytes (ohne extension headers).
Auf dem Weg durchs Netz, passiert das Pakete einige Router, der Router behandelt jedes IP-Paket völlig unabhängig und „interessiert" sich lediglich für den header, die übertragene Information lässt er dabei unberührt.
Grundsätzlich gibt es 3 Methoden ein Paket zu verschicken – Unicast, Anycast und Mulicast, die alle ein Präfix der Adresse beschreiben. Ein Paket wird üblicher Weise an eine Unicast Adresse gesendet, d.h. von A nach B. Es ist jedoch auch möglich das ein Paket von A nach Gruppe B, die ein Netzwerk verschiedener Teilnehmer bilden, in diesem Falle kommt das Anycast Präfix zum Einsatz, dabei wird einfach der am Naheliegenste Teilnehmer das Paket empfangen. Es ist auch möglich ein Paket an jeden Einzelnen einer Gruppe zu schicken, dabei wird die Gruppe mittels Mulicast Adresse zusammengefasst.

Der IPv6 header

Der header ist genau standardisiert und muss präzise eingehalten werden, ansonsten ist das Paket ungültig! Im einzelnen besteht ein header aus folgenden Feldern:
o Version (4 bits)
o Priority (4 bits)
o Flow Label (24 bits)

o Payload Length (16 bits)
o Next Header (8 bits)
o Hop Limit (8 bits)
o Source Adress (128 bits)
o Destination Adress (128 bits)

-> Version: Beschreibt lediglich die Versionsnummer des Paketes, z.b. 4 oder 6
-> Priority: Bei grosser Netzwerklast kann der Router entscheiden welche Pakete er bevorzugt behandeln will, z.b. ist eine interaktive Nachricht „wichtiger", als eine eMail.
-> Flow Label: „Die Flussmarke" ist ein neues Feature und soll IPv6 zum Erfolg führen. Mit einem Flow Label lassen sich Datenströme kennzeichnen, die über eine längere Zeit aktiv bleiben, z.b. ein Multimediastream. Trifft zum ersten Mal ein Paket mit einem neuen Flow Label in einem Router ein, so kann dieser gewisse Ressourcen (CPU, Bandbreite) für diesen Fluss reservieren. Ein Fluss erfolgt immer nur in eine Richtung und muss jeweils immer von A nach B gehen. Das Flow Label soll Anwendungen wie Video-on-demand, Internet-Radio, Voice-over-IP und weitere Realtime-Anwendungen endlich zum Durchbruch verhelfen.
-> Payload Length: Enthält einen interger Wert, der die Länge der Information im Paket entspricht.
-> Next Header: Verweist auf den extension header im Paket, falls vorhanden.
-> Hop Limit: Damit kann die maximale „Hoplänge" der Pakete festlegen. Ein Hop entspricht dem passieren eines Routers, liegen zwischen der Quelle und dem Ziel 5 Router, so beträgt der Hopcount 5. Jeder Router dekrementiert jeweils den Hop Limit Eintrag um 1, bis es 0 erreicht, falls es nun bei einem Router eintrifft wird das Paket verworfen. Das Hop Limit ist ein sehr wichtiger Eintrag, damit lässt sich verhindern, dass ein schlecht konfiguriertes Netzwerk sich zum überlaufen bringt, indem es bei einer Endlosroutingschleife irgendwann Hop Limit 0 erreicht hat und das Paket fallen lässt, ansonsten würden die im Kreis zirkulierenden Pakete das Netzwerk unnötig verstopfen.
-> Source Adress: Enthält die Adresse des Absenders, somit kann der Empfänger feststellen von wo das Paket stammt.

-> Destination Adress: Eigentlich der wichtigste Eintrag, die Adresse des Empfängers. Der Router schaut sich die IP-Adresse an und entscheidet an Hand einer routingtable zu welchem nächsten Router er das Paket weiterschicken wird.

Extension headers

Wie bereits erwähnt gibt es sogenannte erweiterungs header, diese sind ein ziemlich komplexes und umfangreiches Kapitel des IPv6. Der extension header folgt direkt nach dem eigentlichen Standartheader.

Beispiel eines extension headers ist der routing extension header, damit lässt sich bestimmen welchen „Weg" das IP-Paket nehmen soll. Somit lässt sich beispielsweise verhindern, dass sensible Daten ein „unsicheres" Netzwerk passieren, indem man selbst einen anderen Weg bestimmt (falls vorhanden).

Ebenfalls ein wichtiger Teil von IPv6 ist die standartmässige Implementierung eines Authentication extension headers, damit lassen sich sichere 128-bit verschlüsselte Informationen austauschen.

Realated links

http://www.ipv6.org/

http://www.ipv6.com/

http://www.ipv6forum.com/

http://www.wide.ad.jp/wg/ipv6/

TCP und UDP

GIT-Seminar

14.05.2001

by Matías Fernández

1. Introduction

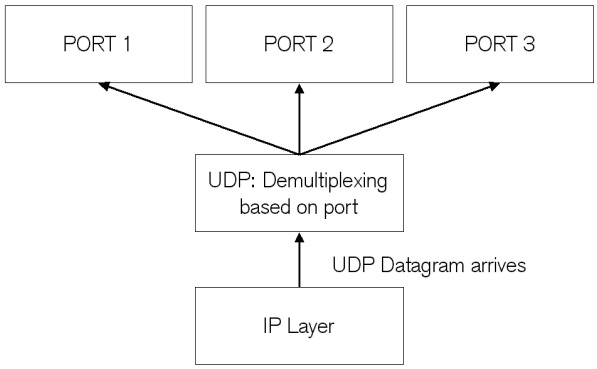
Here we are going to look at two best-known protocols of the transport layer. The User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP). We will talk about the basic concepts behind the two protocols as well as some specific implementations.

2. UDP

UDP (User Datagram Protocol) is the primary mechanism that allows application programs to send datagrams to other application programs on other hosts. It uses IP to transport a message and provides the same unreliable, connectionless datagram delivery. UDP does not make sure packets arrive at the destination and it doesn't control throughput.
Each application program that wants to use the service of UDP is given a port through which it can pass and receive data from UDP. UDP uses this ports to distinguish among various application programs.

2.1. Multiplexing, demultiplexing and ports

UDP accepts data from many application programs and passes them to IP for transmission, it also accepts arriving UDP datagrams from IP and passes them to the appropriate application program. The multiplexing and demultiplexing happens through the port mechanism, each application program must negotiate with the operating system to get a port assigned. Once it has got the port number the application program can send data through that port and UDP will write that port number in the UDP 'source port' field of the UDP header.



The probably best-known internet service is the reliable stream delivery service; it is defined by the Transmission Control Protocol (TCP). TCP is mentioned most commonly as a part of the TCP/IP protocol suite, but it is an independent, general-purpose protocol that can be adapted for use with other delivery systems as well!

3. TCP (TRANSMISSION CONTROL PROTOCOL)

3.1. Do we need stream delivery?

As you know, the computer communication network, at a low level, provides unreliable packet delivery. Remember that packets can be lost or destroyed, they can be delivered out of order or after substantial delay, even duplicates can be delivered.
Application programs from higher layers often need to send large volumes of data from one computer to another, using an unreliable connectionless delivery system would be tedious and annoying. What we would like to have is a general-purpose solution to the problems of providing reliable stream delivery, a single instance of stream protocol software that all application programs use. This would help isolate application programs from the details of networking, and would make it possible to define a uniform interface for the stream transfer service.

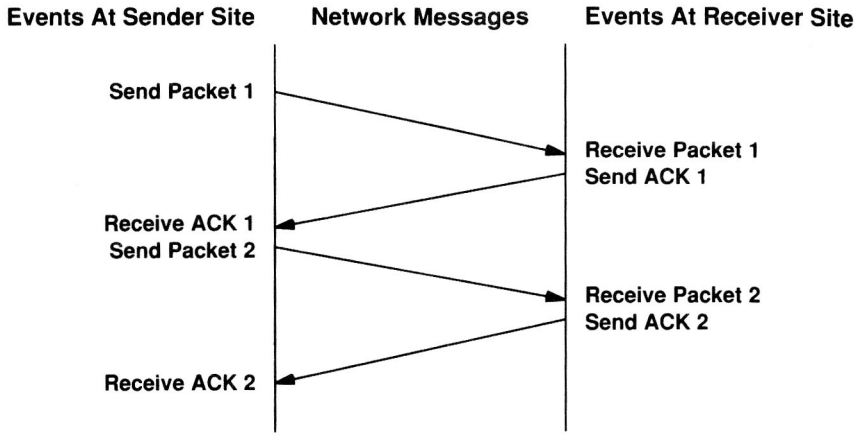
3.2. Five properties of the reliable delivery service

- stream orientation: a sequence of octets passed to the stream delivery service on the source is passed to the receiver in exactly the same order
- virtual circuit connection: making a stream connection is analogous to placing a telephone call. One machine places a 'call', which must be accepted by the other. The application program views the connection as a dedicated hardware circuit, but the reliability is purely an illusion provided by the stream delivery service. That's why we call it 'virtual circuit connection'.
- buffered transfer: the protocol software is free to divide the stream into packets independent of the pieces the application program transfers. Trying to make transfer more efficient and to minimize network traffic, implementations usually collect enough data from the stream to fill a reasonably large datagram before transmitting it across an internet. Thus, if the application program generates the stream one octet at a time, transfer across an internet may be quite efficient. On the other hand the protocol software can choose to divide each block into smaller pieces for transmission if the application program chooses to generate extremely large blocks of data.
- unstructured stream: Application programs cannot control the way TCP structures the stream.full duplex connection: connections provided by TCP/IP stream service allow concurrent transfer in both directions. From the application process' point of view of, a full duplex connection consists of two independent streams flowing in opposite directions, with no apparent interaction.

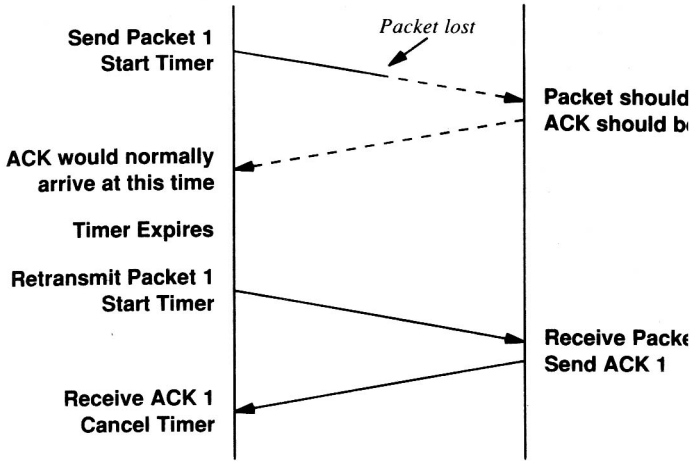
3.3. Providing reliability

How is it possible to provide reliable stream delivery when the underlying communication system only offers unreliable packet delivery? Well, the protocol uses a single fundamental technique called 'positive acknowledgement with retransmission. It requires the recipient to communicate with the source, namely sending back an acknowledgement as it receives data. The sender keeps record of each packet sent and

waits for an acknowledgement before sending a next packet. At the same time the sender starts a timer when it sends a packet and retransmits it, if the timer expires before an acknowledgement arrives.



As you can see here, the sender sends a packet. After a time the packet a while the packet arrives at the receiver which then sends back an acknowledgement back. After receiving the acknowledgement the sender then can send the second packet and so on.

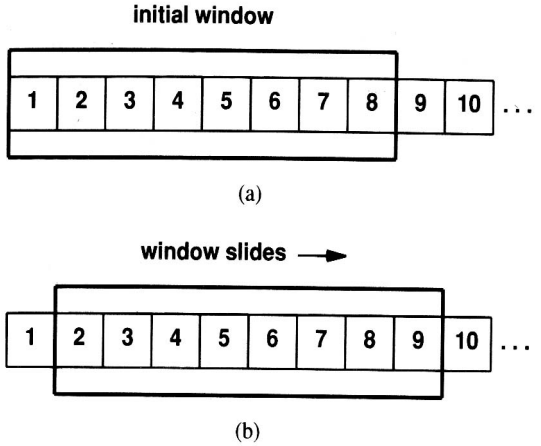


At the same time the sender sends the packet he starts a timer. If the packet is lost, the receiver can't send doesn't receive it and doesn't send back the acknowledgement. The sender waits for the acknowledgement until the timer expires, then he retransmits the same packet.

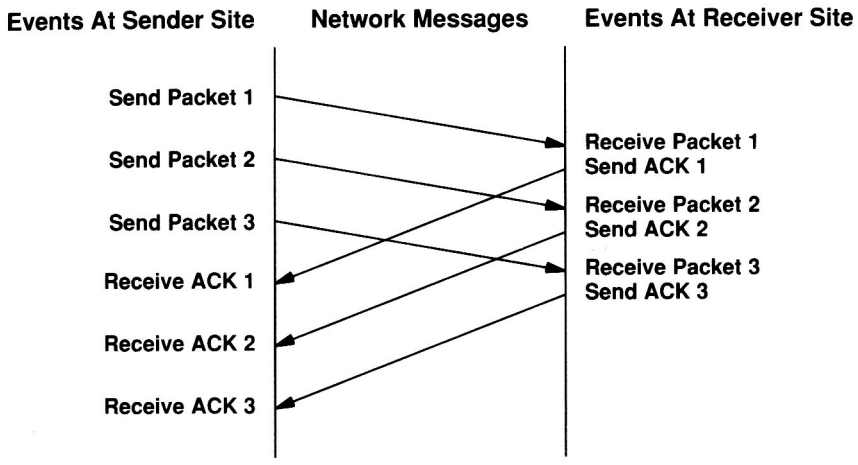
3.4. The idea behind sliding windows

As we have seen, in the simple positive acknowledgement protocol the sender transmits only one packet at a time and then waits for an acknowledgement before transmitting another. This means that data between two machines only flows in one direction at any time, even if the network is capable of simultaneous communication in both directions. It is clear that a simple acknowledgement protocol wastes a substantial amount of

network bandwidth because it must delay sending a new packet until it receives an acknowledgement for the previous packet.
The easiest was to imagine sliding window operations is to think of a sequence of packets to be transmitted. The protocol places a small window on the sequence and transmits all packets that lie inside the window. An unacknowledged packet is a packet that has been transmitted but no acknowledgement has been received yet.



The window slides as soon as the acknowledgement for the first packet in the window has arrived, then it slides along and sends the next packet. The window continuous to slide as long as acknowledgments are received. A well-tuned window protocol keeps the network completely saturated with packets, it obtains substantially higher throughput than a simple positive acknowledgement protocol.



Here an example of a sliding window protocol when sending three packets. The sender transmits all three packets before receiving any acknowledgments. Conceptually, a sliding window protocol always remembers which packets have been acknowledged and keeps a separate timer for each unacknowledged packet. When the sender slides its window, it moves past all acknowledged packets. At the receiving end, the protocol software keeps an analogous window, accepting and acknowledging packets as they arrive.

3.5. Ports, connections and endpoints

TCP allows multiple application programs and a given machine to communicate concurrently, and it demultiplexes incoming TCP traffic among application programs. TCP uses protocol port numbers to identify the ultimate destination within a machine. But it uses the connection, not the protocol port, as its fundamental abstraction; connections are identified by a pair of endpoints. TCP defines an endpoint to be a pair of integers (host, port), where host is the IP address for a host and port is a TCP port on that host.
Here an example for a connection:

(18.26.0.36, 1069) and (128.10.2.3, 25)

Meanwhile another connection might be in progress:

(128.9.0.32, 1184) and (128.10.2.3, 53)

But this is perfectly possible at the same time:

(128.2.254.139, 1184) and (128.10.2.3, 53)

From a programmer's point of view, the connection abstraction is significant. It means a programmer can devise a program that provides concurrent service to multiple connections simultaneously without needing unique local port numbers for each connection.

4. Summary

UDP is a transport protocol as simple as it can be. TCP has additional functions like flow control that responds dynamically to capacity and congestions. But the main difference is that TCP is a reliable stream delivery service, UDP doesn’t guarantee any reliability!

source: internetworking with tcp/ip
by Douglas E. Comer
volume I; principles, protocols, and architecture
second edition
prentice hall international editions
ISBN 0 – 13 – 474321 – 0

HTTP

Hypertext Transfer Protocol

Autor:
Schaller Philippe

Dozent:
Prof. Dr. Stiller

1. Einführung

Das World Wide Web ist ein gigantisches Informationslager. Auf unzähligen Web-Seiten sind diese Informationen gespeichert. Damit man aber überhaupt an die Informationen rankommt, braucht es ein Verfahren, mit dem man darauf zugreifen kann. Und dieses Verfahren nennt sich Hypertext Transfer Protocol, kurz HTTP. Es stellt eine Schlüsselkomponente im Web dar.
Es basiert auf einer Client/Server-Archtektur, bei welcher der Client von einem Server Informationen will und dazu zu dem Server eine Verbindung aufbauen muss. Die meisten Web-Benutzer sind noch nie mit HTTP direkt in Berührung geraten. Man sieht es nur ab und zu als Fehlermeldung „40 4 (not found) „ auf dem Bildschirm, und dann ist man selten erfreut darüber.

40 4 Not Found
The requested URL was not found on this server

Kenntnisse von HTTP sind nur für die wichtig, die wissen wollen wie das Web intern arbeitet. Und natürlich für die, die damit arbeiten, wie etwa Administratoren von Web-Servern und den „Common Gateway Interface „ Programmierern.

2. Geschichte

Beim ursprünglichen Entwurf von HTTP hatte man nur zwei Ziele:

- Einfachheit des Protokolls
- Schnelligkeit des Protokolls

Beim ersten Punkt ging es darum, dass das Protokoll sich problemlos auf allen möglichen Servern und Clients implementieren liess. Natürlich wurde auch darauf geschaut, dass es nicht zuviele Ressourcen verbraucht.
Da der Aufbau des Netzes eine grosse Anzahl von Daten auf einer grossen Anzahl von Servern zur Folge hatte, sollte das Protokoll so schnell wie möglich sein.

2.1 HTTP/0.9

Diese erste Version wurde am Anfang eigentlich nur HTTP genannt. Erst bei der Weiterentwicklung gab man im dann das Anhängsel /0.9. Es war einzig und allein die Methode GET vorhanden. Der Client baute die Verbindung zum Server auf, um dann eine Zeile mit dem Wort GET und dem Namen des Dokumentes an diesen Server zu senden. Nach der Übertragung des Dokumtentes wurde die Verbindung vom Server abgebrochen, um das Ende von eben diesen Dokument anzuzeigen.
Es gab vorwiegend zwei Nachteile. Einerseits konnten nur Textdokumente übertragen werden, und andererseits war es für den Client unmöglich, Daten an den Server zu schicken.

2.2 HTTP/1.0

Bereits 1992 begann man an einer neuen Version von HTTP zu arbeiten, um die oben genannten Nachteile zu überwinden. Die endgültige Version wurde allerdings

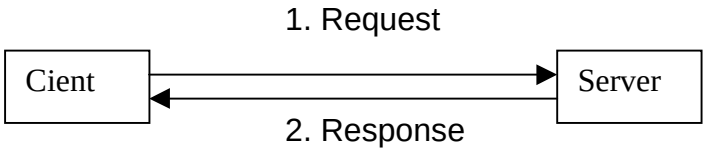
erst im Mai 1996 veröffentlicht. Aber sie war nur Information, diese Ausgabe kam nie wirklich zu Einsatz.
Grundsätzlich basierte es immer noch auf demselben System wie sein Vorgänger. Nach der Request/Response-Interaktion wurde die Verbindung immer noch abgebrochen.
Trotzdem gab es eine Reihe von markanten Verbesserungen. So beinhaltete es das Medientypenkonzept (MIME). Damit konnte der Server zusätzlich zu dem Dokument auch noch Infos über das Dokument senden. HTTP/1.0 definierte ein flexibles Nachrichtenformat, es bestand aus einer Anfangszeile und dann aus Header-Feldern. Diese Header-Felder konnten zum Übertragen von Informationen verwendent werden. Es gab jetzt auch neue eine Reihe von Methoden.

2.3 HTTP/1.1

Bei dieser Version ging es vorallem um die Verbesserung der Effizienz, hohe Ausnutzung der Ressourcen. Es gab einen Konkurrenzkampf zwischen verschiedenen Varianten, die wichtigsten zwei waren das Persistent HTTP (P-HTTP) und das HTTP over Transaction TCP (T/TCP). Die wichtigste Neurerung beim P-HTTP war, dass die Verbindung bestehen blieb. Das gab dann auch den Ausschlag für die Integrierung von P-HTTP in der neuen HTTP/1.0 Version, die im Januar 1997 eingeführt wurde.
Es gab eine Menge kleiner Neuerungen wie etwa das neue Header-Feld HOST, Akzeptanz von absoluten URI's in Requests und neue Request-Methoden. Neu kann man auch nur Teile von Dokumenten anfordern, was vorallem bei grossen Dateien hilfreich sein kann. Auch mit der Content Negotiation wird eine interessante Neuerung eingebracht. Damit kann man zwischen verschiedenen Darstellungsformen einer Ressource entscheiden. Zum Beispiel bei verschiedener Sprache, Qualität, Codierung usw.
Mittlerweile gab es eine Reihe von Revisionen, aber am Grundaufbau wurde immer festgehalten.

3. Funktionsweise von HTTP/1.1

HTTP stellt ein einfaches, auf einem verlässlichen verbindungsorientierten Transportdienst aufbauendes Request/Response-Protokoll dar. Es gibt dabei eine klare Rollenverteilung. Einerseits gibt es den Client, der Requests abschickt, und andererseits den Server, der darauf mit einer Response antwortet. Dazwischen gibt es unterschiedliche Zwischenstationen wie Proxys, Gateways und Tunnels. Alle diese Begriffe sind Programme, und was wichtig ist, sie schliessen einander nicht aus.



Grundlegende Funktionsweise des HTTP

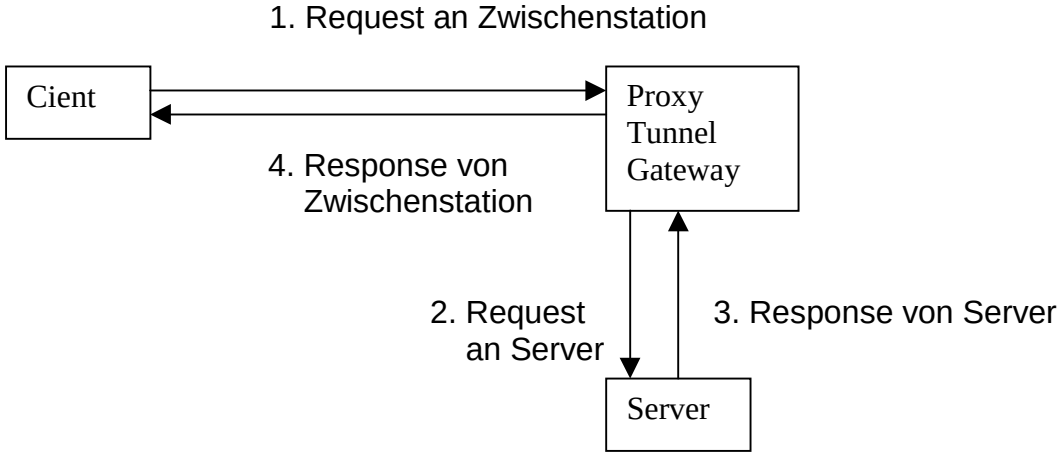
- Client: Baut Verbindungen zum Internet auf, um dann Requests zu versenden. Normalerweise handelt es sich dabei um einen WWW-Browser.

- Server: Dieses Programm lässt solche Verbindungen zu und. Es kann Requests empfangen und verstehen und eine Response zurückschicken.

- Proxy: Das ist eine Art Zwischenstation zwischen Server und Client. Es kann Requests und Responses empfangen und dann wieder weiterschicken. Der Vorteil dabei ist, dass der Proxy Requests auch intern bedienen kann, indem er auf seinen Cache zurückgreift.

- Gateways: Eigentlich ist es dasselbe wie ein Proxy, nur weis hier der Client von gar nichts. Er weiss nicht, dass eine Zwischenstation existiert.

- Tunnel: Ein Tunnel ist eine Verbindungsstation, die Nachrichten nur befördert. Es kann sie weder verstehen noch irgendwie beantworten.



HTTP unter Einbeziehung einer Zwischenstation

4. Aufbau von Nachrichten

Obwohl das Grundprinzip von HTTP sehr einfach ist, hat es sichmit den Jahren zu einem doch recht komplizierten Programm entwickelt. Diese Entwicklung kann man vorallem an der sich immer wieder verändernden Form der beiden Nachrichtentypen Request und Response erkennen.
Gab es am Anfang nur die GET-Methode zusammen mit einem Dateinamen, so braucht es heute einen ganzen Abschnitt von unterschiedlichen Befehlen. Dazu gibt es viele Varianten. Grundsätzlich sollte eine Nachricht aber die nachfolgende Form haben.

Start – line
*message – header
CRLF
[message – body]

Die Start-line ist bei einer Request eine request-line oder bei einer Response eine Status-line. Danach kommen beliebeig viele Header-Felder. Diese lassen sich in vier Gruppen einteilen.

- General Header: Kann bei beiden Nachrichtentypen eingesetzt werden. Solche Header beinhalten Informationen über die Nachricht, nicht aber über das transportierte Dokument.
- Entity Header: Gibt es ebenfalls für beide Nachrichtentypen. Es beinhaltet sogenannte Metainformationen über das Entity. Dabei handelt es sich um Informationen über Länge und Codierung des Entitys, nicht um dessen Inhalt.
- Request Header: Kann Informationen über den Request und sogar über den Client beinhalten. Aber auch mit diesen Headern werden keine Informationen über den Nachrichtenkörper weitergegeben.
- Response Header: Mit diesen Headern werden Informationen über den Server und über die beförderte Nachricht verschickt, über die Ressource.

5. Content Negotiation

In vielen Fällen findet man eine Ressource in allen möglichen oder wenigstens in mehreren verschiedenen Varianten vor. Vom Request muss dann eine Entscheidung getroffen werden, welche Variante es verwendent. Warum es verschiedene Varianten von einer Ressource gibt, ist einleuchtend. Nachfolgend sind die wichtigsten Gründe aufgezählt.

- Sprachspezifische Varianten
- Qualitätsspezifische Varianten
- Codierungsspezifische Varianten

Es gibt zwei entgegengesetzte Arten der Content Negotiation. Die Server-Driven Content Negotiation und die Agent-Driven Content Negotiation. Darüber hinaus können die beiden auch noch kombiniert werden.

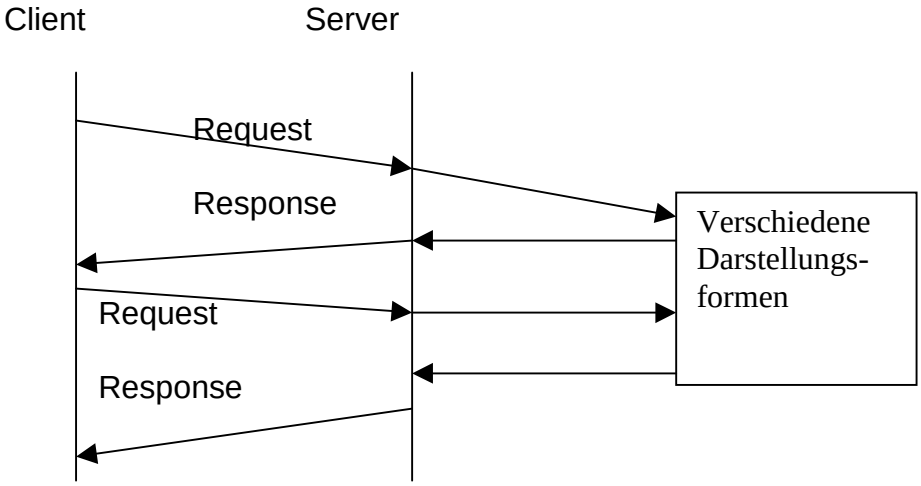
5.1 Server-Driven Content Negotiation

Falls sich der Server für die Auswahl einer bestimmten Darstellungsform verantwortlich zeigt, dann spricht man von Server-Driven Content Negotiation. Der Server kann dabei auf unterschiedliche Informationsquellen zurückgreifen.
Er weiss genau, über welche verschiedenen Varianten er verfügen kann, welche Darstellungsformen möglich sind. Dann kann der Client mit Request-Headern die Auswahl noch eingrenzen, indem er für ihn geeignete Darstellungsformen angibt. Daneben existieren auch noch andere Quellen wie zum Beispiel die Netzwerk-adresse des Clients usw.
Wo immer es unterschiedliche Arten oder Versionen von Programmen oder anderen Dingen gibt, da gibt es auch Vor- und Nachteile. So hat auch die Server-Driven Content Negotiation ihre Pro und Kontras.
Sicher ein Vorteil ist es, wenn sich die Auswahl einer bestimmten Darstellungsform nur schwer beschreiben lässt oder vollkommen auf server-internen Kriterien beruht. Ein grosser Nachteil ist es indes, weil der Server für eine optimale Entscheidung alle Informationen über den Client haben sollte, was aber nur theoretisch möglich ist. Auch ist es ineffizient, wenn in jedem Request alle Informationen über den Client stehen müssen, obwohl vielleicht der Server nur über eine Darstellungsform verfügt.

Dazu kommt noch, dass dieses System die Server-Implementierung verkompliziert und auch einiges an Rechenleistung abverlangt.

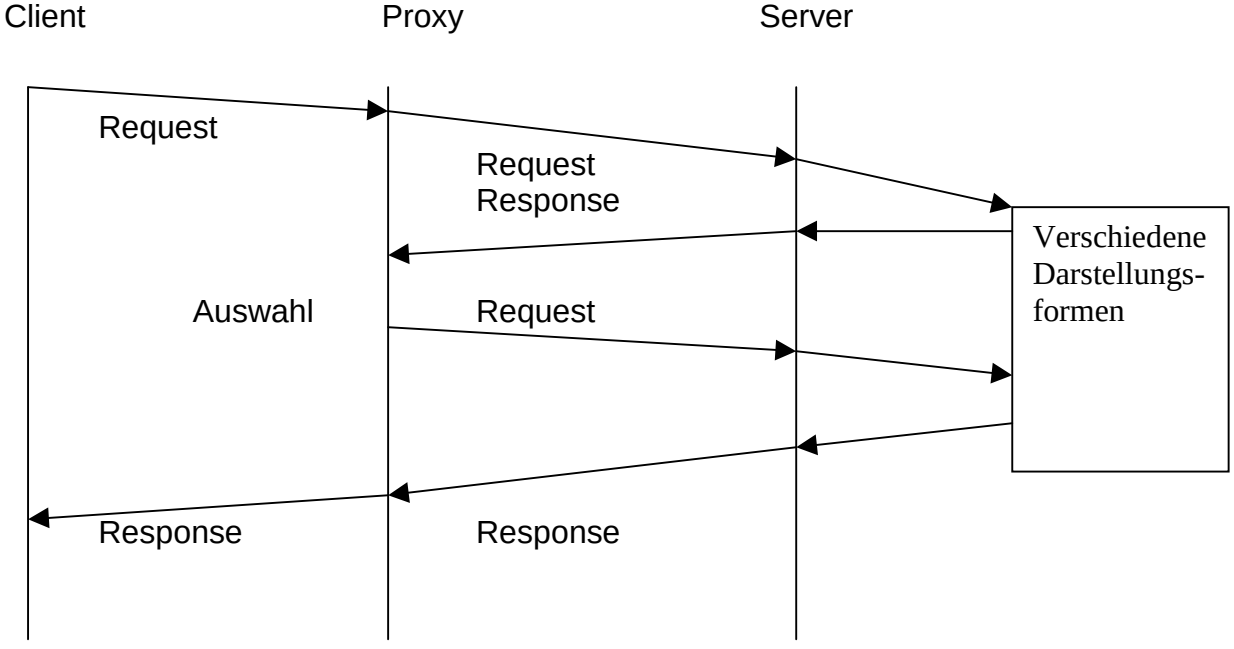
5.2 Agent-Driven Content Negotiation

Bei stark ausgelasteten Servern stellt die Server-Driven Content Negotiation eine enorme Belastung dar. Darum gibt es da noch eine andere Variante, die Agent-Driven Content Negotiation. Bei dieser Variante von Content Negotiation antwortet der Server mit einer Liste aller verfügbaren Darstellungsformen. Der Client kann dann selber entscheiden, was er denn genau will.
Ein Nachteil dieses Vorgangs ist, dass zwei Requests und Responses für eine Auswahl abgeschickt werden müssen.



Agent-Driven Content Negotiation

5.3 Transparent Negotiation



Transparent Content Negotiation

Transparent Content Negotiation stellt eine Kombination der beiden oben erwähnten Arten dar. Die Idee ist, dass der Client nur eine Request abschickt und nur eine Response erhält. Dazwischen befindet sich ein Proxy der den Rest erledigt. Dieser Rest sieht dann so aus. Der Proxy sendet das Request vom Client an den Origin Server weiter. Dieser antwortet mit einer Liste der verfügbaren Darstellungsformen. Die Response wird dann wiederum vom Proxy abgefangen, der eigenmächtig eine Auswahl trifft. Diese Auswahl bekommt dann der Client zu sehen.
Das bietet zwei wichtige Vorteile. Einerseits wird die Last zwischen den Clients verteilt, und andererseits muss der Client eben nur eine Request verschicken.

6. Zusammenfassung

HTTP ist ein wichtiges Bindeglied zwischen dem Internetnutzer und den Webseiten. Surfen im Internet in der jetzigen Form wäre ohne so ein Protokoll unvorstellbar. Es hat sichmit den Jahren weiterentwickelt und ist heute sehr weit von seiner ursprünglichen Form entfernt. Auch wenn die Ziele am Anfang Schnelligkeit und Einfachheit lauteten, so ist dem heute nicht mehr so. HTTP/1.1 gilt in Fachkreisen als unhandlich und ineffizent.
Mittlerweile arbeitet man an HTTP-ng (Next Generation). Die ganz grossen Änderungen wird das zwar nicht mehr bringen, aber es geht dabei vorallem um Verfeinerungen und immer bessere Ressourcennutzung.

- Einfachheit in den Kerngebieten zur Erhöhung der Akzeptanz.
- Erweiterbarkeit in einem verteiltem Umfeld.
- Skalierbarkeit in globalem Maßstab.
- Effiziente Nutzung der Netzwerk-Ressourcen.
- Flexibilität bei der Wahl des Transportmechanismus

Seminar
Grundlagen der Internettechnologie

HYPERTEXT MARKUP LANGUAGE <HTML>

Vortrag von Nicolas Studhalter
snicolas@ee.ethz.ch
21. Mai 2001

1. Einführung

Jeder der schon mal mit dem Internet zu tun hatte, ist der Sprache, die zum Gestalten von Web-Seiten verwendet wird, begegnet. Allerdings ist man sich dessen gar nicht bewusst, weil eine Web-Seite im Normalfall formatiert angezeigt wird, da der Browser (z.B. Netscape, Explorer, Arena) die HTML-Datei interpretiert und daraus eine formatierte Web-Seite darstellt.

HTML wurde 1990 erschaffen und wurde seither immer wieder überarbeitet, erweitert und benutzerfreundlicher gemacht. Der aktuelle Höhepunkt dieser Entwicklung stellt die bisher leistungsfähigste *HTML-Version 4.0* dar.

Zu Anfang wurden die Richtlinien von HTML definiert:

- **Leistungsfähigkeit**
  HTML sollte viele Anwendungen unterstützen, dies wurde möglich, weil HTML allgemein genug gehalten wurde. Dank dieser vielfältigen Anwendungsmöglichkeiten wurde das Ziel der Leistungsfähigkeit erreicht.
- **Einfachheit**
  Trotz der Leistungsfähigkeit soll HTML dennoch einfach zu verwenden sein, so dass viele Internet-User ermutigt werden HTML einzusetzen, und so ihren Beitrag zur Entwicklung des Internets beitragen.
- **Zugänglichkeit und Plattformunabhängigkeit**
  Da die Web-Seiten einem möglichst grossen Publikum zugänglich sein sollten, war schon anfangs klar, dass HTML plattformunabhängig sein sollte. Dies wurde durch die Konzentration auf den Inhalt realisiert.

2.Entwicklung und Geschichte

Im Mai 1989 wurde HTML am *European Labarotory for Particle Physics (CERN)*, in Genf, als Teil eines Hypermedia-System-Projekts verfasst. Im Herbst 1990 startete man die Entwicklung eines Prototyps und bis Ende des Jahres war eine erste Version fertig gestellt.
Dieser Prototyp umfasste einen zeilenorientierten und bereits einen graphischen Browser, dies ermöglichte eine Anwendung auf verschiedenen Plattformen.
Schliesslich wurde 1992 das erste Konferenzpapier über das Web vorgestellt *(Siehe Quellenangangaben: 3.)*. Obwohl diese erste HTML-Version verglichen mit der heutigen Version sehr einfach war, beinhaltete sie doch alle Basiskonzepte.

2.1 HTML 2.0

Bereits Ende 1991 entwickelte Dave Ragget (der auch an Version 4.0 mitarbeitete) eine neue verbesserte Version, HTML+. In den folgenden Monaten kamen immer mehr neue Browser auf den Markt, denen natürlich auch neue Funktionen hinzugefügt wurden. Ebenso wurde ersichtlich, dass das Web auch in Zukunft rasant wachsen würde, also entschied man sich dann für eine HTML-Version, die alle von den verschiedensten Browsern hinzugefügten Funktionen beinhaltete.
Im Juli 1994 kam dann die HTML-Version 2.0, ebenfalls 1994 wurde Netscape gegründet, das Unternehmen begann sofort mit dem Entwickeln neuer Elemente. Diese neuen Elemente

förderten zwar die Entwicklung von HTML, aber sie führten auch unweigerlich zum Veralten der damals aktuellen Version.

2.2 HTML 3.2

Mit der Gründung des World Wide Web Consortium (W3C) wurde die Empfehlung laut für HTML-Version 3.0, deren Entwurf aber nie verabschiedet wurde.
Dem bekannten Problem, dass die Browser in der Lage waren neue HTML-Funktionen zu verarbeiten, wurde 1997 mit der HTML-Version 3.2 Rechnung getragen. Die Version 3.2 verfügte unter anderem über Tabellen, Applets, Textfluss um Bilder, Sub- und Superskripte. Sie verfügte jedoch noch nicht über Frames, was wieder eine neue Version notwendig machte, da Netscape diese bereits 1995 unterstützte.

2.3 HTML 4.0

Weniger als ein Jahr später wurde dann die HTML-Version 4.0 veröffentlicht, die dann die vorgeschlagenen HTML-Funktionen beinhaltete.
Zu den neuen Funktionen, gegenüber der Version 3.2, gehörten die Internationalisierung, Unterstützung von Style Sheets, Frames, ein verbessertes Tabellenmodell, Unterstützung für die allgemeine Einbindung von Multimedia-Objekten und verbesserte Formulare.

Um zu verhindern, dass durch die Einführung der neuen Version alle älteren Web-Seiten ihre Gültigkeit verlieren, wurden mit der HTML-Version 4.0 drei SGML Document Type Definitions (DTDs) definiert. Diese DTDs können zum Interpretieren und Erstellen von HTML-Dokumenten herangezogen werden.

- **Transitional DTD**
  Die Transitional DTD dient ausschliesslich zum Interpretieren von HTML-Dokumenten. Sie enthält eine Vielzahl von Elementen und Attributen, die nicht mehr verwendet werden sollten, aber dennoch einen gültigen HTML-Code darstellen. Mit anderen Worten legt sie fest, an was sich der Browser halten sollte.
- **Strict DTD**
  In dieser DTD sind die Konstrukte beinhaltet, die der HTML-Version 4.0 entsprechen. Beim Erstellen von Web-Seiten sollte man die Strict DTD verwendet werden.
- **Frameset DTD**
  Frames werden eigentlich in verschiedenen HTML-Dokumenten definiert, die Framesets geben dabei die Struktur der verschiedenen Frames wieder. Daher spezifiziert die Frameset DTD die darin enthaltenen Dokumente.

3.0 Grundlegender Aufbau eines HTML-Dokuments

Ein HTML-Dokument besteht aus dem HEAD- und dem BODY-Teil., wobei der Document Head inhaltsfremde Informationen über das Dokument enthält, d.h. der Head dient der Beschreibung. Der eigentliche Inhalt des Dokuments wird im Document Body definiert.

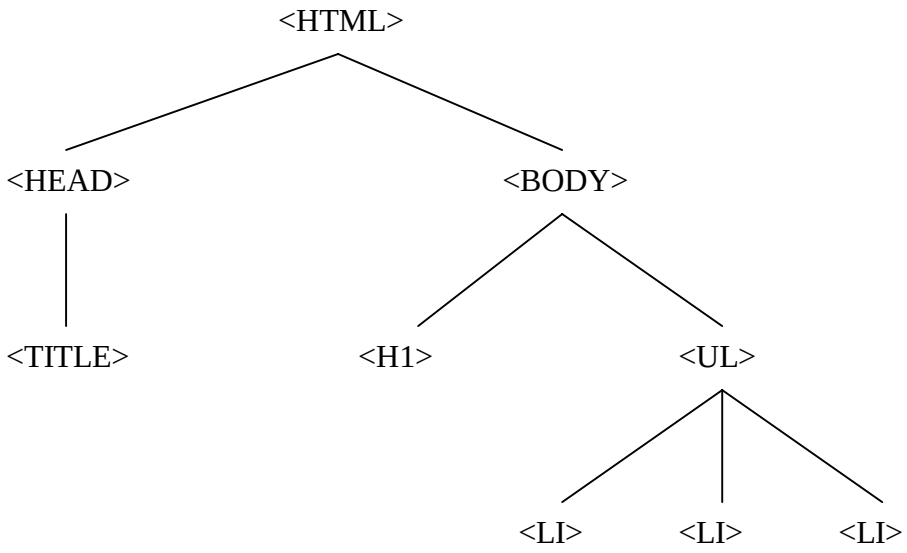


Bild 3.1: *Hier ist eine Hierarchie eines HTML-Dokuments dargestellt, es beinhaltet den Head- und den Body-Teil mit einer Überschrift <H1> und eine Liste <UL> mit drei Listenelementen <LI>.*

3.1 Der Document Head

Der Document Head enthält Informationen über das HTML-Dokument, wie z.B. den Titel, Schlüsselworte, eine Beschreibung und Style Sheets etc.

Head der Beispielseite:

```
<head>
<title>pps-Beispielseite</title>
</head>
```

Der obenstehende Document Head enthält die minimalsten Informationen. Der angegebene Titel erscheint beim Öffnen als Fenstertitel.

Ebenfalls im Head-Teil können Skripte (z.B. Java-Scripts), Style Sheets oder Beschreibungen des Dokuments, die mit Hilfe des Meta-Tags eingefügt werden, diese sind z.B. für Suchmaschinen von Bedeutung, die diese Informationen beim Suchvorgang aufrufen.

3.2 Der Document Body

Hier wir der eigentliche Inhalt des HTML-Dokuments angegeben. Der Body-Tag kann Informationen über die Seiteneigenschaften enthalten, mit ihm ist es z.B. möglich die Hintergrundfarbe (backcolor), die Textfarbe (text), die Farbe von Hyperlinks (link), die Farbe von benutzten Hyperlinks (vlink) und Hintergrund (background) festzulegen.
Beispiele sind im HTML-Code der Beispielseite *(Seite 8: Anhang B)* aufgeführt, u.a. die Textfarbe, die Hyperlinkfarbe und die Hintergrundfarbe.

4. Schlussfolgerungen

HTML hat sicher dazu beigetragen, dass das WWW so rasant gewachsen ist, oder besser gesagt, HTML hat das Web nicht daran gehindert. Den Verfassern ist es gelungen eine plattformunabhängige und leicht verständliche Beschreibungssprache zu entwickeln, die auch noch in den nächsten paar Jahren ihren Dienst tun wird.
Obwohl das 3WC gezeigt hatte, das es bereit ist auf die Forderungen der kommerziell eingestellten Browser-Hersteller einzugehen, wurden jedoch schon vor ein paar Jahren Kritiken laut, dass HTML zu wenig Funktionen erlaube, deshalb wird schon heute an einer Ablösung von HTML getüftelt. Die Alternative wird wohl XML (Extensible Markup Language) sein, die jedem User ermöglicht, seine eigene Markup-Sprache zu definieren. Mit dieser Variante wären dann wohl auch die kommerziellen Browser-Hersteller, die sich heute an den HTML 4.0 Standard zu halten haben, zufrieden.

5.0 Quellenangaben

1. E.Wilde: World Wide Web – Technische Grundlagen; Springer Verlag, 1999 Berlin Seiten 191-249
2. PPS-Seminar-Unterlagen
   http://www.tik.ethz.ch/~stiller/GIT.d.html
3. http://www.3w.org/History
4. HTML 4.0 Reference
   http://www.htmlhelp.com
5. Einführung in HTML
   http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html

Anhang A: Web-Seite (Beispiel)

pps-Beispielseite

Auf dieser Seite sind die wichtigsten Tags von HTML dargestellt

Untenstehend ist ein Beispiel für eine ungeordnete Liste

- für Textanordungen werden folgende Zeichen verwendet:

- Absatz <p>
- Zeilenwechsel
- Ausrichtung <align=left> (oder =center>, oder =right>)
- Textgrösse <font size="1-7">
- formatierte Texteingabe <pre> ... </pre>
- Hervorhebung von Text und ; Schrift fett darstellen ...; Kursivschrift ...
- Textfarbe mit font <color=white>

Beispielverweise (Links)

ein Link wird mit **a href** eingeleitet

HTML 4.0 Reference

ein Emailverweis wird mit **a href="mailto:Emailadresse"** eingelietet

E-Mail

Bilder einfügen

Bilder werden mit <img src="Name> der Bilddatei" eingefügt

Error! Unknown switch argument.

Tabellen

Feld 1	Feld 2
Feld 3	Feld 4

Anhang B: Dazugehöriger Quelltext

```
<html>
<head>
<title>pps-Beispielseite</title>
</head>

<body bgcolor="#99FFCC" text="#000099" link="#FF0033">
<br>
<h1 align="center">pps-Beispielseite</h1>
<pre>
</pre>
<p><font size="3"><b><font size="4">Auf dieser Seite sind die wichtigsten Tags
 von HTML dargestellt</font></b></font></p>
<p><b><font size="3">Untenstehend ist ein Beispiel f&uuml;r eine ungeordnete Liste</font></b></p>
<p><font size="3">- f&uuml;r Textanordungen werden folgende Zeichen verwendet:</font></p>
<ul>
 <li><font size="3">Absatz &lt;p&gt;</font></li>
 <li><font size="3">Zeilenwechsel &lt;br&gt;</font></li>
 <li><font size="3">Ausrichtung &lt;align=left&gt; (oder =center&gt;, oder =right&gt;)</font></li>
 <li><font size="3">Textgr&ouml;sse &lt;font size=&quot;1-7&quot&gt;</font></li>
 <li><font size="3">formatierte Texteingabe &lt;pre&gt; ... &lt;/pre&gt;</font></li>
 <li><font size="3">Hervorhebung von Text &lt;em&gt;und &lt;strong&gt;; Schrift
  fett darstellen &lt;b&gt;...&lt;/b&gt;; Kursivschrift &lt;i&gt;...&lt;/i&gt;</font></li>
 <li><font size="3">Textfarbe mit font &lt;color=white&gt</font></li>
</ul>
<p> </p>
<h2 align="left">Beispielverweise (Links)</h2>
<p align="center">ein Link wird mit <b>a href</b> eingeleitet</p>
<p align="center"><a href="http://www.htmlhelp.com"><b>HTML 4.0 Reference</b></a></p>
<pre>
</pre>
<p align="center">ein Emailverweis wird mit <b>a href=&quot;mailto:Emailadresse&quot;</b>
 eingelietet </p>
<p align="center"><a href="mailto:nst78@freesurf.ch"><font size="4" color="#00FF99"><b>E-Mail</b></font></a></p>
<p align="left"> </p>
<h2 align="left">Bilder einf&uuml;gen</h2>
<p align="center">Bilder werden mit &lt;img src=&quot;Name der Bilddatei&quot&gt; eingef&uuml;gt</p>
<p align="center"><img src="ethlogo.gif" width="240" height="67"></p>
<h2 align="left">Tabellen</h2>
<center>
 <table width="266" height="126" border="1">
  <tr>
  <td>
   <div align="center">Feld 1</div>
  </td>
  <td>
   <div align="center">Feld 2</div>
  </td>
 </tr>
 <tr>
  <td>Feld 3</td>
  <td>Feld 4</td>
 </tr>
</table>
</center>
</body>
</html>
```

PPS Seminar: Grundlagen der Internettechnologie

Die Datenstrukturierungssprache XML

von Thomas Mühlemann

21. Mai 2001

Was ist XML ?

XML steht für „Extensible Markup Language“ (= erweiterbare Markup Sprache). Der Name weist auf eine wichtige Eigenschaft dieser neuen Sprache hin: die Möglichkeit, eigene Tags und Attribute zu definieren. XML ermöglicht dadurch die Definition neuer Markupsprachen, z.B. ist XHTML eine mit XML redefinierte Version von HTML. XML wurde als Teilmenge von SGML (Standard Generalized Markup Language) vom W3C definiert. SGML ist ein ISO Standard und wird für eine Vielzahl von Dokumenten verwendet. Zitat aus einem W3C Faq: *"SGML is the `mother tongue', used for describing thousands of different document types in many fields of human activity, from transcriptions of ancient Irish manuscripts to the technical documentation for stealth bombers, and from patients' clinical records to musical notation."* Bemerkenswert ist, das jedes XML Dokument ein gültiges SGML Dokument darstellt.

Die mit XML definierten Markupsprachen werden als XML-Anwendungen bezeichnet. XML-Anwendungen eignen sich einerseits für die Darstellung in Web-Browsern und andererseits für die Elektronische Datenverarbeitung (in Datenbanken, Textverarbeitungen oder für das e-Commerce) und als Austauschformat zwischen Anwendungen. Dies wird durch folgendes Prinzip begünstigt: Tags in XML dienen nicht wie in HTML der Darstellung des Inhalts sondern der Definition ihrer logischen Struktur, ähnlich wie die Spalten oder Zeilenbezeichnung in einer Tabellenkalkulation. Dies ermöglicht die verlustfreie Konvertierung von Daten von und nach XML bei geeignetem Quell- rsp. Zielformat; dieses Paper wird darauf später noch eingehen, im Vordergund soll aber die Verwendung von XML als Websprache stehen.

Die Motivation, XML zu definieren

Eine neue (Web)Sprache einzuführen ist mit einem nicht unerheblichen Aufwand verbunden, zumal sich HTML als extrem dominanter Standard durchgesetzt hat (auch wenn HTML bei weitem nicht das Spektrum an Möglichkeiten bietet, das XML bieten kann). Es bestanden und bestehen erhebliche Anreize, die Einführung von XML voranzutreiben.

Zum einen ist HTML für sehr viele Anwendungen zu simpel. HTML definiert praktisch nur die Art und Weise wie Daten von einem Browser dargestellt werden können. HTML ist von dieser Seite her betrachtet einfach zu spezialisiert. Sucht man in einem HTML Dokument, das eine Namensliste enthält, zum Beispiel nach dem Namen "Schwarz" bekommt man als Resultat nicht nur den Namen "Schwarz" sondern auch "Schwarze", "Schwarzenbacher" oder "Schwarzstrasse", "Schwarzpulver" u.s.w. Natürlich kann man mit einer geeigneten Auswahl der Suchkriterien einiges erreichen (Gross- und Kleinschreibung, ganze Wörter u.s.w) aber auch die erkennen nicht den logischen Kontext, in dem das Wort steht ("Name: Schwarz, Lieblingsfarbe: Schwarz"). Bei komplexeren Suchkriterien versagt HTML endgültig (suche alle Personen mit dem Namen Schwarz, die in den 40er Jahren geboren wurden). Anstatt XML einzuführen hätte die Möglichkeit bestanden, HTML zu erweitern, so dass es teilweise die gewünschte Leistungsfähigkeit hätte, doch dies würde eine eklatante Schwächung des für viele Anwendungen sehr gut geeigneten Standards HTML bedeuten.

Wie Eingangs erklärt, ist XML (wie übrigens HTML auch) als Teilmenge von SGML definiert worden. SGML würde dem gleichen Zweck dienen, für den XML entworfen wurde, ist aber um einiges komplexer (die Spezifikation von SGML umfasst ca. 500 Seiten, die von XML nur deren 26). Dies geben die grossen Browserhersteller als Grund an, SGML nicht zu implementieren. XML stellt ein vermindertes Set an Features zur Verfügung, was die Implementierung der verarbeitenden Software erleichtert. Zum Beispiel kennen SGML wie HTML die sogenannte Markup Minimization, die es erlaubt, Endtags wegzulassen, falls diese aus dem Zusammenhang heraus ergänzt werden können. In XML gibt es keine Tags ohne Endtags (von einigen offensichtlichen Ausnahmen, den sogenannten Empty-Tags abgesehen). Dies ermöglicht die Implementierung einfacherer und dadruch leistungsfähigerer Parser.
Eine weitere Alternative zu XML wären proprietäre Datenformate, doch diese haben einen entscheidenden Nachteil: sie sind eben proprietär und somit nur einer beschränkten Gruppe von Personen zugänglich; denjenigen, die die entsprechenden Lizenzen erwerben.

Ein Code Beispiel

Um das Gesagte zu verdeutlichen und einige Technische Aspekte an XML zu erläutern, folgt nun ein Beispiel. Es ist ein Auszug aus dem XML Dokument, das die XML-Recomandation des W3C beinhaltet; darunter das "gleiche" Dokument in HTML.

```
1    <authlist>
2    <author><name>Tim Bray</name>
3    <affiliation>Textuality and Netscape</affiliation>
4    <email
5    href="mailto:tbray@textuality.com">tbray@textuality.com</email></author>
6    <author><name>Jean Paoli</name>
7    <affiliation>Microsoft</affiliation>
8    <email href="mailto:jeanpa@microsoft.com">jeanpa@microsoft.com</email></author>
9    <author><name>C. M. Sperberg-McQueen</name>
10   <affiliation>University of Illinois at Chicago</affiliation>
11   <email href="mailto:cmsmcq@uic.edu">cmsmcq@uic.edu</email></author>
12   </authlist>
13   <abstract>
14   <p>The Extensible Markup Language (XML) is a subset of
15   SGML that is completely described in this document. Its goal is to
16   enable generic SGML to be served, received, and processed on the Web
17   in the way that is now possible with HTML. XML has been designed for
18   ease of implementation and for interoperability with both SGML and
19   HTML.</p>
20   </abstract>
```

Listing 1.0

```
1    <dt>Editors:</dt>
2    <DD>Tim Bray
3     (Textuality and Netscape)
4    <A HREF='mailto:tbray@textuality.com'>&lt;tbray@textuality.com&gt;</A></DD>
5    <DD>Jean Paoli
6     (Microsoft)
7    <A HREF='mailto:jeanpa@microsoft.com'>&lt;jeanpa@microsoft.com&gt;</A></DD>
8    <DD>C. M. Sperberg-McQueen
9     (University of Illinois at Chicago)
10   <A HREF='mailto:cmsmcq@uic.edu'>&lt;cmsmcq@uic.edu&gt;</A></DD>
11   </dl>
12   <H2>Abstract</H2>
13   <P>The Extensible Markup Language (XML) is a subset of
14   SGML that is completely described in this document. Its goal is to
15   enable generic SGML to be served, received, and processed on the Web
16   in the way that is now possible with HTML. XML has been designed for
17   ease of implementation and for interoperability with both SGML and
18   HTML.</P>
```

Listing 1.1

Grundsätzlich fällt bei der Betrachtung des Codes auf, das derjenige in XML für einen Menschen auch ohne Kenntnisse der Markupsprachen sehr verständlich ist. Beim XML erkennt man klare Blöcke, die mit einem Starttag beginnen und mit einem Endtag enden. Das erste Element, man spricht bei einer durch zwei Tags eingefasste Struktur von Element, ist die Autorenliste <authlist> und beginnt in Zeile 1 und endet in Zeile 12. Das zweite Element ist der Abstract (=Zusammenfassung) <abstract> und beginnt in Zeile 13 und endet in Zeile 20. Ein Element kann nun selber weitere Elemente enthalten. Bei der Autorenliste sind das sinnigerweise die Autoren <authors>. Die wiederum "bestehen" aus einem Namen <name>, aus einer Zugehörigkeit zu einer Firma oder Organisation <affiliation> und einer e-Mail Adresse. Das <abstract> Element besteht hier nur aus einem (Unter-) Element, einem Absatz <p>. Dies alles führt zu einer leicht zu erstellenden Baumstruktur. Dies ist deshalb so einfach, weil XML korrekte Schachtelung und paarweise auftretende Tags verlangt. HTML kennt Markup Minimization die es erlaubt, viele End-Tags wegzulassen, zusätzlich sind die Browser so konzipiert, dass sie Fehler im HTML Code wie fehlende Tags und falsche Schachtelung selbsttätig korrigieren. Dies führt dazu, dass viele HTML Dokumente nicht korrekt geschrieben werden.

Weiter fällt auf, dass der XML Code seinen Elementen eine logische Bedeutung zuordnet. Ein Autor ist eben als Autor gekennzeichnet und nicht einfach als neue Zeile. Das bedeutet zum einen, dass der Browser den XML Code alleine nicht darstellen kann (er weiss ja nicht, wann er eine neue Zeile beginnen soll oder wann er etwas fett schreiben soll, dazu mehr im nächsten Kapitel), andererseits bedeutet es auch, dass Anwendungsprogramme die den XML Code bearbeiten oder lesen, viel leistungfähiger sein können. Zum Beispiel lässt sich so ganz leicht eine Liste mit allen Autoren zusammenstellen.

Aufbau und Darstellung von XML Dokumenten

Eine Markupsprache für das Web soll primär Inhalte visuell darstellen können. Wie geschieht das im Fall von XML? Auch hier zieht man am Besten einen Vergleich mit HTML heran. HTML unterstützt nur einen Dokumenttyp[1], abgekürzt DTD (Document Type Definition). Eine DTD definiert wie das Dokument aufgebaut ist – über die effektive visuelle Darstellung wird in der DTD nichts gesagt, bei HTML ist es jedoch so, dass die Darstellung anhand der DTD erfolgen kann, da die Tags die Darstellung definieren d.h. die Semantik der Elemente ist eben die Art und Weise, wie sie dargestellt werden sollen.
Die DTD definiert bei Markupsprachen unter anderem das Set an Tags das vorkommen kann. Wie gesagt, im Falle von HTML existiert *eine* DTD, das heisst es existiert *ein* fester Satz an Tags resp. Markups die in einem HTML-Dokument verwendet werden können. HTML hat diesbezüglich zwei Vorteile: die DTD ist immer bekannt, sie ist im Browser implementiert und liefert zudem Informationen über die Art und Weise wie das Dokument dargestellt wird. Das hat aber den Nachteil, dass sie unflexibel und zudem im Falle von HTML ziemlich rudimentär ist.

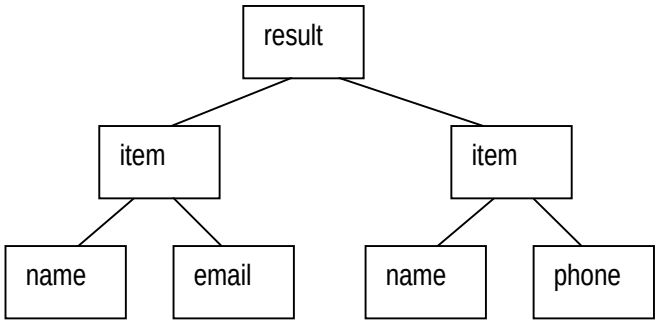
Im Falle von XML ist die DTD nicht vorgegeben. Der Author des Dokuments kann sie selber festlegen, muss sie aber im Gegenzug mit dem Dokument "mitliefern. XML Files kommen auch ohne DTDs aus, dies weil sie "well-formed", wohl geformt sein müssen. Das heisst, dass es zu jedem Tag immer ein End-Tag gibt und die Schachtelung der Tags immer korrekt ist, so dass die Struktur eines Dokuments auch erkannt werden kann, ohne die DTD zu kennen. Aufgrund dieses logischen Aufbaus, kann der Parser das Dokument analysieren; man spricht von einer "Konzentration auf den Inhalt".
Wir betrachten nun folgendes Beispiel:

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

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

<result>
<item><name>Bart</name><email>bart@thesimpsons.com</email></item>
<item><name>Lisa</name><phone>5555472</email></item>
</result>
```

Hier ist die DTD direkt in das XML-File integriert; XML lässt das zu. Der erste Block, hier nur aus einer Zeile bestehend, gibt die Version des verwendeten XML Codes an. Der folgende Block ist die DTD. Sie sagt folgendes aus: das Dokument soll ein Element namens result enthalten (das Start- und Endtag dafür wird <result> rsp. </result> sein). Dieses result-Element enthält ein oder mehrere item-Elemente. Das Element item besteht wiederum aus einem Element "name" und entweder aus einem "phone"-Element oder einem "email" Element. Ein "phone" oder "email" Element besteht wiederum aus einfachem Text ohne weitere Tags. Der dritte Block ist das Dokument selber. Alternativ zur direkten Einbindung der DTD in das Dokument kann man die DTD auch als selbsständiges File importieren. Dies ermöglicht natürlich den Austausch der gleichen DTD unter verschiedenen Partnern.

Oben wurde von XML Files ganz ohne DTD und von einer damit verbundenen "Konzentration auf den Inhalt" gesprochen. Aufgrund der Tatsache, dass XML-Dokumente well-formed sein müssten, ist eine logische Struktur, ein Baum herauslesbar; im obigen Beispiel wäre das folgender:

[1] Eigenlich sind es mittlerweile drei verschiedene DTDs. Allerdings sind diese fest definiert und in den Browsern implementiert, so dass man für die theoretische Betrachtung auch davon ausgehen kann,dass es nur eine ist.

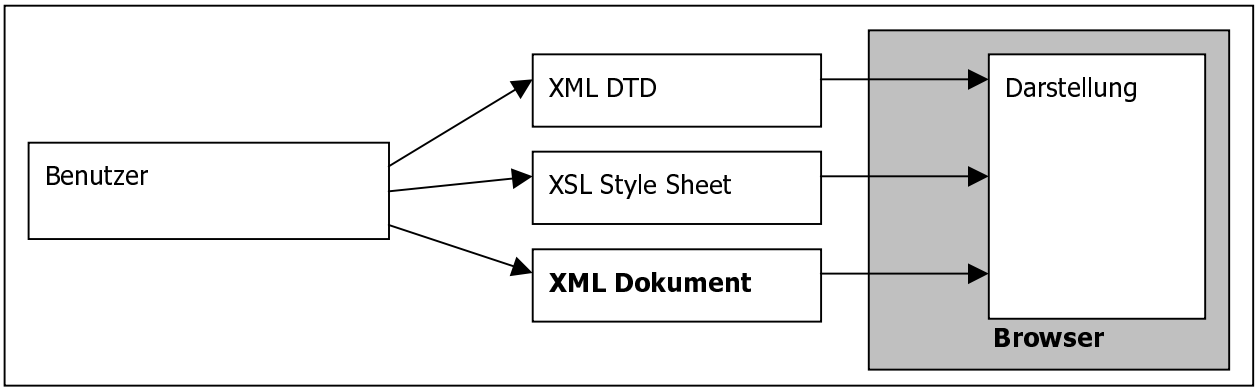


Weder DTD noch Dokument selber liefern direkt eine Information darüber, wie die Elemente dargestellt werden sollen. Dies muss eine Art Style Sheet besorgen. Für die *erweiterte* Darstellung von HTML wurde CSS (Cascade Style Sheet) entworfen (die *standardmässige* Darstellung von HTML Dokumente beruht auf der Semantik der HTML-Elemente). CSS kann zwar mit XML verwendet werden, es bietet allerdings nicht die gewünschte Leistungsfähigkeit. Für XML existiert ein spezieller Ansatz, XSL (Extensible Style Language), der ähnlich funktioniert. Für jedes Element kann eine Darstellungform definiert werden, für den oben stehen Code zum Beispiel so:

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

Dies würde zu folgender Darstellung führen: **Bart** bart@thesimpsons.com
Lisa *5555472*

Zur Verdeutlichung des vorher gesagten folgt eine Grafik welche die beschriebenen Elemente für das Webpublishing in Beziehung setzt.



Konvertierung von und nach XML

Wenn ein neues Format eingeführt werden soll, muss sichergestellt werden, dass Daten und Dateien die bereits in anderen Formaten vorliegen, nach XML umwandelbar sind. Nachfolgend eine kurze Zusammenstellung der wichtigsten Aspekte in diesem Zusammenhang:

Konvertierung eines Quellformates nach XML:
In diesem Zusammenhang interessieren vor allem zwei Fragen. Erstens, wie kann man sogenannte In-House Daten[2] nach XML umwandeln und gegebenenfalls im Web bereitstellen, und zweitens, wie kann man in HTML vorhandene Dokumente nach XML umschreiben. Beim ersten Vorgang ist die Erstellung einer geeigneten DTD das zentrale Problem, das es zu lösen gilt. Die DTD muss in geigneter Weise mit dem jeweils vorhandenen Format korrespondieren,

[2] Als "In-House Daten" werden Datenformate bezeichnet, die in der lokalen Datenverarbeitung verwendet werden und meist anwendungs- oder herstellerspezifisch sind.

so dass bei der Umwandlung keine Informationen verloren gehen und die Konvertierung möglichst ohne "Handarbeit" von statten gehen kann. Für das Publishing, falls erwünscht, muss gegebenenfalls noch eine Transformation XML nach XML (siehe unten) eingeschoben werden, da in der Regel Dokumente nicht in ihrem gesamten Umfang publiziert werden sollen, was verschiedene Gründe wie Datensicherheit oder die Beschränkung der Datenmenge haben kann. Schlussendlich bedarf es für die Webdarstellung natürlich noch eines geeigneten Style Sheets zum Beispiel in XSL.

Besonders interessant ist die Konvertierung von HTML nach XML: Ein gültiges HTML Dokument stellt kein gültiges XML Dokument dar et vice versa. Eine Lösung dieses Problem besteht in der Erstellung einer XML DTD für HTML, welche als XHTML bezeichnet wird und eine Anpassung der HTML DTD an XML darstellt. Ein erster Schritt bei der Umwandlung von HTML nach XHTML ist die "Korrektur" der HTML Dokumente, so dass sie der XML Spezifikation entsprechen. Das beinhaltet u.a. die Ergänzung fehlender Tags und die XML-konforme Festlegung von Attributwerten. Ein weiterer Aspekt ist die bereits erwähnte Tatsache, dass die meisten HTML Dokumente im Web gar nicht der HTML Spezifikation entsprechen und nur wegen toleranter Browser gelesen werden können – ein HTML nach XHTML Konverter müsste wahrscheinlich eine ähnliche Toleranz gegenüber diesen Abweichungen zeigen wie die gebräuchlichen HTML-Browser.

Konvertierung von XML in ein Zielformat:

XML wird äusserst schleppend von den Browser-Herstellern implementiert (siehe "Gegenwart und Zukunft von XML"). So ist vor allem die Konvertierung von XML nach HTML interessant. Diese ist im allgemeinen wegen der Wohlgeformtheit der XML-Dokumente einfacher als der umgekehrte Vorgang.
Die Konvertierung von XML in ein Zielformat beinhaltet auch die Konvertierung zwischen zwei XML Dokumenttypen. Dies um Daten zwischen XML basierten Anwendungen auszutauschen oder wie oben erwähnt XML Dokumente für das Publishing umzustrukturieren. Dieser Vorgang ist oft mit einem eventuell gewollten Informationsverlust vebunden. Für den Export aus XML in eine andere Markupsprache wurde vom W3C ein auf XSL basierendes Format entworfen: XSLT (das T steht für "Transformation"). Eine XSLT Datei ist in XML geschrieben und stellt im einfachsten Fall (zum Beispiel für die Konvertierung nach HTML) Regeln auf, wie die XML-Elemente in Elemente der anderen Sprache übersetzt werden. XSLT ist sehr mächtig und ermöglicht zum Beispiel auch die Verwendung benutzerdefinierter Funktionen und bedingte Verabeitung.

Links in XML Dokumenten

Da die XML-DTD frei definierbar ist, ist einem XML verarbeitenden Programm die Semantik der Elemente nicht bekannt. Um Links innerhalb eines XML-Dokuments zu verwenden, so dass diese nach aussen als solche zu erkennen sind und damit Aktionen verknüpft werden können, muss ein entsprechendes Umfeld vorliegen. Dieses XML-Umfeld wird durch XLink (XML Linking Language) definiert, die festlegt wie Links im Zusammenhang mit XML verwendet werden.

Grundsätzlich existieren verschiedene Link-Konzepte. HTML verwendet nur ein recht einfaches Link-Konzept: ein HTML-Link muss Teil einer der Ressourcen sein die verknüpft werden. Weiter kann ein HTML-Link nur zwei Ressourcen wie zum Beispiel Webpages miteinder vernküpfen und die Links sind immer unidirketional, d.h. man kann ihnen nur in einer Richtung folgen.

XLink ermöglicht viel allgemeinere Linktypen. Links in XML können, müssen aber nicht zu einer der verknüpften Ressourcen gehören, man spricht dann von Out-of-Line Links. Dies bedingt allerdings, dass diese Out-of-Line Links zuerst gefunden werden müssen, d.h. es muss ein Link auf den Out-of-Line Link existieren oder nach geeigneten Kriterien gefunden werden. Dies ermöglicht die Errichtung von bi- resp. multidirektionalen Out-of-Line Links und herausgehende Links aus schreibgeschützten Dokumenten (diese würden in einem zusätzlichen, nicht schreibgeschützen Dokument definiert und zusammen mit dem schreibgeschützen Dokument verwendet). Die Idee von XLink besteht konkret darin, Attribute zu definieren, so dass beliebige XML Elemente so behandelt werden, dass sie die von XLink geforderte Semantik besitzen.

Gegenwart und Zukunft von XML

Der XML Standard Version 1.0 ist vom W3C bereits am 10. Februar 1998 verabschiedet worden, XLink und XSL haben zurzeit den Status einer candidate recommendation resp. proposed recommendation. Es existieren bereits einige XML-Anwendungen: WML (für Online-Informationen auf kleinen Displays z.B. denen von Handys oder PDAs), XHTML, SVG

(Scalable Vector Graphics, verschiedene Hersteller wie Adobe und Quark haben die Unterstützung dieser XML-Applikation angekündigt rsp. verwirklicht), MathML (Darstellung von Mathematischen Formeln), CML (für Chemische Formeln) und viele mehr. Microsofts Internet Explorer unterstützt XML und XSL ab Version 5.0, der Netscape Navigator ab Version 6.0, eine solide Implementierung des Standards wird allerdings erst möglich sein, wenn alle wichtigen XML Komponenten vom W3C als Recommendations verabschiedet worden sind. Die komplette XML Unterstützung in Browsern wird nicht vor 2002 erwartet.
XML wird im Bereich des e-Commerce und e-Banking eine tragende Rolle übernehmen können. Wenn In-House Daten in einem XML Dokument vorliegen, muss man lediglich noch ein geeignetes Style Sheet kreieren und kann die Daten ins Web stellen. Microsofts .NET Serverreihe basiert zum Beispiel sehr stark auf XML. Überhaupt ist das Mass, indem die "Grossen" der Branche den Standart in ihre Produkte integrieren überraschend. Die steigenden Ansprüche im Bereich der moblien Kommunikation bedürfen ebenfalls flexibler Mittel für die Darstellung und Verarbeitung von Informationen. Auch hier besteht ein grosses Potential.

Resumé

XML ist zweifellos ein vielversprechendes und notwendiges Konzept. Es wird HTML und die anderen Formate im Web ergänzen wenn auch nicht ersetzen. HTML ist für viele Anwendungen, vorallem für das Publishing, ausreichend und Umstellungen sind immer mit grossem Aufwand verbunden. Und trotzdem muss es klar sein, dass das Web nach neuen, leistunsfähigeren Formaten als HTML verlangt und in Zukunft immer mehr verlangen wird. Die Aktivitäten im Internet sind schon lange nicht mehr nur auf das Publishing beschränkt. XML bietet einige zentrale Eigenschaften, die es für das Web geeignet machen:

Flexibilität	Der Benutzer kann seine eigene DTD definieren und sie in das Dokument integrieren. Darstellung, DTD und Dokument bleiben aber grundsätzlich getrennt und bieten so ein breites Spektrum an Möglichkeiten.
Plattformunabhängigkeit	XML ist nicht an eine spezielle Platform oder Software gebunden und es handelt sich bei XML Dokumenten lediglich um Plain-Text, sie können also per http ohne Probleme übertragen werden.
Freiheit	Die XML Spezifikation ist für alle frei zugänglich. Es müssen keine Lizenzen erworben werden, jeder kann XML verarbeitende Software entwickeln.
Vielschichtigkeit	XML ist nicht nur ein starkes Konzept für das Publishing es bietet sich auch als Austaschformat für jede Art von Daten an.
Solidität	XML ist vom W3C verabschiedet worden und stellt einen ISO Standard dar.

Bibliographische Angaben

E. Wilde: "World Wide Web – Technische Grundlagen"; Springer Verlag, Berlin 1999
H.Behm: "XSLT: Transformation von XML-Dokumenten"; iX, Heft 11, 1999
Verschiedene Autoren: "XML Überall"; iX, Heft 6, 2001
http://www.w3c.org

Multimediale-Unterstützung im Web

SMIL Synchronized Multimedia Integration Language

Geschrieben von
Andreas Meier
Wachtstr. 36
8134 Adliswil
andreame@student.ethz.ch

Seminarleitung
Prof. Stiller
Betreuer
Jan Gerke
Datum
28. Mai. 2001

Einleitung

SMIL ist eine Sprache zur Integration unabhängiger Multimediaobjekte in eine zeitabhängige Multimediapräsentation. SMIL wird vor allem zur Erstellung von Multimediapräsentation verwendet, die mit Hilfe der Streamingtechnologie über das Internet "gesendet" werden.

SMIL wurde von vielen Personen unter der Aufsicht der Synchronized-Multimedia-Arbeitsgruppe des Wold-Wide-Web-Konsortiums entwickelt. Besonders beteiligt waren Mitarbeiter der gemeinnützigen Institutionen INRIA Frankreich, GMD Deutschland, sowie CWI Niederlande und W3C USA mit Unterstützung von Mitarbeitern der Firmen Realnetworks, Apple, Phillips, Netscape und Lucent-Bell Labs.

Was ist SMIL

SMIL steht für **S**ynchronized **M**ultimedia **I**ntegration **L**anguage. SMIL ist HTML sehr ähnlich. Mit HTML ist es möglich, das Layout einer HTML-Seite exakt zu gestalten, Objekte unterschiedlicher Formate zu integrieren, und sowohl statische als auch dynamische Präsentationen zu erzeugen. Auch SMIL bietet diese Funktionalität, allerdings bezogen auf jede Art von Multimediaobjekt. Im Vergleich zu HTML bietet SMIL die Möglichkeit, Audio- und Videoobjekte zu integrieren, sowie die zeitliche Präsentation von Objekten genau zu steuern.

Ein HTML-Dokument wird vom Web-Server mit dem Hypertexttransferprotokoll zum Client übertragen. Das übertragene HTML-Dokument wird dabei schrittweise auf dem Bildschirm sichtbar, ohne dass der zeitliche Ablauf der Übertragung oder der Präsentation der einzelnen Objekte kontrolliert werden kann. Dies ist bei traditionellen HTML-Seiten, die aus Texten und Bildern bestehen, auch gar nicht erforderlich. Texte und Bilder einer HTML-Seite erscheinen nach ihrer vollständigen Übertragung auf dem Bildschirm. Ihre Präsentation ist an keinen festen zeitlichen Ablauf gebunden.

Dies ist bei multimedialen Präsentationen anders. Text, Ton und statische oder laufende Bilder werden in einer vorher genau definierten zeitlichen Folge präsentiert. Bei einem Film ist es wichtig, dass der Ton zusammen mit dem entsprechenden Bild wahrgenommen werden kann, usw.. Um dies zu erreichen, müssen Multimediaobjekte kontinuierlich und nach einem klar definierten zeitlichen Schema übertragen werden. Die genaue zeitliche Steuerung und Kontrolle der Übertragung ist für eine Multimediapräsentation also eine entscheidende Determinante. Um dieses Problem zu lösen, wurde SMIL entwickelt. Über die zeitliche Steuerung und Kontrolle der Übertragung und des Ablaufs eine Multimediapräsentation hinaus, kann SMIL auch zur Kontrolle des Layouts der Präsentation verwendet werden. Vereinfacht ausgedrückt: SMIL dient der Positionierung, Synchronisation und Präsentation von Multimediaobjekten.

SMIL-Präsentationen können auf einer CD oder Harddisk eines Computers abgelegt sein und bei Bedarf abgerufen werden. Die Nutzung von SMIL ist also nicht an das Internet gebunden. Die Stärke von SMIL liegt aber eindeutig bei Multimediapräsentationen im Internet.

SMIL ist nicht die einzige Technologie, mit deren Hilfe Multimediapräsentationen erstellt werden können. SMIL besitzt allerdings gegenüber anderen Multimediatechniken erhebliche Vorteile:

- Es können unterschiedlichen Datei-Formate zu einer einheitlichen Präsentation zusammengefasst werden.
- Die in einer Präsentation benutzten Multimediaobjekte müssen physikalisch nicht auf einem einzigen Server liegen.
- Multilingualität: Um beispielsweise ein Video in unterschiedlichen Sprachversionen übertragen zu können, produziert man eine Videodatei ohne Tonspur; nachträglich erstellt man Audiodateien mit den unterschiedlichen Sprachversionen; in einem SMIL-Dokument werden Video- und Audiodateien so miteinander verbunden, dass ein Nutzer automatisch die gewünschte Sprachversion des Videos erhält.
- SMIL unterstützt unterschiedliche Bandbreiten; auf diese Weise ist es möglich, die Übertragung ein- und derselben Version einer Multimediapräsentation an die Bandbreite des Nutzers anzupassen.
- Präsentationen können so gestaltet werden, dass sie sich automatisch an die Gegebenheiten des Browsers oder Players des Nutzers anpassen.
- Es werden keine teuren Autorenwerkzeuge gebraucht. Eine SMIL-Präsentation kann in jedem beliebigem Text-Editor erstellt werden.

Syntax

SMIL basiert auf XML. Die Tags sind ‚case sensitive' und müssen allesamt abgeschlossen werden.

Für die Abfassung eines SMIL-Dokuments gelten einige allgemeine Regeln:

- Jedes Dokument startet mit <smil> und endet mit </smil>.
- Es besteht aus einem head-Teil (optional) und einem body-Teil.
- Tags und Attribute müssen in Kleinbuchstaben geschrieben werden.
- Attributwerte sind immer in Anführungszeichen zu setzen.
- SMIL-Dateien werden mit dem Suffix .smi oder .smil identifizierbar gemacht.
- Kommentare können mit <!-- --> eingefügt werden.
- Das Einrücken von Code-Zeilen in ist nicht obligatorisch. Allerdings wird das Einrücken von Code-Zeilen zur besseren Lesbarkeit empfohlen.

Layout

Das Layout einer Multimediapräsentation wird im Layout-Abschnitt des SMIL Dokuments mit dem <layout>-Element definiert. Grundsätzlich kann die Layoutdefinition mit Cascading-Style-Sheets-Elementen (CSS) vorgenommen werden. Der Layout-Abschnitt eines SMIL-Dokuments befindet sich in dessen head-Teil. Innerhalb des <layout>-Elements werden alle layoutspezifischen Definitionen getroffen. Da der head-Teil eines SMIL-Dokuments optional ist, ist auch die Definition eines Layouts in SMIL optional. Wird kein Layout definiert, werden die

Multimediaobjekte in ihrer Originalgröße präsentiert. Bei einer reinen Audiopräsentationen entfällt die Layoutdefinition grundsätzlich.

```
<head>
  <layout>
    <root-layout width="800" height="600" background-color="blue" title=”Surf”/>
    <region id="_full" left="0" top="0" width="100%" height="100%" z-index="1" fit="fill"/>
    <region id="_leftup" left="0" top="0" width="60%" height="60%" z-index="3" fit="meet"/>
    <region id="_rightdown" left="40%" top="40%" width="60%" height="60%" z-index="2" fit="slice"/>
  </layout>
</head>
```

<root-layout>

Mit diesem Tag werden grundlegende Parameter des Layouts definiert. (Grösse, Hintergrundfarbe, Titel)

<region>

Nachdem mit <root-layout> festgelegt wurde, gilt es nun in einem zweiten Schritt einzelne Präsentationsbereiche zu definieren, in denen die jeweiligen Multimediaobjekte sichtbar werden sollen. Mit dem <region>-Element kann die Position und Größe eines Präsentationsbereiches sowie die Skalierung eines Multimediaobjekts festgelegt werden.

id: Mittels diesem Schlüssel wird das Präsentationsfenster referenziert.

left/top: Linke obere Ecke des Präsentationsfensters.

width/height: Grösse des Fensters.

z-index: Wenn sich zwei Fenster überlappen, erscheint das Fenster mit kleinerem Index oben.

fit: „fill“: Das Objekt füllt das ganze Fenster aus. (Falls nötigt gestreckt und verzerrt)
„meet“: Das Objekt wird proportional nur soweit vergrössert, bis der linke oder untere Rand des Fensters erreicht ist.
„slice“: Es wird soviel von Objekt angezeigt, wie im Fenster platzt hat.

Medienelemente

Mit SMIL lassen sich die unterschiedlichsten Multimediaobjekte zu einer einheitlichen Präsentation zusammenfassen. Es lassen sich folgende Multimediaobjekte integrieren:

Objekt	**Element**	**Format**
Bild	<img />	gif, jpeg
Text	<text />	txt
Realtext-Textstrom	<textstream />	rt
Audio	<audio />	rm, wav, aif, mov, mp3
Video	<video />	rm, avi, mov, asf, viv, mpeg
Animation	<animation />	swf

Die Integration in die Präsentation ist sehr einfach.

```
<body>
  <img src="pic64.jpg" region="_full" dur="4s"/>
</body>
```

src: Die Referenzierung des Objektes. Sie kann absolut oder auch realtiv sein. Genau gleich wie bei HTML.

region: In diesem Präsentationsbreich wird das Bild angezeigt.

dur: Das Bild wird 4 s lang angezeigt

Zeitliche Steuerung

In einer Präsentation will man verschiedene Objekte darstellen. Dafür muss deren zeitlichen Ablauf genau geregelt werden.

Zeitliche Folge von Objekten – das <seq>-Element

Mit dem <seq> Element können die Elemente nacheinander dargestellt werden. Falls kein Synchronisationselement angegeben wird, gilt automatisch das <seq>-Element.

```
<seq>
  <img src="pic14.jpg" region="_full" dur="5s" begin="0s"/>
  <text src="text1.txt" region="_full" dur="3s" begin="2s"/>
</seq>
```

In diesem Beispiel wird zuerst für 5s ein Bild angezeigt, und nach 2s Pause wird für 3s ein Text eingeblendet.

Zeitliche Parallelität von Objekten – das <par>-Element

Mit diesem Tag können die Elemente zeitgleich angezeigt werden.

```
<par>
  <img src="pic7.jpg" region="_rightdown" dur="5s" begin="0s"/>
  <img src="pic47.jpg" region="_leftup" dur="6s" begin="3s"/>
</par>
```

In diesem Beispiel wird rechts unten für 5s ein Bild eingeblendet. Das zweite Bild wird 3s nach dem ersten Bild links oben angezeigt. Somit sind danach noch für 2s beide Bilder sichtbar. Dann verschwindet das erste und das zweite wird noch für 4s angezeigt.

Begin, end und repeat als Attribut eines Synchronisations-elements

Sowohl beim <par>-Element als auch beim <seq>-Element kann die Präsentationszeit mit dem begin-, end- und repeat-Attribute kontrolliert werden.

```
<par begin="2s" end="8s" repeat=”2”>
  .....
</par>
```

Hier wird die Einblendung aller im <par>-Element eingeschlossenen Multimediaobjekte nach 2 Sekunden gestartet; 8 Sekunden nach der Einblendung werden die Objekte wieder ausgeblendet. Das ganze wird zweimal wiederholt.

Kombination von <seg>- und <par>-Elemente

Beide Elemente können nach Bedarf kombiniert und verschachtelt werden. Hier ein kleines Beispiel das die oben besprochenen Elemente zeigt.

```
<smil>
<head>
   <meta name="copyright" content="Andreas Meier" />
   <layout>
      <root-layout width="800" height="600" background-color="blue" />
      <region id="_full" left="0" top="0" width="100%" height="100%" z-index="1" fit="fill"/>
      <region id="_leftup" left="0" top="0" width="60%" height="60%" z-index="3" fit="meet"/>
      <region id="_rightdown" left="40%" top="40%" width="60%" height="60%" z-index="2" fit="fill"/>
   </layout>
</head>
<body>
   <par dur=”35s”>
      <audio src="sound1.mp3" begin="1s"/>
      <seq>
         <img src="pic1.jpg" region="_full" dur="7s" begin="0s"/>
         <par>
            <img src="pic2.jpg" region="_rightdown" dur="9s" begin="1s"/>
            <text src="text1.txt" region="_leftup" dur="8s" begin="4s"/>
         </par>
         <img src="pic3.jpg" region="_full" dur="7s"/>
      </seq>
   </par>
</body>
</smil>
```

Mit dem ersten <par> Tag wird erreicht, das während der ganzen Präsentation Musik läuft. Als erstes wird das pic1 angezeigt. Eine Sekunde später fängt die Musik an zu spielen. Danach wird ein Bild zusammen mit Text dargestellt (parallel). Und zum Schluss wird pic3 angezeigt.

Systemabhängige Darstellung

Die Wahlmöglichkeiten auf der Basis entsprechender Testattribute sind ziemlich umfangreich. Für eine solche Unterscheidung wird der <switch>-Tag gebraucht. Die folgenden Beispiele sollen einige Wahlmöglichkeiten demonstrieren.

Wahl der optimalen Übertragungsgeschwindigkeit

```
<par>
  <text .../>
  <switch>
    <par system-bitrate="40000">
    ...
    </par>
    <par system-bitrate="24000">
    ...
    </par>
  </switch>
</par>
```

Wahl unterschiedlicher Audioquellen je nach Übertragungsgeschwindkeit

```
<switch>
   <audio src="bessere-audioversion" system-bitrate="16000" />
   <audio src="audioversion" system-bitrate="8000" />
</switch>
```

Wahl der Sprachversion in Abhängkeit von der Sprache des Betriebssystems

```
<switch>
```

```
  <audio src="audio-französich" system-language="fr" />
  <audio src="audio-deutsch" system-language="de" />
</switch>
```

Wahl der Bildschirmgröße und –tiefe in Abhängigkeit von den technischen Gegebenheiten des Nutzers

```
<par>
 <switch>
  <par system-screen-size="1280X1024" system-screen-depth="16">
   ….
  </par>
  <par system-screen-size="640X480" system-screen-depth="32">
   ….
  </par>
  <par system-screen-size="640X480" system-screen-depth="16">
   ….
  </par>
 </switch>
</par>
```

Zusammenfassung

SMIL ist ein mächtiges Mittel um im Web multimediale Präsentationen zu gestalten. Mit der angekündigten Integration im Internet Explorer 6, werden vielleicht auch die Webdesigner auf diesen relativ neuen Standart aufmerksam. Mal schauen, ob sich SMIL im weltweiten Datennetz durchsetzten kann.

Montag, 11 Juni 2001

Sichere Kommunikation

Michael Casty

PPS Seminar: Grundlagen der Internet-Technologie

1. Einleitung

Immer mehr konventionelle Produkte werden von den verschiedensten Herstellern auf dem Internet angeboten. Möchte man nun als Kunde dieses Angebot nützen, so benötigt man oft nur eine Kreditkarte, um mit ihrer Nummer das verlangte Entgelt zu entrichten. Nun muss nur noch ein kurzes Formular ausgefüllt werden, in dem unter anderem die Adresse angegeben wird, an welche die Ware zugestellt werden soll. Nichts einfacheres als das!
Doch was wäre, wenn der bestellte und bezahlte Artikel nicht ankommt, weil das Unternehmen, dessen Webseite man besuchte, in Wirklichkeit gar nicht existiert. Wenn auf dem Kontoauszug bemerkt wird, dass mehr als erwartet mit der Kreditkarte bezahlt wurde, weil die über das Internet übertragene Kreditkartennummer von einem Mithörer abgefangen wurde und er diese illegaler weise für seine eigenen Einkäufe verwendete. Oder wenn ein Mithörer beim übertragenen Formular seine eigene Adresse eintrug und nun ihm statt mir die bezahlte Ware zugestellt wird.
Wer sichergehen will, dass die Kommunikation mit anderen Anwendern im internen wie im externen Netzwerk nicht belauscht wird, hat ein Problem, denn die herkömmlichen Internetprotokolle wie E-Mail weisen schwerwiegende Sicherheitsrisiken auf. Fast sämtliche Kommunikation im Netz erfolgt im Klartext. Unter anderem werden auch Benutzername und Passwort z.B. beim Abholen der E-Mails vom Server mittels POP3 unverschlüsselt übertragen.
Voraussetzungen für sichere Kommunikation sind:
Vertraulichkeit: Nur ein bestimmter Personenkreis darf den Inhalt der Daten erfahren, sie müssen also verschlüsselt werden.
Authentizität: Das heisst, beide Kommunikationspartner können zu jeder Zeit sicher sein, mit wem sie wirklich kommunizieren. Die Möglichkeit, dass die Kommunikation über einen Mittelsmann läuft, welcher die Daten verändern kann, ist ausgeschlossen, da die Herkunft der Daten zweifelsfrei bestimmbar ist.
Integrität: Die Daten müssen vor Veränderung geschützt sein. Dabei kann man natürlich nicht verhindern, dass die Daten verfälscht werden. Man kann jedoch Prüfsummen einsetzen, um die Veränderung sicher feststellen zu können.
Nichtabstreitbarkeit: Sie bedeutet, dass die Herkunft der Daten auch gegenüber Dritten eindeutig belegbar ist. Sie setzt zwingend ein Public Key Verfahren voraus.

2. Krypthografische Verfahren

2.1 Klassifizierung

Ein starkes Verfahren lässt sich aufgrund der zur Verfügung stehenden Rechenleistung nicht brechen, schwache Verfahren dagegen schon. Ein gutes bezieht seine Sicherheit lediglich aus der Unkenntnis des Schlüssels, der Algorithmus dagegen wird als bekannt vorausgesetzt. Wird der Algorithmus eines Verfahrens geheimgehalten, so liegt der Verdacht nahe, dass ein schwaches Verfahren vorliegt.
Als Mass für die Sicherheit wird allgemein die Schlüssellänge angesehen, wobei mindestens zwischen symmetrischen und asymmetrischen Verfahren unterschieden werden muss, denn ein symmetrischer Schlüssel mit 80 Bit ist etwas leichter zu entschlüsseln als ein asymmetrischer mit 1024 Bit;

2.2 Symmetrische Verfahren (Secret oder Private Key Verfahren)

Sender und Empfänger teilen sich einen gemeinsamen Schlüssel, der notwendigerweise geheim zu halten ist.

Blockchiffren

Bei dieser Methode werden einzelne Textblöcke unabhängig voneinander verschlüsselt. Sie durchlaufen mehrere Verschlüsselungsrunden. Kernfunktionen darin sind die Substitution (Teilblöcke werden durch andere Bitfolgen ersetzt) und die Permutation (vertauscht die Position der einzelnen Bits).
Ein Beispiel dafür ist das in den 70er Jahren entwickelte DES (56 Bit Schlüssel). Heute steht das Tripple DES (3-DES) zur Verfügung (112Bit)

Stromchiffren

Stromchiffren arbeiten mit einzelnen Bits oder einem Byte. Gleiche Textstücke werden unterschiedlich chiffriert, da auch die Position innerhalb des Textes eine Rolle spielt.
Eine geheime Pseudozufallszahlenfolge (abhängig vom Schlüssel) und die Bits des zu verschlüsselten Textes werden durch eine einfache Funktion verknüpft (z.B. XOR).
Zur Entschlüsselung wird die inverse Funktion (hier ebenfalls XOR) und die geheime Zahlenfolge benötigt. Beispiel: A5 (54Bit) wird vom GSM Mobilfunknetz verwendet.

2.3 Asymmetrische Verfahren (Public Key Verfahren)

Erst seit 1982 stehen brauchbare Verfahren zur Verfügung. Zur Ver- bzw. Entschlüsselung werden unterschiedliche Schlüssel gebraucht, welche in einem komplizierten mathematischen Zusammenhang stehen, voneinander aber nicht abgeleitet werden können. Im Vergleich zu symmetrischen sind die asymmetrischen Verfahren sehr rechenaufwendig.
Der Schlüssel zur Verschlüsselung kann veröffentlicht werden. Nachrichten welche mit diesem Schlüssel verschlüsselt sind, können so nur vom Besitzer des Private Key's entschlüsselt werden.
Es gibt jedoch auch Anwendungen, welche einen geheimen **Ver**schlüsselungs-Schlüssel verwenden. Der Sinn davon ist nicht, eine Nachricht unlesbar für dritte zu machen, da sie ja jeder mit dem öffentlichen Entschlüsselungs-Schlüssel entschlüsseln kann, sondern die Identität des Verfassers zu beweisen, welcher ja der einzige ist, der die Nachricht so verschlüsseln kann.

RSA

Das am weitesten verbreitete Verfahren ist das RSA. Der öffentliche und private Schlüssel stehen wie schon gesagt in einem komplizierten mathematischen Zusammenhang. Ich will hier nur das Prinzip erläutern:
N: Produkt zweier Primzahlen (mind. 1024Bit)
Der öffentliche Schlüssel: C(N,d) = M^e (mod N)
Der geheime Schlüssel: M(N,d) = C^d (mod N)
N, d und e stehen dabei natürlich in einer gewissen Beziehung:
X=X^(e*d) (mod N) für X<N, X ganze Zahl.
Die Berechnung des geheimen Exponenten d aus e ist nur möglich, wenn die Primzahlenzerlegung von N bekannt ist.

2.3.1 Hashfunktionen

Die Hashfunktion bekommt einen Text als Input und liefert als Output eine Ausgabe fester Länge, üblicherweise 128 oder 160 Bit. Die Funktion ist nicht umkehrbar.
Ändert sich in der Eingabe nur ein Bit, so ändern sich im Hashwert etwa die Hälfte der Bits. Deshalb bezeichnet man den Hashwert auch als Fingerabdruck einer Eingabe. Anhand dieses Hashwertes lassen sich zwei Dateien miteinander vergleichen. Stimmen die Werte überein, so sind die Dateien mit überwältigender Wahrscheinlichkeit gleich.
Bekannte Funktionen sind MD5(message digest 5) und SHA-1(secure hash algorithm).

2.3.2 Digitale Signaturen

Wie schon erwähnt, kann ein geheimer Verschlüsselungsschlüssel dazu verwendet werden, die Identität des Verfassers zu ermitteln. Beim Entschlüsseln mit dem public Key lässt sich also zeigen, wer die Daten verschlüsselt hat.

Die abgesetzte Signatur

Dabei wird nur der Hashwert des Textes verschlüsselt. Diese Signatur wird dem Empfänger übermittelt, welcher mit dem Public-Key die Nachricht und den Hashwert wieder entschlüsseln kann. Er errechnet aus dem erhaltenen Text mit der gleichen Funktion wie der Absender den Hashwert und vergleicht diesen mit dem, welcher er mit der Nachricht bekommen hat. Stimmen diese Werte überein, so ist der Absender eindeutig identifiziert und es steht fest, dass das Dokument auf dem Weg nicht verändert wurde.

2.3.3 Zertifikate

Ein Zertifikat stellt die Verbindung zwischen einem öffentlichen Schlüssel und einer Identität her. Diese Zusammengehörigkeit wird durch eine digitale Signatur einer Zertifizierungsstelle bestätigt. Der Aussteller eines Zertifikates bestätigt also, dass ein bestimmter Schlüssel dieser Identität gehört. In einem so stark vermaschten Netz wie das Internet gibt es aber keine Zertifizierungsstellen im eigentlichen Sinne. Vielmehr entscheidet jeder Nutzer selber, wem er ein Zertifikat ausstellen will. Umgekehrt kann auch jeder Nutzer selbst entscheiden, wessen Zertifikate er vertraut. So kann sich ein Nutzer A über zum Beispiel einem ihm unbekannten Nutzer C vertrauen, wenn die beiden einen gemeinsamen Bekannten B haben, welcher wiederum beiden vertraut.
Ein Zertifikat ist also immer nur so vertrauenswürdig wie sein Aussteller.

3. Secure Sockets Layer (SSL)

Die IETF (Internet Engeneering Task Force) führt den Standardisierungsprozess unter dem Nahmen TLS, Transport Layer Security)

Mit dem Ziel, eine sichere Kommunikation über ein unsicheres Medium zu ermöglichen, definiert SSL ein Protokoll, das eine Verbindungssicherheit bereitstellt, die 3 grundlegende Eigenschaften besitzt.

Verbindungssicherheit:

Nach einem anfänglichen sogenannten Handshake wird mit Hilfe eines Verschlüsselungsverfahren ein geheimer Schlüssel definiert. Die Datenverschlüsselung verwendet ein symmetrisches Verfahren.

Optionale Authentifizierung:

Die Identität des Kommunikationspartners kann mit Hilfe von asymmetrischen Verfahren authentifiziert werden.
Es lassen sich drei Stufen unterscheiden: Anonymität, Server-Authentifizierung (nur der Server muss ein vom Client akzeptiertes Zertifikat vorweisen) oder Authentifizierung beider Parteien.

Zuverlässigkeit einer Verbindung

Die Nachrichtenübertragung schliesst eine Integritätsprüfung (unter Verwendung sicherer Hash Funktionen) ein.

3.1 Ziele von SSL

Kryptografische Sicherheit

SSL soll dazu verwendet werden, eine für dritte nicht einsehbare Verbindung zwischen zwei Stellen aufzubauen

Interoperabilität

Unabhängige Programmierer sollen SSL einsetzende Anwendungen entwickeln können, die in der Lage sind, Verschlüsselungsparameter auszutauschen, ohne jeweils den Code der anderen zu kennen.

Erweiterbarkeit

SSL versucht einen Rahmen bereitzustellen, innerhalb dessen sich neue Verfahren den Erfordernissen entsprechend kombinieren lassen.

Relative Wirksamkeit

Da die asymmetrischen Verfahren besonders rechenaufwendig sind, enthält SSL ein Schema zum Caching der Sitzungen, um die Anzahl der von Grund neu aufzubauenden Verbindungen niedrig zu halten.

3.2 Verbindungsaufbau

Die Verbindungsaufnahme initiiert prinzipiell immer der Client mit einer Hello-Nachricht. Sie dient dazu, dem Server mitzuteilen, welche Verfahren bekannt sind. Der Server wählt davon die stärkste ihm bekannte aus und sendet dies in der Server-Hello-Nachricht zurück. Der Sitzungsschlüssel errechnet sich nun aus der ClientKeyExchange Nachricht und den beiden Hello-Nachrichten, welche mit dem öffentlichen Schlüssel des Servers verschlüsselt ist. Die Authentizität des Servers ist dadurch gewährleistet, dass er die ClientKeyExchange Nachricht entschlüsseln kann. Die des Clients wird mit der CertificateVerify Nachricht sichergestellt. Die ChangeCipherSpec Nachricht schliesst die Aushandlungen ab und überprüft deren Erfolg.

3.3 Kombinieren von HTTP und SSL, HTTPS

Da der Client muss wissen, dass er anstelle einer TCP eine SSL Verbindung zum Server aufbauen muss. Das wird dadurch erreicht, dass der URL den Präfix „https“ anstelle von „http“ verwendet.
Leider ist SSL bis heute in den wenigsten Betriebssystemen als standardmässige Transportprotokollschicht implementiert. Darum muss jedes Anwendungsprogramm (z.B. ein Browser) SSL selber implementieren.

3.4 Secure HTTP (S-HTTP)

S-HTTP stellt eine Alternative zu HTTPS dar. Obwohl die Verschlüsselungsfähigkeiten der beiden Protokolle ähnlich sind, ist S-HTTP weniger verbreitet.
S-HTTP definiert ein auf HTTP basierendes Nachrichtenformat. Dieses erweitert HTTP durch Sicherheitsmerkmale, wie Authentifizierung, verschlüsselte Datenübertragung sowie Verhandlungsoptionen zwischen Client und Server. Eine S-HTTP-Nachricht besteht aus einer gekapselten HTTP-Nachricht und einigen vorangestellten Kopfzeilen, die das Format der gekapselten Daten beschreiben. Beide Seiten können im Rahmen einer Verhandlung Angaben über die verwendbaren beziehungsweise geforderten Erweiterungen gegenüber HTTP machen. Dazu gehören: Nachrichtenformate, Typen der Zertifikate, Schlüsselaustauschmechanismus, Verfahren für digitale Unterschriften, Hash-Algorithmus sowie Verschlüsselungsverfahren für Kopf und Inhalt.
S-HTTP unterstützt zwei kryptografische Standardformate:

MIME Object Security Services (MOSS) und
Cryptographic Message Syntax (CMS).
Es ist jedoch problemlos möglich, andere Formate in S-HTTP aufzunehmen.

5. Secure-Shell (SSH)

SSH wurde 1996 von Tatu Ylonen entwickelt. Die einfache Benutzung und Administration machten SSH zu einem der meistgenutzten Kryptographieprogramme. 1998 kam die Version 2.x auf den Markt. Sie ist fast komplett neu geschrieben, schneller, sicherer und flexibler bezüglich Public-Key Authentifizierung (je nach Einstellung müssen mehrere Authentifizierungs-Prüfungen bestanden werden). Als Neuigkeit bietet SSH2 auf Servern je nach Schlüssel einstellbare Befugnisse. Traut man z.B. der Aufbewahrung eines Schlüssels nicht sonderlich, so lässt sich seine Nutzung so einschränken, dass nur die Nutzung eines bestimmten Programms möglich ist.
SSH2 bietet zusätzlich auch einen sftpd (Secure FTP Daemon), welcher eine gesicherte FTP Verbindung garantiert.
Wer zwar die neue Version nutzen, Clients und Server der alten Version jedoch trotzdem bedienen will, muss auch beide Versionen installieren. Fragt nun ein SSH1-Client nach einem Login, so reicht dieser die Anfrage an einen Server der alten Generation weiter.
Die Nutzung von SSH2 ist durch seine gegenüber dem Vorgänger sehr restriktiven Lizenzbedingungen stark eingeschränkt. Die Nutzung von SSH1 ist sogar Firmen erlaubt, solange es nur dem operativen Geschäft dient. Bei SSH2 ist die freie Nutzung ausschliesslich in Bildungseinrichtungen zulässig, in keinem Falle innerhalb eines Unternehmens.
Nun hat sich eine Gruppe von Entwicklern daran gemacht, eine freie SSH Version zu entwickeln. Die momentan erhältliche Version von Open-SSH unterstützt erst die Version 1.x des Protokolls.

5.1 Verbindungsaufbau

1)Der Client initiiert eine SSH Sitzung
2)Der Server startet einen neuen SSH Daemon und generiert ein Schlüsselpaar (SK publ,SK sec)
3)Nun tauschen Client und Server die Protokollversionen aus. Dabei ist nur ein Fall problematisch: Benutzt der Client Version 2, der Server die alte Version, so lehnt dieser die Kommunikation ab.
4)Nun sendet der Server den HK publ und den SK publ. Der Client vergleicht HK publ mit dem Eintrag in seiner Liste der bekannten Server. (Kennt er ihn noch nicht, so kann er ihn „lernen“. Dabei ist allerdings Vorsicht geboten, es könnte sich auch um einen anderen als um den vermeintlichen handeln).
5)Hat der Client den HK publ des Servers akzeptiert, so erzeugt er einen Sitzungsschlüssel und schickt ihn verschlüsselt mit HK publ und SK publ an diesen zurück. Er teilt dem Server auch mit, welches symmetrisches Verfahren er anwenden möchte.
6)Der Server kann den Sitzungsschlüssel mit SK sec und HK sec wieder entschlüsseln.
7)Die Kommunikation verläuft von nun an verschlüsselt.
8)Nun muss sich der Client ausweisen(z.B. durch Passwort-Authentifiierung und oder durch Authentifizierung mittels eines Public-Key Verfahrens)
9)Ungefähr jede Stunde oder nach einem Gigabyte gesendeter Daten wird der Sitzungsschlüssel gewechselt.

6. PGP: Pretty Good Privacy

PGP ermöglicht es, verschlüsselte Emails zu versenden. Es implementiert auch Funktionen zur Verarbeitung von PGP-Paketen für andere Anwendungen.
Der Typische Ablauf einer Email Übertragung mittels PGP:
1)Der Sender signiert das zu versendende Dokument
2)Nun wird das Dokument und die Signatur komprimiert
3)Danach wird ein geheimer Sitzungsschlüssel generiert und damit das komprimierte Dokument verschlüsselt.
4)Der Schlüssel wird mit dem öffentlichen Schlüssel des Empfängers chiffriert.
5)Das ganze wird nun noch in ASCII Zeichen umgewandelt, damit es mit Email verschickt werden kann.
Der Empfänger führt die Schritte in umgekehrter Reihenfolge aus.

7. Zusammenfassung, Schlussfolgerungen

Die wichtigsten Merkmale einer kryptographisch gesicherten Verbindung sind gesicherte Vertraulichkeit, Authentizität und Integrität.
Die ersten Kommunikationsschritte der meisten Verschlüsselungsverfahren sind mit einem rechenaufwendigen asymmetrischen Verfahren verschlüsselt. Danach läuft die Kommunikation über ein weniger aufwendiges, symmetrisches Verfahren. Hashwerte überprüfen, ob die Daten auf dem Übertragungsweg verändert wurden, Zertifikate (Digitale Signaturen) überprüfen die Authentizität.
Ich habe euch für die sichere Übertragung von http Nachrichten das s-http und https, welches auf SSL basiert, vorgestellt. Ein etwas komplexeres Programm ist das SSH, welches je nach Einstellung mehrere Sicherheitsprüfungen vornimmt. Für einigermassen sichere Emails reicht allerdings auch das schon etwas veraltete PGP.
Mit zunehmender Rechenleistung sinkt auch die Sicherheit der verschiedenen Verschlüsselungsmethoden. So ist zum Beispiel schon eine ursprüngliche Version von SSL teilweise geknackt worden, da die USA aus Militärtechnischen Überlegungen die Ausfuhr des Programms mit Schlüssellängen über 40Bit verboten hatten. Mit steigender Rechenleistung müssen daher auch die Algorithmen komplizierter und die Schlüssel länger werden. Das ganze hilft aber natürlich nichts, wenn es möglich ist, jemandem eine falsche Identität vorzugaukeln. Daher ist vor allem bei der Vergabe von Zertifikaten grosse Vorsicht angebracht.

Internet Telephonie

Autor:
Olivier Gillieron

Einführung

Die Idee, das Internet als Transportmedium für die Telefonie zu nutzen, wurde erstmals von der israelischen Firma VocalTec im Jahre 1995 auf den Markt gebracht. Die Technik arbeitete zunächst nur halbduplex. (Heutige Verfahren sind vollduplex und auch für konventionelle Telefonanlagen ausgelegt.) 1996 schien dann die IP-Telefonie der nächste grosse Schub für das Internet zu sein. Der damalige Netscape-Chef Jim Clark sprach sogar von der Abschaffung des klassischen Telefons. Ein Jahr später hiess es: Die Experten diskutieren nicht mehr darüber, ob es sinnvoll ist Sprache durch das Internet zu leiten, sondern welcher Prozentsatz des Telefonsverkehrs demnächst auf diesem Weg laufen wird. Andere prophezeiten die völlige Verschmelzung von Telefonnetz und Internet.

Hauptvorteil für Nutzer der IP-Telefonie ist eine signifikante Kostensenkung insbesondere bei Ferngesprächen, da für die Telefonverbindung von und zu den Einwahlknoten in der Regel lediglich Ortstarife anfallen. Doch es gibt auch schwerwiegende Nachteile gegenüber dem normalen Telefonieren.

- Der Gesprächspartner muss ebenfalls online sein
- Man muss die Adresse kennen, unter der die jeweilige Person erreichbar ist
- Für jedes Telefon müsste zuerst der eigene PC-hochgefahren werden
- Beim Gesprächspartner fallen ebenfalls kosten (zum Ortstarif) an

Angesichts dieser Nachteile beschränkte sich die Internet-Telefonie anfänglich auf eine Nische für Online-Spieler. Inzwischen haben aber die Grossen der Daten- und Telecom-Branche sich der Technik angenommen. Unter dem Oberbegriff Voice over IP (VoIP), versuchen sie neue Geschäftsfelder zu erschliessen.

VoIP Anwedungen

Für den normalen Anwender sind Voice-over-IP-Techniken momentan nur Interessant, wenn er eine bestehende Internet-Verbindung gleichzeitig für einen Voice-Chat benutzen will. Es gibt aber zwei andere Kundengruppen, die ein grosses wirtschaftliches Interesse an VoIP haben. Zum einen sind das grössere Unternehmen. Die Vereinheitlichung der Netzstruktur bringt Vorteile bei der Verwaltung mit sich. Auch ist die Anschaffung einer einheitlichen Netzstruktur oft billiger als die Installation von zwei parallelen Netzen. Zum anderen haben auch Call Center ein grosses Interesse an VoIP. Call Center sind Dienstleistungsbetriebe. Durch Dienste wie Click to Dial könnte ein Benutzer auf einer Web-Seite ein Telefongespräch aufbauen. (Fig.1 zeigt die parallele Installation von Internet und Telefonanlage)

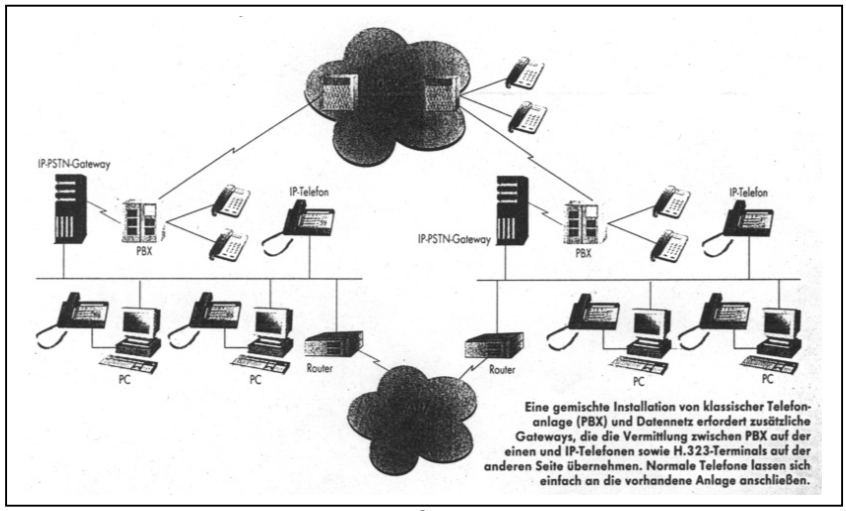


fig.1

Funktionsweise/Technische Nachteile

Das Internet ist ursprünglich nicht für die Sprachübertragung konzipiert worden, was eine Reihe von technischen Konsequenzen hat.

- Heutige Internet-Router arbeiten im allgemeinen langsamer als Telefon Switches. Zudem routen sie Datenpakete häufig auf grossen Umwegen zum Zielort, wodurch Echtzeitanwendungen beeinträchtigt sind.
- Die Tonqualität hängt essentiell von der momentanen Netzbelastung. Generell erreicht sie jedoch häufig nicht die gewohnte Qualität konventioneller Telefonie, was sich in Verzerrungen, Rauschen, Echos und sogar im Verlust einzelner Datenpakete bemerkbar macht.
- Firewalls erschweren die IP-Telefonie oder machen sie in Einzelfällen sogar unmöglich
- Bisher ist die IP-Telefonie nur unzureichend standardisiert, was die Kommunikationsmöglichkeiten entscheidend einschränkt. Sofern bei der Variante PC zu PC der Angerufene nicht die selbe Telefonsoftware wie der Anrufer benutzt, ist er i.a. auch nicht erreichbar. Ein vorrangiges Bestreben des VoIP ist es, die weit verbreiteten proprietären Lösungen durch internationale Normen wie z.B den ATM-Standard H.323 zu ersetzen.

Trotz dieser technischen Nachteile ist durch die ständige Weiterentwicklung von IP-basierten Techniken und Endgeräten eine signifikante Verbesserung der Uebertragungsqualität zu erwarten.

Die H.323-Norm

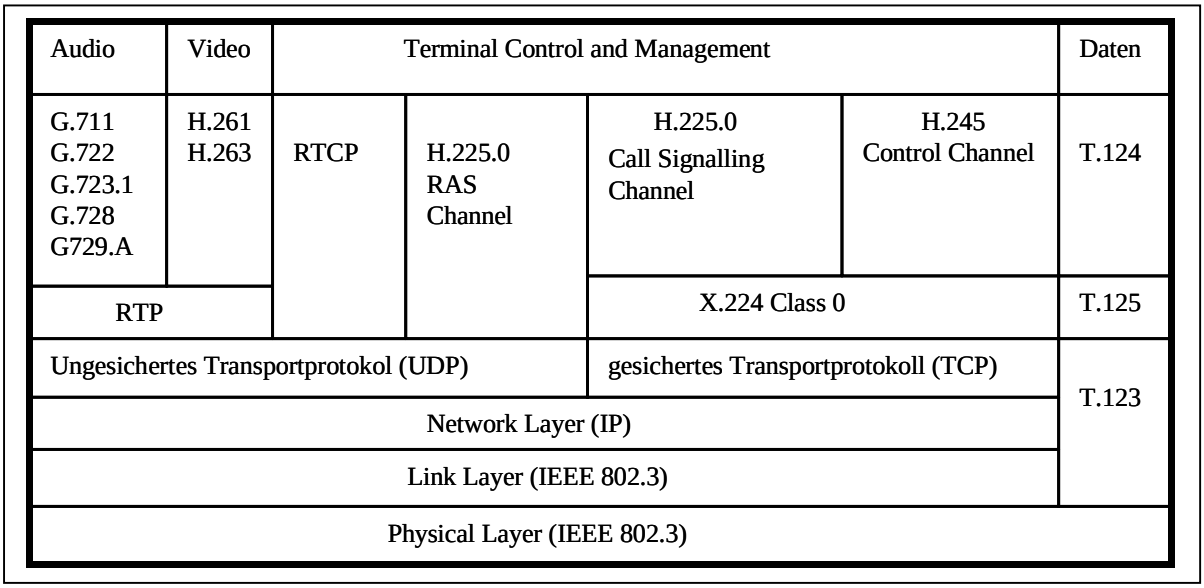


fig.2

Die H.323-Norm (fig.2) ist eine standardisierte Norm, die die Uebertragung von Sprache in Datennetze beschreibt. In den Definitionen ist ein ganzer Protokollstack zusammengefasst, der unterschiedliche Schnittstellen für Sprach-, Video- und Datenübertragung bietet.
Der Anrufaufbau kann in drei Phasen aufgeteilt werden:

RAS: Das H.323 Terminal sendet eine Nachricht zum Gatekeeper, mit Name und Telefonnummer der anzurufenden Person. In dieser Phase erledigt der Gatekeeper drei Funktionen. Er übersetzt die Telefonnummer in die IP-Adresse und überprüft und steuert die Verbindung.

Q.931: Ueber das TCP wird nun die direkte Verbindung hergestellt.
H.245: In dieser Phase überprüfen die Endgeräte welche Dienste sie unterstützen (Audio, Video oder Daten), wobei dann die verschiedenen Codecs (G.711 bisG.729) zum Einsatz kommen.

Für die Kommunikation wird das Real-Time-Protokoll (RTP undRTCP) verwendet. RTP wird über UDP übertragen und enthält Zeit- sowie Synchronisationsinformationen, damit die Datenpakete beim Empfänger in der richtigen zeitlichen Reihenfolge zusammengesetzt werden können. RTCP wird gebraucht um bestimmte Dienstmerkmale der Uebertragung zu gewährleisten.

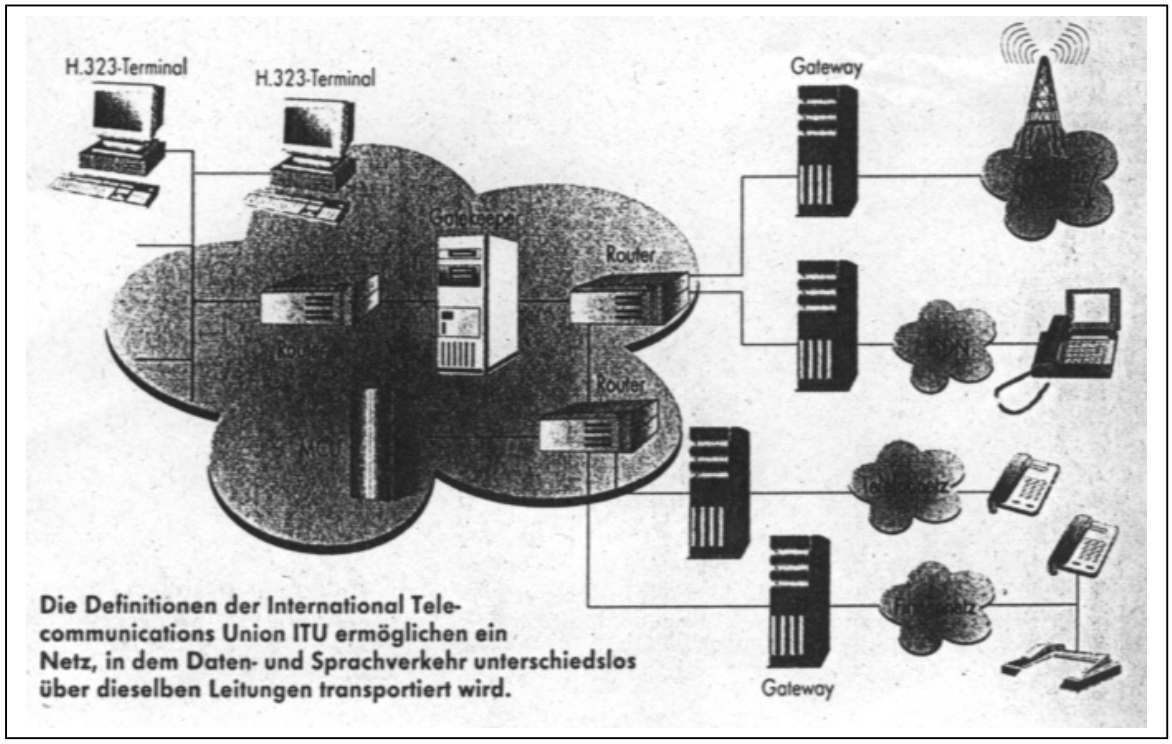


fig.3

Unter dem H.323 Standard können drei verschiedene Objekte im Netzwerk angewählt werden (fig.3). Es können ganz noramle H.323-Terminals sein, Multipoint Control Units, welche Telefonkonferenzen ermöglichen, oder Gateways, die als Schnittstelle zu anderen Netzen dienen. Der Gatekeeper hat beim IP-Netz die gleiche Funktion wie eine Telefonanlage. Der Gatekeeper setzt beim IP-Netz Sprache in Datenpakete um. Ein separater Rechner ist nicht notwendig. Er kann auch in einem Rechner implementiert sein oder eine Software übernimmt seine Aufgabe. Um aber eine Verbindung zum normalen Telefonnetz zu ermöglichen werden zusätzliche Gateways benötigt.

Blick in die Zukunft

Momentan ist VoIP für den Endanwender kaum von Bedeutung. VoIP-Techniken sind dagegen vor allem ein Werkzeug für Firmen und Dienstanbieter, um neue Geschäftsfelder zu erschliessen und die internen Strukturen zu vereinfachen.
Angesichts der diversen Gruppen, die ein Interesse an der Durchsetzung von VoIP haben, ist es nur eine Frage der Zeit, bis sich VoIP auf breiter Basis etablieren wird. Doch bis dahin werden die Unterschiede in den Kosten im Vergleich zum normalen Telefonnetz minimal sein.

Quellenangaben:

-Axell Kossel	Netzgespräche	Reportage c’t 1999 Heft 10
-Jürgen Kuri	Sprache in Päckchen	Reportage c’t 1999 Heft 10
-Rizetto/C. Catania	A voice over IP service architecture	Internet Computing June1999

-http://didaktik.cs.uni-potsdam.de/HyFISCH/Internet/InternetTelefonieBarth.htm

Elektronische Post

Autor:
Martin Walder

Einleitung

Das Versenden von elektronischer Post* ist heutzutage zweifellos eine der wichtigsten Netzwerkanwendung. Seit ungefähr 20 Jahren ist es möglich, Emails zu versenden. Somit ist diese Anwendung des Internets die älteste, die es gibt. Email hat die Kommunikation auf der Erde in vielen Bereichen revolutioniert. Für so manchen ist Email zum wichtigsten Kommunikationsmittel geworden. Vor allem im Geschäftlichen kommt dieser Art, Kontakt mit Menschen auf der ganzen Welt aufzunehmen, eine grosse Bedeutung zu, denn der Zeitverlust, der bei der Kommunikation per Brief oft ein Hindernis für den effektiven Kontakt darstellt, entfällt hier. Elektronische Post gelangt nämlich per Internet blitzschnell zum Empfänger und kann binnen weniger Sekunden auch den entlegensten Winkel der Welt erreichen. Aber inzwischen hat Email auch in der persönlichen, zwischenmenschlichen Kommunikation eine wichtige Stellung eingenommen. Obwohl es heutzutage möglich ist, mit einem Handy jederzeit und (fast) überall mobil Kontakt mit Menschen aufzunehmen, ist das Versenden von elektronischer Post äusserst beliebt. Da Email eine textbasierende Kommunikationsart ist, schafft es – im Gegensatz zur Telefonkommunikation – ein gewisses Mass an Abstand, was sowohl im privaten als auch im geschäftlichen Bereich geschätzt wird. Offenbar vermag der SMS (Short Message Service) beim Mobiltelefon das Email nicht zu verdrängen. Da die Anzahl Zeichen, die mit einer Handy-Kurzmeldung verschickt werden können auf 160 beschränkt ist, eignet es sich im Gegenteil zu Email nicht für den Austausch ausgedehnter Texte mit normalem Inhaltsumfang. Vor allem aber auch die Tatsache, dass per Email zusätzlich zum Text noch Dateien angehängt werden können (sogenannte Attachements), trägt dazu bei, dass Email trotz dem komfortablen Kurzmeldungs-Dienst bei Handys eine starke Stellung eingenommen hat und sich behaupten kann.

Doch auch die negativen Seiten des Email seien hier erwähnt: Da die Email-Adresse inzwischen wie der Vor- und Nachname, wie die Postadresse zu den Daten einer Person gehört, besitzen Unternehmen oft riesige Datenbanken mit Email-Adressen. Dies wird oft ausgenutzt, um unaufgefordert Emails zu versenden, mit denen unerwünschte Werbung gemacht wird. So sind Email-Benutzer oft in der unangenehmen Situation, ihre Mailbox von sogenannten Junk- oder Spam-Mails überflutet vorzufinden.

Datenformat eines Emails

Ein Email besteht aus einem Header, einer Leerzeile und einem Body. Der Header enthält Informationen über Absender und Empfänger und über das Email selbst. Er ist etwa mit einem Briefumschlag zu vergleichen.

Header	**Bezeichnung**
To:	Email-Adresse des primären Empfängers
Cc:	Carbon Copy: Email-Adresse weiterer Empfänger
Bcc:	Blind Carbon Copy: Versteckte Adressen des Emails
From:	Name des Absenders

* Meist E-Mail oder Email genannt: englische Abkürzung für electronic mail. In den vorliegenden Ausführungen wird der Ausdruck „das Email“ verwendet.

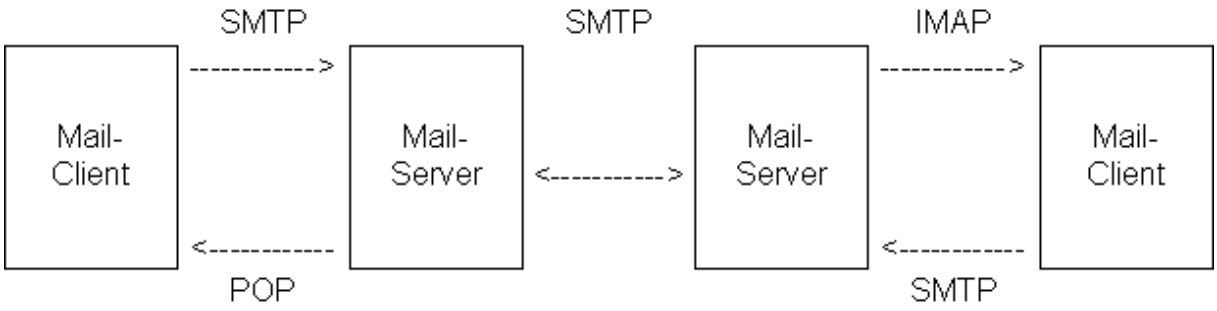
Sender:	Email-Adresse des Absenders
Subject:	Betreff
Date:	Absendedatum
Return-Path:	Identifizierung des Weges zurück zum Absender
Message-ID:	Einzigartige ID, mit der das Email referenziert werden kann
Reply-To:	Email-Adressen derjenigen, an die Antworten gehen sollen

Der Body (der Nachrichtenkörper) enthält die eigentliche Nachricht, also Text aus beliebig vielen Zeichen des ASCII-Codes. Das bedeutet, dass nur Zeichen mit Codes zwischen 0 und 127 zulässig sind (7-Bit-Code). Heute möchte man aber auch Nicht-ASCII-Zeichen versenden können, also spezielle Buchstaben (zum Beispiel Umlaute, Buchstaben mit Akzenten) oder Elemente aus Sprachen ohne Alphabet. Ausserdem möchte man die Möglichkeit haben, Files (Bilder, Audios oder Präsentationen), sogenannte Attachments, versenden zu können. Solche Nicht-ASCII-Zeichen können beliebige Bytes zwischen 0 und 255 enthalten. Folglich muss man den 8-Bit-Datenstrom kodieren, so dass er sich in 7-Bit-ASCII-Zeichen unterbringen lässt. Deshalb hat man einen Standard eingeführt, der dies erlaubt. Es handelt sich dabei um MIME (Multipurpose Internet Mail Extensions), den es seit 1996 gibt. Er führt innerhalb des Body eine Struktur ein, die durch speziell formatierte Zeilen im Header vorgegeben ist. Ein Email lässt sich auf diese Weise in mehrere Teile zerlegen, wobei jedem ein spezieller Header vorangeht, der angibt, um welche Art von Information es sich handelt und wie sie kodiert ist. Sogar verschachtelte Strukturen sind möglich, das heisst ein Teil kann selbst wieder aus mehreren Einzelteilen bestehen.
Hier einige MIME-Typen als Beispiele:

Typ	**Untertyp**	**Beschreibung**
Text	Plain	Unformatierter Text
Text	Richtext	Text mit einfachen Formatierungskommandos
Image	Gif	Bild im GIF-Format
Image	Jpeg	Bild im JPEG-Format
Audio	Wav	Sound im WAV-Format
Video	Mpeg	Video im MPEG-Format
Application	Octet-stream	Uninterpretierte Byte-Sequenz
Application	Postscript	Druckbares Dokument im Postscript-Format

Email-Transfer

Da Email eine Reihe von Protokollen ist, muss man eine Unterteilung in zwei grosse Bereiche vornehmen. Der erste Bereich betrifft das Problem, Nachrichten durch das Internet an einen bestimmten Mail-Client zu versenden. Dies wird durch das SMTP (Simple Mail Transfer Protocol) oder durch das ESMTP (Extended SMTP) erledigt. Der zweite Bereich betrifft die Frage, wie Email-Nachrichten durch den Client vom Mail-Server abgeholt werden können. Die dafür verwendeten Protokolle sind POP (Post Office Protocol) oder IMAP (Internet Message Access Protocol). Die folgende Darstellung veranschaulicht diese Zusammenhänge:



Protokolle für die Email-Übertragung

Der Mail-Client kann entweder als Ziel- oder als Sendermaschine agieren – je nach dem, ob er empfängt oder sendet. Wird in der obigen Darstellung beispielsweise der linke Mail-Client als Empfänger betrachtet, so sieht man, dass dieser das Protokoll POP benutzt, um Emails zu abzurufen.
Um ein Email an den gewünschten Ort versenden zu können, braucht es eine eindeutige Identifizierung der Empfängermaschine. Dazu dienen Email-Adressen, die aus einem Benutzernamen, gefolgt von einem „@“ und einem Domain-Namen bestehen.

Zum Beispiel: von-der-heide@ee.ethz.ch

Beim Versenden eines Emails wendet sich der Mail-Client zunächst an den DNS (Domain Name Server), um aus dem symbolischen Rechnernamen (der Text, der dem „@“ folgt, im Beispiel „ee.ethz.ch“) die dazu eindeutig bestimmte IP-Adresse zu bestimmen. Der Domain-Name ist wie bei einer Internet-Adresse hierarchisch aufgebaut: Er besteht aus mehreren durch Punkte getrennten Teilen, wobei der letzte Teil (im Beispiel „ch“) die sogenannte Top-Level-Domain darstellt. Die Hierarchie kann prinzipiell beliebig tief geschachtelt werden. Der Name vor dem „@“ schlussendlich dient dann der genauen Identifikation des Accounts.

Im Folgenden werden die beiden Abläufe und die dazugehörigen Protokolle näher beschrieben.

Senden von Email

In der Situation, bei der eine Maschine Emails direkt aus dem Internet empfangen kann, sieht der Prozess des Sendens wie folgt aus: Ein Email wird von einer Quelle verschickt, indem die zugehörige Maschine des Senders eine TCP-Verbindung auf dem Port 25 zu einer Zielmaschine herstellt. An diesem Port lauscht auf der Empfängerseite ein Hintergrundprozess (Daemon), der das SMTP (Simple Mail Transfer Protocol) versteht. Der Daemon akzeptiert eintreffende Verbindungen und kopiert Nachrichten in ihre zugehörigen Mailboxen auf dem Mail-Server. Kann eine Nachricht nicht ausgeliefert werden, wird eine Fehlermeldung zurückgeschickt. Nach dem Aufbau der TCP-Verbindung über Port 25 agiert die sendende Maschine als Client und die empfangende Maschine als Server.

SMTP (Simple Mail Transfer Protocol)
Die Basis für das Senden von Emails bildet das SMTP, ein einfaches ASCII-Protokoll. Es ist das Internet-Standardprotokoll zum Übertragen von Email-Nachrichten. Es vermittelt also zwischen Sender und Empfänger. SMTP ist im

Internet-Standard RFC[**] 821 [i] definiert. Wie es der Name schon sagt, funktioniert SMTP ziemlich einfach: Der Absender eines Emails (gewöhnlich ein Email-Programm oder ein Programm, wie zum Beispiel ein Web-Browser, das diese Funktion enthält) öffnet eine Verbindung zum Empfänger und sendet die Nachricht mittels einiger weniger SMTP-Befehle. Dabei kann der Empfänger entweder der Mail-Host des Adressanten sein oder ein Übertragungssystem, welches das Email solange weiterleitet, bis es beim Mail-Host des Adressanten angekommen ist. Der Mail-Host ist der Rechner, der ein Email vie SMTP empfängt; er wird oft auch mit Mail-Server bezeichnet. Wie bereits erwähnt, erlaubt der SMTP-Standard nur das Verwenden von ASCII-Zeichen und somit unterstützt er das Anhängen von binären Dateien nicht.

ESMTP (Extended SMTP)
Da das SMTP für einige Anwendungsgebiete zu einfach und somit ungenügend ist, wurden im Internet-Standard RFC 1869 [ii] einige Erweiterungen definiert, die diese Lücken schliessen sollen. Das so entstandene erweiterte SMTP wird sinngemäss ESMP (Extended Simple Mail Transfer Protocol) genannt. Voraussetzung für das erfolgreiche Verwenden von ESMTP ist natürlich, das beide Seiten, das heisst Sender und Empfänger, diese Erweiterung unterstützen.

Die meistgenutzten Erweiterungen sind hier aufgelistet:

- 8-Bit-Zeichensätze:
  Bei SMTP werden nur ASCII-Zeichen unterstützt. ESMTP definiert eine Möglichkeit, wie Nicht-ASCII-Zeichen sicher übertragen werden können.
  (RFC 2047 [iii])
- Ankündigen der Nachrichtengrösse:
  Da Emails sehr umfangreich sein können, kann es sinnvoll sein, dass der Absender den Empfänger zunächst über den Umfang einer Nachricht in Kenntnis setzt. Der Empfänger kann dann entscheiden, ob er die Nachricht akzeptieren oder ablehnen möchte.
  (RFC 1078 [iv])
- Zerlegung umfangreicher und binärer Nachrichten:
  Beim Senden umfangreicher Nachrichten kann es sinnvoll sein, ein Email in mehrere Teile zu zerlegen. Im Falle eines Übertragungsfehlers ist es dann ausserdem möglich, nur einzelne Teile der Nachricht erneut zu versenden. So muss nicht das ganze Email nochmals versendet werden.
  (RFC 1845 [v])

Empfangen von Email

Obwohl das Senden von Email für eine erfolgreiche Nachrichtenübermittlung wichtig ist, muss auch das entgegengesetzte Problem des Empfangens gelöst werden. Zu Beginn der Email-Geschichte wurde die Nachricht immer direkt auf dem Mail-Host gelesen. Der Mail-Host (=Mail-Server) ist der Rechner, der ein Email via SMTP empfängt. So entsprach das Lesen eines Emails in den meisten Fällen dem Lesen einer Datei, in die das SMTP-Programm alle ankommenden Nachrichten schrieb.

[**] RFC: Request for comment. Beschreibungen von Protokollspezifikationen und anderen Standards für das Internet. Alle Dokumente sind mit einer laufenden Nummer versehen. RFCs bilden die Diskussionsgrundlage für die Entwicklung neuer Standards

Dieses Modell funktioniert allerdings nur, wenn der Benutzer auf das gleiche Dateisystem zugreifen kann wie das SMTP-Programm.

Man möchte aber auch Zugriff auf Emails auf dem Mail-Server haben, ohne dass die Notwendigkeit eines gemeinsamen Dateisystems besteht. Dazu hat sind spezielle Netzwerkprotokolle definiert worden. In einem solchen Modell greift der Benutzer als Client über ein Netzwerk auf den Mail-Server zu. Es sind drei verschiedene Arten von Zugriffen möglich:

- Offline:
  Im Offline-Betrieb verbindet sich der Client periodisch mit dem Mail-Server, holt dort neue Nachrichten ab und beendet die Verbindung wieder. Die Daten werden dabei auf dem Server gelöscht und lokal auf dem Client-Rechner weiterverarbeitet.
- Online:
  Im Online-Betrieb verbleibt das Email auf dem Mail-Server und wird aus der Entfernung bearbeitet. Der Vorteil dieses Modells ist, dass der Benutzer die Nachrichten von verschiedenen Client-Rechnern abrufen kann.
- Getrennt:
  Der getrennte Betrieb entspricht dem Offline-Betrieb mit dem Unterschied, dass die Mails auf dem Server nicht gelöscht werden. Die Nachrichten werden beim Client in einem Nachrichten-Cache abgelegt und können lokal bearbeitet werden. Bei einer Verbindung mit dem Server werden die Daten synchronisiert.

Die beiden wichtigsten Protokolle für den Zugriff auf Emails, die auf dem Mail-Host gespeichert sind, sind POP und IMAP, die im Folgenden näher beschrieben sind:

POP (Post Office Protocol)
Ist das Versenden einer Nachricht erfolgreich, wird sie auf dem Mail-Server gespeichert. Meistens nimmt der Empfänger nur gelegentlich (periodisch) Kontakt zu Mail-Server auf, um seine Mails zu lesen. Dieses Abrufen der gespeicherten Nachricht kann als Analogie zum Gang zur Poststelle und dem Nachschauen im Postfach angesehen werden. Deshalb trägt das dazu verwendete Protokoll auch den Namen Post Office Protocol. POP ist im Internet Draft Standard RFC 1939 [vi] definiert. POP unterstützt lediglich den Offline-Betrieb beim Nachrichtenzugriff (siehe oben). Die grundlegende Funktion von POP ist sehr einfach. Der Mail-Server, der POP benutzt (der POP-Server), wartet auf eingehende Verbindungen auf einem bestimmten Port. Letztere wird über TCP erstellt, wenn der Client eine POP-Sitzung wünscht. Mit einem User-Name und dem dazugehörenden Passwort kann sich der Client einloggen. Ist die Verbindung erstellt, kann er im Wesentlichen drei Kommandos ausführen:

- Abrufen der Anzahl Mails
- Anzeigen eines bestimmten Mails
- Löschen eines bestimmten Mails

Nachdem die gewünschten Aktionen getätigt sind, wird die Verbindung wieder geschlossen. Die Anzahl POP-Befehle ist also sehr klein, so dass ein POP-Server oder POP–Client recht einfach zu implementieren ist.

Eine Verbindung zum POP-Server kann zum Beispiel ganz einfach per Telnet-Verbindung zum Standard-Port für POP, dem Port 110, hergestellt werden. Zurzeit ist POP3 die aktuelle Version dieses Protokolls.

IMAP (Internet Message Access Protocol)
Das aufwendigere IMAP bietet wesentlich mehr Möglichkeiten als POP und ist deswegen benutzerfreundlicher. Beim Einsatz von IMAP bleiben die Mails auf dem Mail-Server gespeichert (Online-Betrieb, siehe oben). Die wichtigsten Vorteile von IMAP gegenüber POP sind hier angegeben:

- Unterstützung verschiedener Ordner:
  Es besteht die Möglichkeit, neben dem Inbox-Ordner noch weitere Ordner zu erstellen und zu bearbeiten. Auch Ordnerhierarchien werden unterstützt.
- Ordnerbearbeitung über das Netzwerk:
  Nachrichten können von einem Ordner in einen anderen verschoben und Nachrichten-Flags gesetzt werden, um Nachrichten zum Beispiel als gelesen zu markieren.
- Optimierte Online-Performance:
  Da MIME-Emails sehr umfangreiche Teilstücke (wie beispielsweise Bilder oder Videos) enthalten können, erlaubt IMAP das getrennte Abholen von MIME-Teilstücken. Deswegen unterstützt IMAP das Erkennen der Nachrichtenstruktur, um zum Beispiel nur einen Teil herunterladen zu müssen. Dies führt zu effizienteren Sitzungen im Sinne des zeitlichen Gewinns.

Obwohl IMAP im Vergleich zu POP funktionell klar überlegen ist, muss auch ein Nachteil erwähnt werden: IMAP, das zurzeit in der Version 4 vorliegt, ist erheblich komplexer als POP und deshalb schwieriger zu implementieren und fordert die Ressourcen der Hardware mehr.

Zusammenfassung

Email ist eine Internetanwendung, die aus der heutigen Welt nicht mehr wegzudenken ist. Die RFC-Standards haben Email dazu verholfen, eine der wichtigsten Kommunikationsarten zu werden. Obwohl das Versenden von elektronischer Post auch missbraucht werden kann (beispielsweise für die Verbreitung von bösartigen Virus-Programmen), wird sich Email längerfristig auf jeden Fall behaupten. Es ist kaum denkbar, dass Email in naher Zukunft von einem Kommunikationsmittel mit wesentlichen Vorteilen abgelöst wird. Allerdings werden die Möglichkeiten, elektronische Post mobil abzurufen oder zu versenden, noch stark ausgebaut respektiv verbessert werden.

---

[i] RFC 821: Jonathan B. Postel, August 1982, University of Southern California
[ii] RFC 1869: J. Klensin, November 1995, Brandenburg Consulting
[iii] RFC 2047: K. Moore, November 1996 University of Tennessee
[iv] RFC 1078: M. Lottor, November 1988
[v] RFC 1845: D. Crocker, September 1995, Brandenburg Consulting
[vi] RFC 1939: J. Myers & M. Rose, Mai 1996

MobileIP

Christian Morf

ETH Zürich
Seminar Internettechnologie
Sommersemester 2001

1. Motivation für Mobile IP

Als IP entwickelt wurde, dachte noch niemand daran, dass in Zukunft mobile Computer in ein Netzwerk integriert werden müssen. Jeder Computer hatte seinen Standort mit einer fixen Netzanbindung und einer eigenen, einmaligen Adresse, die ebenfalls standortabhängig ist.
Mit dem Aufkommen mobiler Computer traten viele Probleme auf.
Wie sollen mobilen Geräte in ein bestehendes Netzwerk integriert werden, welche IP-Nummer sollten sie haben? Was passiert, wenn ein mobiles Gerät das Subnetz wechselt und über einen anderen Router kommuniziert, der andere IP-Präfixe besitzt? Die IP-Adresse müsste gewechselt, und allen anderen Netzteilnehmern mitgeteilt werden, da ansonsten keine Datenpakete mehr empfangen werden könnten. Zusätzlich müssten noch alle DNS-Einträge aktualisiert werden, da ansonsten das Gerät nicht mehr unter seinem Namen zu finden wäre. Das Wechseln der DNS-Einträge dauert aber zu lange. Zudem ist es aus Sicherheitsgründen nicht jedem Endgerät gestattet, die DNS-Einträge zu manipulieren. Weiterhin würden die bestehenden TCP-Verbindungen abbrechen, wenn die IP-Adresse gewechselt wird. Ein nahtloser Übergang zwischen den Subnetzten ist also nicht möglich.
Es gibt 2 verschiedene Arten von Mobilität. Die eine besteht darin , den mobilen Computer an einem Ort auszusteck und an einem anderen wieder anzuschliessen. Dies ist eigentlich kein Problem mehr, seit es DHCP gibt. Mit der Entwicklung von Wireless Lan‘s wurde eine neue Art von Mobilität möglich. Die Computer konnten nun ihren Standort wechseln, ohne die physische Anbindung zu unterbrechen. Der ganze Netzwerkaufbau wird somit viel dynamischer, ähnlich einem GSM Netzwerk. Um eine permanente und unterbruchslose Verbindung unter derselben IP-Adresse zu garantieren, wurde 1996/97 Mobile IP entwickelt. Mobile IP ist eine Erweiterung des bestehenden Internetprotokolls. Am bestehenden Protokoll wird nichts geändert, da dieses bereits in unzähligen Netzwerken Standard ist und eine Anpassung aller Geräte an ein neues Protokoll nicht zumutbar wäre.

2. Anforderungen an Mobile IP

• Kompatibilität

Mobile-IP muss zusammen mit dem bestehenden IP funktionieren. Die Verbreitung von IP ist zu gross, als dass daran Änderungen vorgenommen werden können. Alle schon bestehenden Programme wie Browser müssen weiterhin funktionieren. Dasselbe gilt für Betriebssysteme und die Software in den Routern. Mobile-IP muss ebenfalls mit den tieferliegenden Schichten kompatibel sein. Dies bedeutet, dass Mobile-IP identische Schnittstellen besitzen muss, wie das herkömmliche IP. Auch das Format der IP-Adresse sollte derselbe bleiben.

• Transparenz

Die Mobilität soll für die anderen Schichten und die Anwendungsprogramme unsichtbar sein. Selbst wenn das mobile Endgerät seinen Aufenthaltsort wechselt, sollen die höherliegenden Schichten ohne Unterbruch weiterarbeiten d.h., die Verbindung darf nicht unterbrochen werden. Die IP-Adresse des mobilen Geräts muss diesselbe bleiben, auch wenn es sich in einem anderen Subnetz mit anderen Präfixen aufhält.

• Effizienz und Skalierbarkeit

Die Bandbreite in kabellosen Netzwerken ist beschränkt. Deshalb soll die zusätzliche Datenmenge von Mobile-IP möglichst gering gehalten werden. Um späteres Wachstum zu ermöglichen, sollte Mobile-IP weltweit eine grosse Zahl von mobilen Endsystem unterstützen können.

• Sicherheit

Mobilität ist immer ein Sicherheitsproblem. Fremde, nicht autorisierte Geräte könnten sich an das Netzwerk anbinden. IP überprüft nur ob die Adresse des Empfängers korrekt ist. Jedes Datenpaket, das via Mobile-IP versandt wird, muss sich authentifizieren.

3. Terminologie

Die folgenden fünf Begriffe werden im Zusammenhang mit Mobile IP verwendet und an dieser Stelle kurz erklärt.

• Mobile Node (MN)

Als MN wird das Endsystem bezeichnet, welches den Netzanschluss wechseln kann, ohne seine IP-Adresse zu wechseln.

• Home Agent (HA)

Der HA befindet sich im Heimnetz des MN, typischerweise auf dem Router, der den Aufenthaltsort des MN verwaltet. Als Heimnetz wird das Subnetz verstanden, zu dem der MN laut seiner IP-Adresse gehört.

• Foreign Agent (FA)

Der FA befindet sich im momentanen Fremdnetz, also in dem Netz, indem sich der MN momentan aufhält. Er befindet sich ebenfalls auf dem Router, welcher die vom HA empfangenen Pakete an den MN weiterleitet.

• Care of Address (COA)

Die COA stellt den momentanen Aufenthaltsort des MN dar. Alle an den MN gesendeten Pakete gelangen zuerst zur COA. Die Weiterleitung der Pakete zum MN geschieht durch einen Tunnel. Die COA gibt den Tunnelendpunkt an.

• Correspondent node (CN)

Der CN ist der Kommunikationspartner des MN. Er kann fix oder ebenfalls mobil sein.

4. Datentransfer

Die an den MN zu senden Pakete werden vom CN wie üblich mit der IP-Adresse des MN versehen und erreichen auf dem herkömmlichen Weg den Router im Heimnetz des MN. Auf diesem befindet sich der HA des MN. Dieser weiss, ob sich der MN momentan im Heimnetz oder in einem Fremdnetz befindet. Befindet sich der MN im Heimnetz, wird das Paket unverändert weitergeschickt. Ist er momentan in einem Fremdnetz, wird das Paket abgefangen und **gekapselt**. Dies bedeutet, dass das ursprüngliche und vollständige IP-Paket (inklusive Header) in den Datenteil eines neuen IP-Paketes gepackt wird. Als Zieladresse des neuen Pakets wird die COA eingetragen. So verpackt wird das Paket **getunnelt**, was eigentlich nichts anderes bedeutet als dass es auf einem klar vorgegebenen Weg durch das Internet geschickt wird. Über die COA, welche den **Tunnelendpunkt** darstellt, gelangt das Paket schlussendlich zum FA. Dieser entkapselt das Paket und erhält somit wieder das ursprüngliche Datenpaket. Der FA leitet es dann an den MN weiter.

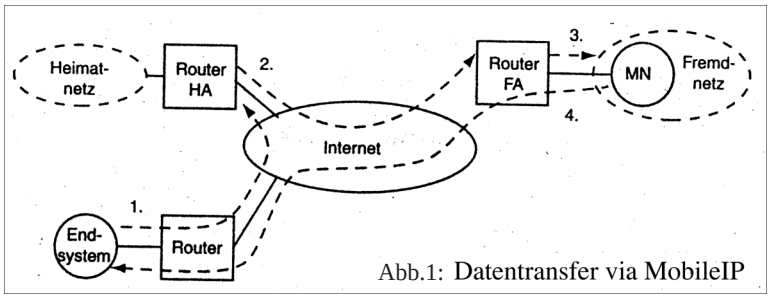


Abb.1: Datentransfer via MobileIP

Der Rückweg, vom MN zum CN ist wesentlich einfacher. Der MN kennt die Adresse des CN aus dem Absender des empfangenen Datenpaketes und schickt nun seine Pakete auf direktem Weg, ohne Umwege über FA/HA, an den CN. Als Absender gibt der MN seine IP-Adresse an. Diese Methode wird **Triangular-Routing** (siehe Abb.2) genannt. Der CN erfährt auf diesem Weg nicht, dass sich der MN nicht in seinem Heimnetz befindet. Er schickt seine Pakete weiterhin an die Adresse des MN. Auch der MN „sieht“ nichts von der Mobilität. Er erhält die Pakete so, wie sie vom CN gesendet wurden und schickt diese auch auf dem herkömmlichen Weg zurück. Wäre der CN ebenfalls ein mobiles System, würde der soeben erklärte Ablauf in umgekehrter Richtung nochmals stattfinden. (Abb.1)

Beim direkten Senden vom MN zum CN gibt es noch ungelöste Sicherheitsprobleme. Sendet der MN seine Pakete aus einem fremden Netzwerk direkt zu einem CN, ist der Absender des MN topoligisch inkorrekt, d.h., der MN besitzt eine IP-Adresse, die nicht mit den Adresspräfixen des fremden Subnetzes übereinstimmt. Viele Firewalls von Firmen lassen solche Pakete nicht passieren, um zu verhindern, dass ein fremder Computer ausserhalb des Firmennetzwerks sich mit einer internen IP-Adresse als firmeninterner Computer ausgeben kann.

5. Agent Advertisement

Wie findet ein MN einen FA, wenn er das Netzwerk wechselt? Wie merkt der MN überhaupt, dass er seinen Aufenthaltsort gewechselt hat? Um diese Probleme zu lösen, senden FA und HA periodisch **advertisement messages**. Diese Nachrichten können als Rundfunk betrachtet werden, die ohne konkretes Ziel ins jeweilige Subnetz gesendet werden. Diese Art von advertisement messages benutzten Router schon vor Mobile IP um ihre Präsenz zu markieren. Ein MN kann nun über diese advertisements die COA empfangen und weiss nun, ob er sich im Heimnetz oder in einem Fremdnetz befindet. Falls keine advertisements vorhanden sind, kann der MN **agent solicitations** ins Subnetz senden, um einen Agenten zu finden.

Dieser Anmeldevorgang findet immer statt, auch dann, wenn der MN eine Verbindung zu einem FA/MN hat. Dies ist nötig, um einen nahtlosen Übergang von einem Subnetz in das andere zu gewähren. Der MN sucht ständig nach besseren Verbindungen, während er noch über den „alten“ Weg sendet.
Der nächste Schritt nach dem Empfangen der COA ist die **Registration**. Dabei informiert der MN den HA über seinen aktuellen Aufenthaltsort. Die Registration kann über den FA oder direkt vom MN zum HA geschehen. Diese Registrationsnachrichten müssen alle authentisiert sein, um eine Anmeldung eines nicht befugten Rechners zu verhindern.

Befindet sich der MN zum erstenmal im Netz, besitzt er noch keinen HA. In diesem Fall wird einfach ein HA auf dem nächstgelegenen Router dem MN zugewiesen. Dieser HA informiert dann die anderen Rechner und Router, dass nun ein neues Endsystem über ihn zu erreichen ist.

6. Optimierungen

• Umgehen des Triangular-Routing

Das Senden von Paketen via HA zum MN und zurück vom MN zum CN, auch als Triangular-Routing bezeichnet (siehe Abb.2), stellt oft einen unnötigen Umweg für Pakete dar, der zu grösseren Verzögerungen und einer erhöhten Netzlast führen kann. Ein Lösungsansatz für dieses Problem besteht darin, dass ein Sender den aktuellen Aufenthaltsort eines MN kennenlernt und seine Pakete dann direkt zu diesem schickt. Den aktuellen Aufenthaltsort kann ein Sender vom HA mitgeteilt bekommen, nachdem er das erste Mal ein Paket via HA zum MN gesendet hat.

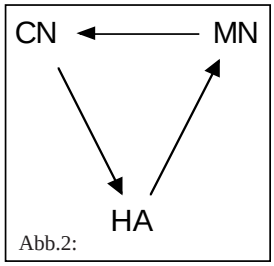


Abb.2:
Triangular Routing

• Minimale Kapselung

Bei der IP-in-IP-Kapselung besitzt das Paket genau genommen zwei Header. Den aktuell gültigen Header des Aussenpaketes und den Header des ursprünglichen Pakets, welches sich im Datenteil des Aussenpakets befindet. Viele Felder dieser beiden Header sind gleich. Um die Paketgrösse zu vermindern, werden Felder des inneren Headers, die dieselben Daten wie der äussere Header enthalten, gelöscht. Im inneren Header befindet sich im wesentlichen nur noch die Adresse des MN und die des CN sowie der Typ der enthaltenen Nachricht. Bei der Entkapselung wird der innere Header mit Hilfe des äusseren Headers wieder komplett hergestellt.

• Kontinuität des Datenstroms beim Wechseln des FA

Diese Optimierung betrifft die IP-Pakete, die noch zum alten FA unterwegs sind, während der MN sich vom alten zum neuen FA weiterbewegt hat. Damit diese Pakete nicht verloren gehen, kann der alte FA vom neuen FA benachrichtigt werden, so dass der alte FA noch für den MN ankommende Pakete an den neuen FA weiterleiten kann. (Abb.3)

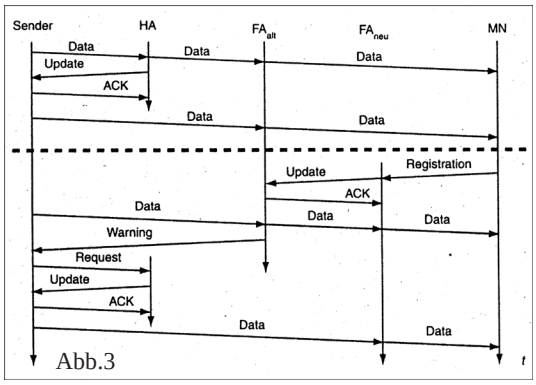


Abb.3

7. Mobile IP & IPv6

Die neue IP Version 6 erleichtert einiges für Mobile-IP. Sicherheitsmechanismen müssen nun nicht mehr zusätzlich aufgesetzt werden, da sie schon in IPv6 integriert sind. Weiterhin wird kein FA mehr benötigt. Alle Router beherrschen nun das sogenannte Router Advertisement, welches anstelle des speziellen Agent Advertisement eingesetzt werden kann und der MN ist mit IPv6 in der Lage, ankommende Pakete selbst zu entkapseln.
Um die Probleme beim direkten Senden vom MN zum CN (siehe Kapitel 4) zu lösen, wurde das **Reverse Tunneling** entwickelt. Dabei ist ein Tunnel für die Daten zwischen dem MN und dem CN vorgesehen. Somit können die Daten durch Firewalls geschickt werden, falls diese das Tunneling zulassen.

Grundlagen der Internet-Technologie

Drahtlose Kommunikation im Internet

WAP

Adrienne Heinrich
Vortrag 14, 18. Juni 2001

1. Einführung

Das Bedürfnis nach mehr Mobilität und Unabhängigkeit wird in unserer Gesellschaft immer grösser. Es reicht uns nicht mehr, von überall jeden erreichen zu können, sondern wir wollen Mobilität in jedem Lebensbereich. Das Handy soll uns im Auto Staumeldungen, im Zug die aktuellen Aktienkurse, die Wetterprognosen in der Warteschlange im Supermarkt anzeigen und mit Spielen unterhalten, wenn es uns langweilig ist.
Um den Handy-Nutzern mehr anbieten zu können, versuchte man zwei explosionsartig wachsende Gebiete, den Mobilfunk und das Internet, miteinander zu verbinden.

2. Probleme

Die Übertragung von HTML-Seiten auf mobilen Endgeräten ist nur begrenzt sinnvoll aufgrund einiger technischer Probleme wie:

- zu kleines Display
- stark begrenzter Speicher
- niedrige Übertragungsrate (9,6kbps)
- Graphiken nur beschränkt einsetzbar
- unterschiedliche Handys

Um einen Standard auszuarbeiten, der mit diesen Problemen fertig werden soll, wurde das WAP-Forum gegründet.

3. Das WAP Forum

Das WAP Forum ist eine Non-Profit-Organisation, die aus dem Zusammenschluss der Mobilfunkhersteller Ericsson, Nokia, Motorola und Phone.com (früher Unwired Planet) im Dezember 1997 hervorgegangen ist.
Als das WAP-Forum gegründet wurde, versuchte man eine möglichst heterogene Zusammensetzung von Interessengruppen zu erreichen. Es sollte ausgeschlossen werden, dass weder Hardware- noch Softwarefirmen die Überhand bei der Entwicklung des WAP erhielten. Der Vereinigung kann jedes Unternehmen beitreten, das sich bereit erklärt, die beschlossenen Spezifikationen einzuhalten. Bis heute stieg die Zahl der Mitglieder auf über 400 an (siehe Literaturquelle [4.2]).
Ziel dieses Zusammenschlusses war es, einen weltweiten Standard für den Datenverkehr in Mobilfunknetzen zu erarbeiten. Das WAP-Forum ist also für die einheitliche Umsetzung von WAP in den verschiedenen Handys verantwortlich. Die gesamte Industrie und vor allem die Nutzer von mobilen Geräten können von einer einzigen offenen Spezifikation profitieren, da sie sich auf die Kompatibilität der Hardware und der angebotenen Applikationen verlassen können.

4. Was ist WAP?

WAP steht für die Abkürzung von „Wireless Application Protocol“ und basiert auf der Sprache WML (Wireless Markup Language).
Mit Hilfe dieser Technologie ist es möglich, Informationen aus dem Internet auf mobilen Endgeräten, wie Handys oder PDAs darzustellen.
Die ersten WAP-Mobiltelefone kamen Ende 1999 auf den Markt.

6. WAP-Architektur

Die WAP-Architektur basiert auf einem schichtenförmigen Modell, wie wir es auch von anderen Netzwerkprotokollfamilien, wie beispielsweise ISO/OSI, her kennen. Das „WAP-Stack“ wird in fünf Schichten unterteilt:

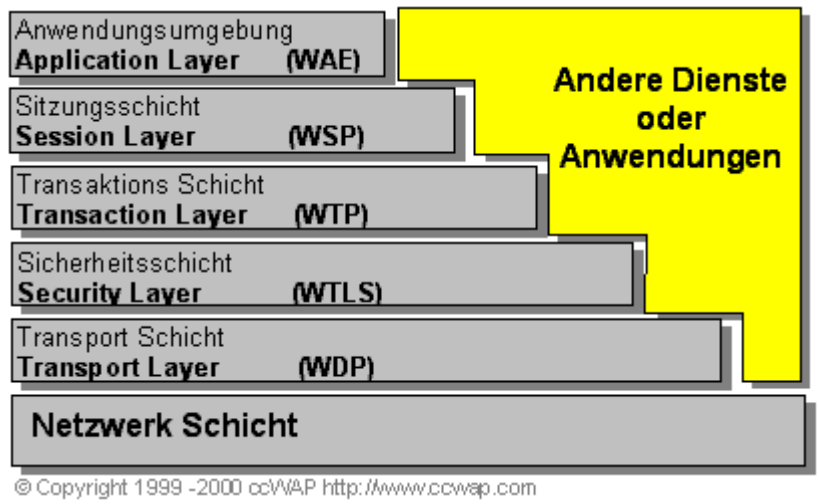


Abb. 1: Schichtenmodell des WAP-Stack

In jeder dieser Schichten kommen Anwendungen und Protokolle gleichermassen zum Einsatz. Diese Schicht-Architektur ermöglicht es anderen Anbietern, Anwendungen und Dienste für die entsprechende Schichte anzubieten, indem sie die vom WAP-Stack unterstützten Funktionen nutzen.

- ***Anwendungsschicht***

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

- ***Sitzungsschicht***

  Das WSP (Wireless Session Protocol) enthält alle Spezifikationen für die Sitzung. Diese besteht hauptsächlich aus drei Phasen:

  - Sitzung starten
  - Inhalte austauschen
  - Sitzung beenden

  Zusätzlich kann eine Sitzung auch abgebrochen und wieder aufgenommen werden.

- ***Transaktionsschicht***

  Die Spezifikationen für die Übertragungsschicht enthält das WTP (Wireless Transaction Protocol). Als Transaktion gilt die Kommunikation zwischen dem Initiator einer Anfrage und einem Antwortenden. Das WTP bietet drei Klassen von Übertragungsdiensten:

  - Unzuverlässige Einweganfrage
  - Zuverlässige Einweganfrage
  - Zuverlässige Zweiweganfrage und –antwort

- ***Sicherheitsschicht***

Das WTLS (Wireless Transport Layer Security) beinhaltet Verschlüsselungseinrichtungen zur Sicherung der Datenintegrität, Privatsphäre und Authentifizierung.

- ***Transportschicht***

Das WDP (Wireless Datagram Protocol) repräsentiert die Transportschicht und ist für die Kommunikation zwischen dem Bearer (Schnittstelle zwischen WAP und physikalischen Netzen wie z.B. GSM) und den darüber liegenden Schichten zuständig. Dank der abstrakten WDP-Schicht ist WAP auch in aussereuropäischen Netzen und zukünftigen Techniken (GPRS, UMTS) einsetzbar.

5. WAP-Request

WAP-Inhalte und –Anwendungen werden in webbasierten Formaten auf einem Webserver abgelegt. Das Anfragen einer solchen speziell für WAP-Handys aufbereitete Seite nennt man „WAP-Request“. Als Vermittler zwischen Handy und Server fungieren sogenannte Gateways.
Ein typischer WAP-Request verläuft nach folgendem Schema (siehe Abb. 2):

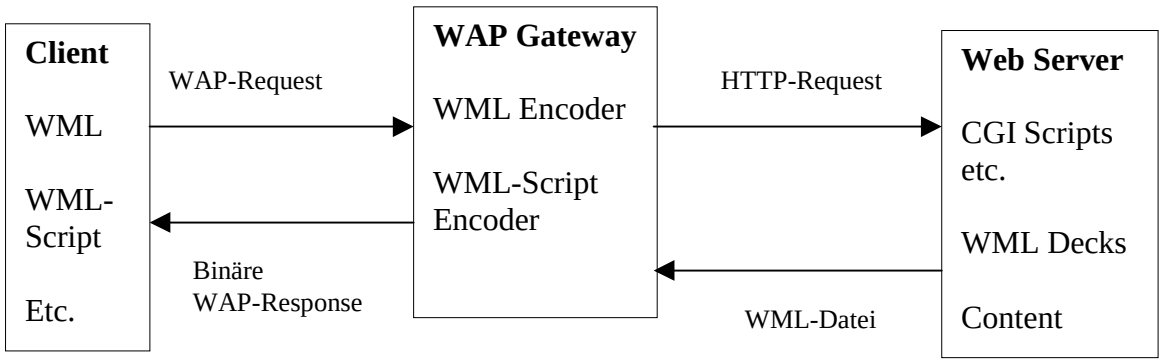


Abb. 2: Ablauf eines WAP-Requests

1. Die Eingabe einer URL, die zum Beispiel als Link im Browser hinterlegt sein kann, erzeugt innerhalb des Client einen Request.
2. Der Request wird mit Hilfe des WAP-Protokolls an ein WAP-Gateway übertragen.
3. Das WAP-Gateway wiederum transformiert den Request in einen herkömmlichen HTTP-Request und leitet diesen an den entsprechenden Web-Server weiter.
4. Der Web-Server bearbeitet den HTTP-Request wie gewohnt und gibt sogenannte WML-Dateien an das WAP-Gateway zurück.
5. Innerhab des WAP-Gateways erfolgt die Verifizierung der WML-Datei und anschliessend die Kodierung in das binäre WML-Format. Diese Umwandlung stellt eine starke Komprimierung der Datenmenge dar. Da WML-Dateien in der Regel reine Textdateien sind, schrumpft dabei die zu übertragende Datenmenge bis auf ein Viertel der ursprünglichen Grösse. Dies ermöglicht eine wesentlich schnellere Datenübertragung als das im Internet übliche HTTP-Protokoll.

Aufgrund der geringen Übertragungsgeschwindigkeit (9,6kbps) können so Wartezeiten kurz gehalten werden.

6. Abschliessend erzeugt das WAP-Gateway eine WAP-Response und sendet diese an den Client.
7. Der Client empfängt die WAP-Response, arbeitet das binäre WML ab und stellt die erste Card des WML-Decks dar.

Liefert der Web-Server anstelle von WAP-Inhalten – in Form von WML oder WMLScript – „normale“ WWW-Inhalte, zum Beispiel HTML-Seiten, so kann ein HTML-Filter innerhalb des WAP-Gateway für die mehr oder weniger automatische Transformation der Inhalte genutzt werden.

7. WML – Die Sprache des WAP

So wie der Internetzugang HTTP mit der Programmiersprache HTML arbeitet, so steht hinter WAP die Programmiersprache WML (Wireless Markup Language), die aufgrund der Einschränkungen der mobilen Endgeräte entwickelt wurde und es ermöglicht, sich Informationen aus dem Internet zum Mobiltelefon zu transportieren. WML ist von XML abgeleitet und ist HTML ähnlich. Zu WML gehört WML-Script, eine Programmier-sprache für Anwendungen auf dem Client.
Der WAP Standard wurde speziell für die mobile Datenkommunikation und kleine Endgeräte entwickelt. Deshalb sieht eine WML-Seite auch komplett anders aus als eine WWW-Seite. Vor allem Textzeilen und Links werden im Display gezeigt, weshalb auf hochauflösende und graphisch aufwendige Webseiten verzichtet wird.

8. Die WML-Struktur

Im Gegensatz zu HTML-Seiten, bei welchen jede Seite in einer eigenen Datei gespeichert ist, werden bei XML mehrere Seiten in einer einzigen Datei gespeichert, dem sogenannten WML-Deck. Eine gesamte Site wird hier also als Deck bezeichnet.
Ein Deck (Stapel) besteht aus einer oder mehreren Cards (Karten). Es wird immer ein komplettes Deck auf das WAP Gerät geladen, das logisch zusammengehörige Cards enthalten sollte (siehe dazu Abbildung 3).
Eine Card ist der Teil eines Decks, mit dem der Benutzer navigiert. Er springt also von einer <card> zur anderen - die sich entweder auf einem Deck (einer Datei) zusammen oder in unterschiedlichen Dateien befinden.

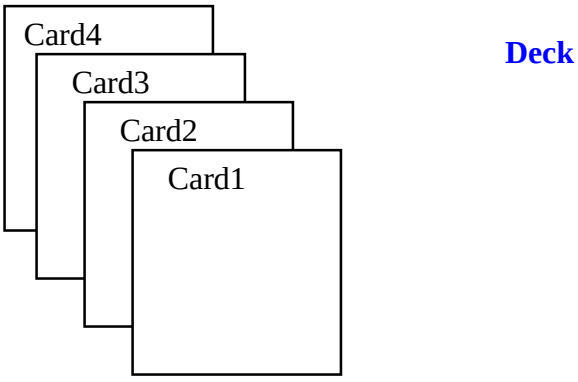


Abb. 3: WML-Struktur

Da WML eine von XML abgeleitete Sprache ist, besteht jedes Element aus einem öffnenden und einem abschließenden Tag.
Das Deck enthält ein <wml> Tag am Anfang und ein abschließendes </wml> am Ende.
Damit ist WML auch case sensitive, d.h. Tags müssen klein geschrieben werden.

Jedes gültige WML Dokument muss einen sog. Prolog besitzen, der auf die verwendete XML-Version und WML-Dokumenten-Beschreibung (DTD, hier Version 1.1) verweist:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

Abbildung 4 zeigt als Beispiel ein einfaches WML-Deck, das eine einzelne Card enthält:

```
1. <?xml version="1.0"?>
2. <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
   "http://www.wapforum.org/DTD/wml_1.1.xml">
3. <wml>
4. <card id = "Card1" title = "Wap.com">
5. <p>
6. Hello World
7. </p>
8. </card>
9. </wml>
```

Abb. 4: WML-Beispiel

Kommentar:

1., 2.: Prolog
3., 9.: Jedes WML-Deck wird durch den wml-Tag definiert und beendet.
4., 8.: Jede WML-Card wird durch den card-Tag definiert und beendet.
id: Die Card-Id, wird als Sprungziel verwendet.
title: Der Titel; er ist dazu gedacht, dem Benutzer Informationen bezüglich der Seite zu geben.
5.-7.: Einzublendender Text, umrandet vom Paragraph-Tag.

Auf dem Display wird folgendes zu sehen sein:

Abb. 5: Resultat auf Display

9. Beurteilung und Aussichten

Durch WAP erlangen wir grössere Mobilität im Bezug auf die Internetnutzung. Die Navigation erfolgt wie vom Computer gewohnt, es sind also keine weiteren Umstellungen notwendig. Allerdings hat WAP bis jetzt nicht den erhofften Durchbruch erzielt.

Nachteile wie:

- geringe Übertragungsgeschwindigkeit (9,6kbs)
- nur WAP unterstützte Webseiten können vom Kunden genutzt werden
- es sind keine aufwändigen Graphiken, Töne und Videosequenzen übertragbar

verursachen Unsicherheit, ob sich dieser neue Standard auch in der Zukunft durchsetzen wird oder nicht und lassen lediglich eine fragliche Akzeptanz bei den Kunden aufkommen.
Nun setzt man auf die Zukunft: Eine neue Übertragungstechnik, GPRS (General Packet Radio Service), soll die Abrechnung nach der Menge der übertragenen Daten ermöglichen. Damit wäre es auch möglich, ständig Online zu sein, ohne zusätzliche Kosten zu haben. Zudem lassen sich mit GPRS wesentlich höhere Übertragungsraten (bis zu 115kbps) realisieren als mit der herkömmlichen GSM-Technik.
Schon im Herbst 2001 soll das WAP 2.0 auf den Markt kommen, welches den japanischen Standard i-mode (ähnlich zu WAP, unterstützt jedoch Farbe und ist HTML-näher) implementiert und abwärts kompatibel zu früheren WAP-Versionen ist. Im Gegensatz zu Japan wird i-mode in Europa in Kombination mit GPRS eingeführt, macht also schon von der höheren Übertragungsrate gebrauch.
Jetzt bleibt nur noch eine Frage offen: Wird der neue Standard WAP zum Durchbruch verhelfen?

Literaturverzeichnis

[1] Andreas Hitzig: *Drahtlos surfen mit Volldampf*; iX 10/99
[2] Lars Röwekamp: *Handy HTML*; iX 2/2000
[3] Wireless Application Forum: *WAP: Wireless Application Protocol*; White Paper, October 1999
[4] Diverse Internet-Seiten:
1. www.ccwap.de/WAP_einfuehrung_%202.htm
2. www.hyperwave.fh-brandenburg.de/kettmand/Medienwirtschaft-WAP.htm
3. www.wap-wissen.de/WAP-Forum/wap-forum.html
4. www.ccwap.de/WAP_forum_member1.htm
5. www.heise.de/newsticker/data/dwi-29.05.01-001/