Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Burkhard Stiller, Pascal Kurtansky (Edt.)

PPS-Seminar: Grundlagen der Internet-Technologie 5

TIK-Report
Nr. 141, July 2002

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Burkhard Stiller, Pascal Kurtansky (Edt.):
PPS-Seminar: Grundlagen der Internet-Technologie 5
July 2002
Version 1
TIK-Report Nr. 141

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 5

Einleitung

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

Ablauf

Die Studierenden erarbeiteten wie in den vergangenen Semestern dieses Mal zu 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. Erstmalig wurde den Studierenden dieses Semester ein zu verwendendes Word-Template abgegeben, um die Form der Ausarbeitungen optisch ansprechend zu gestalten und zu vereinheitlichen.
Neben dieser schriftlichen Arbeit, musste jeder Studierende einen Vortrag von genau 15 Minuten halten – mit einer maximalen Überzeit von zwei Minuten. Die Studierenden sollten lernen, in dieser Zeit den technischen Sachverhalt zusammenzufassen, das Essentielle herauszuschälen und anschliessend prägnant zu präsentieren. Eine nachfolgende kurze Diskussions- und Fragephase erlaubte das interaktive Behandeln von Unklarheiten, offenen Fragen sowie die Verknüpfung von den verschiedenen Themen.

Vorträge, Referenten und Titel

Vortrag	1	Adrian Hottinger	Grundlagen des Internet
Vortrag	2	Philippe Wüger	Netzwerktechnologien für das Internet
Vortrag	3	Martin Pfister	IP, Adressierung und Routing im Internet
Vortrag	4	Simon Solenthaler	IPng – Die nächste Generation des Internet Protokolls
Vortrag	5	Benjamin Marti	MobileIP
Vortrag	6	Sascha Miskovic	TCP/UDP
Vortrag	7	Simon Gregor Wrann	Das HTTP-Protokoll
Vortrag	8	Mischa Demarmels	Die Beschreibungssprache HTML
Vortrag	9	Severin Hafner	Die Datenstrukturierungssprache XML
Vortrag	10	Michael Schnellmann	Sichere Kommunikation – SSL, SHTTP
Vortrag	11	Bernhard Wasser	Elektronische Post im Internet

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Grundlagen des Internet

Adrian Hottinger
hoadrian@ee.ethz.ch
April 2002

1 Einführung

Das Internet ist in aller Munde. In kürzerster Zeit hat sich aus einem einst für Militär und Forschung gedachten Netzwerk eine weltumspannende Kommunikationsplattform entwickelt, deren private wie auch kommerzielle Nutzung scheinbar keine Grenzen kennt. Von den einen als Verwirklichung der freien Meinungsäusserung und Gleichstellung gepriesen, von anderen als Ansammlung pornografischer und gewaltvollen Inhalten verschrien und von kommerziellen Anbietern mit Werbung überschüttet ist das „Netz der Netze“ ein fester Bestandteil der (westlichen) Gesellschaft geworden. Dennoch ist den meisten Nutzern nicht bewusst, welche Technik und welcher Aufwand hinter den drei magischen Buchstaben WWW und dem Internet stecken.

1.1 Definitionen

Durch die Kommerzialisierung des Internets und der daraus folgenden Nutzung durch Konsumenten sind zahlreiche Begriffe und Definitionen der Netzwerktechnologie in die Umgangssprache aufgenommen und in ihrer Bedeutung vereinfacht oder angepasst worden. Dies kann zu Unklarheiten führen.

1.1.1 Technische Definition

Der Begriff Internet („inter“ lat. „zwischen“; „networking“ engl. „vernetzen“) bezeichnet ein weltumspannendes Netzwerk, welches alle Arten von lokalen Netzwerken miteinander verbindet. Somit bildet es die Verbindung von Millionen von Computern. Die Struktur der Verbindungen und der Protokolle ermöglicht es, dass sich, zumindest theoretisch, zwei beliebige Computer, die an dieses Netzwerk angeschlossen sind, problemlos verständigen können.

1.1.2 Dienste im Internet

Basierend auf dem Internet stehen den Benutzern zahlreiche Dienste und Anwendungen zur Verfügung, aufgeführt sind hier nur die wichtigsten:

- **WWW**
  Das sogenannte World Wide Web ist der meistgenutzte Service des Internets. Entgegen der landläufigen Meinung, die Begriffe WWW und Internet seinen gleichbedeutend, kann man das Web als Applikation betrachten, die dem passiven Abfragen von Dokumenten dient. Es basiert auf dem Hypermedia – Prinzip, d.h. die Struktur der Dokumente ist nicht wie in Büchern sequenziell, sondern wird durch miteinander verwobene Informationsstücke gebildet. Diese können unter Umständen global verteilt sein. Man kann Dokumente im WWW als Ansammlung von Informationsbestandteilen ohne feste Reihenfolge betrachten.
  Die Schnittstelle vom Benutzer zum Netz wird mit sog. Browsern gewährleistet. Dies sind grafikbasierte Applikationen, mit denen Hypermedia - Dokumente im Netz beschafft und betrachtet werden können. Dem Umstand der einfachen Bedienung hat dem WWW seinem rasanten Popularitätszuwachs zu verdanken.
- **E-Mail**
  Der zweite vielgenutzte Dienst im Internet ist die elektronische Post. Dank der nahezu verzögerungsfreien Zustellung der „Briefe“ (resp. „Pakete“) und den wegunabhängigen Zustellungskosten erlaubt dieser Dienst eine schnelle und zuverlässige, wenn auch nicht grundlegend sichere globale Kommunikation.
- **Diskussionsforen**
  Während der E-Mail-Dienst eine Ein-Weg-Kommunikation bietet, ermöglicht das Usenet eine eigentliche Diskussion. Eigentlich entwickelt, um wissenschaftliche Erkenntnisse zu verbreiten und zu diskutieren, hat auch dieser Dienst eine öffentliche Nutzung gefunden. So

werden in unzähligen Newsgroups zu allen möglichen Themen Artikel platziert (neudeutsch „geposted“) und diskutiert.

- **IRC – Internet Relay Chat**
  Die direkteste Kommunikation zwischen Nutzern bietet dieser Dienst. In Echtzeit können Gespräche (in geschriebener Form) mit einem oder mehreren Nutzern geführt werden.
- **FTP – File Transfer Protocol**
  Dieses Protokoll bietet die Möglichkeit, Dokumente sicher zu verschicken. Auf Dateien, welche auf FTP-Servern platzierte sind, kann von berechtigten Personen zugegriffen werden.
- **telnet**
  Um aus der Ferne auf Computer zuzugreifen („remote“), wurde dieser Dienst entwickelt. Er erlaubt nebst dem Verwalten von Dokumenten auf dem Zielrechner auch die Nutzung von Programmen auf demselben. Allerdings ist die Verbindung nicht sicher, d.h. der Datenverkehr zwischen den Computer kann „abgehört“ werden.
- **SSH**
  Um den Nachteil des telnet zu umgehen, können Verbindungen zwischen Computern mit der „secure shell“ SSH hergestellt werden. Der so verschlüsselte Datenverkehr kann nicht mehr „abgehört“ werden.

2 Technische Grundlagen

Um die einwandfreie Kommunikation zwischen den doch oft ziemlich verschiedenen Netzwerk- und Computersystemen des Internets zu gewährleisten, braucht es eine klare, einfache Grundkonzeption. Diese sollte sowohl Stabilität, als auch eine einfache Anpassung an neue Technologien und wachsenden Kapazitäten garantieren. Das allgemeine Prinzip eines Netzwerks beruht auf zwei Typen von Computern: Servern, welche Informationen bereitstellt und Kunden oder Clients, welche diese Informationen nutzen.

2.1 Schichtung und Hierarchien

Computernetzwerke sind komplexe Systeme und so ambitionierte Netzwerke wie das Internet sind speziell komplex. Um diese Komplexität zu verwalten, verlassen sich Netzwerkdesigner auf zwei bewährte Prinzipien – Schichtung und Hierarchien.

2.1.1 Schichtungs-Prinzip (Layers)

Die Schichtung einer Problematik bietet den Vorteil, dass die verschiedenen Ebenen strikt getrennt voneinander verarbeitet werden können und auch unabhängig voneinander erneuert werden können. Damit in einem Netzwerk Daten verschickt werden können, muss gewährleistet sein, dass sie an das richtige Ziel geschickt werden und dass sie komplett ankommen. Fig. 1 zeigt, wie ein Datentransfer geschichtet sein könnte.

- Ein Datenpaket ist nichts anderes als eine Reihe von Bits. Damit es während seiner Reise durch das Netzwerk identifiziert werden kann, wird es mit einem eindeutigen Namen versehen. Die dafür nötigen Daten werden üblicherweise vor das eigentliche Dokument angefügt (Schritt 1).
- Um zu gewährleisten, dass die Datei fehlerlos übertragen wurde, wird ein nächster Informationslayer angefügt. Dieser kann sowohl Angaben zur Fehlererkennung wie auch notwendige Daten für eine eventuelle wiederholte Transmission der Datei enthalten, falls ein Fehler entdeckt wurde (Schritt 2).

- Um unter den vielen möglichen Zielgeräten eines Netzwerkes den richtigen zum finden, muss eine weitere Funktion angefügt werden. Diese wird „forwarding“ genannt und fügt wieder die dazu nötigen Information an den Kopf des Dateibündels (Schritt 3).

Die letzten zusätzlichen Informationen dienen der eigentlichen Übertragung. Die Art des Layers ist je nach Typ des Netzwerkes unterschiedlich, beim weitverbreiteten Ethernet ist dies je ein Paket am Anfang und am Ende des Dateibündels (schwarz gekennzeichnet).

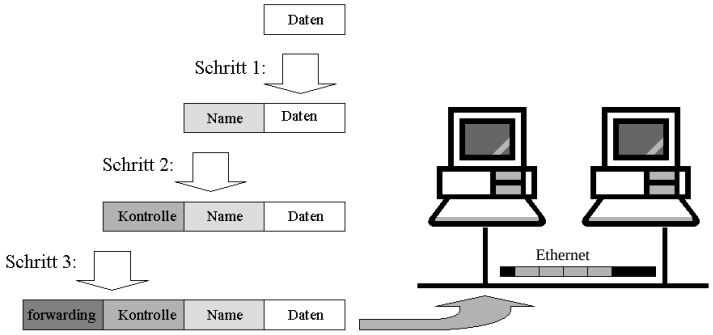

Fig. 1 Schichtung einer Datentransmission

2.1.2 Hierarchie – Prinzip

Wie die Schichtung sind Hierarchien ein weiteres abstraktes Netzwerkkonzept. Sie organisieren Informationen und delegieren Verantwortlichkeit. Vergleichbar ist dieses Prinzip mit dem globalen Telefonnetz. Dieses beruht darauf, dass jeder Anschluss eine eindeutige Adresse, in diesem Fall eine Nummer hat. Somit ist jeder Teilnehmer eindeutig identifizierbar. Jeder Ziffernblock steht dabei für eine Hierarchiestufe – Landeskennzahl, Vorwahl, Gebiets- und Hauptnummer. Der Aufwand, alle Telefonnummern zentral zu verwalten, wäre viel zu gross. Stattdessen existiert eine einfache Hierarchie, welche auf der Geografie basiert. Auch bei diesem Prinzip sind Anpassungen eine lokale Angelegenheit der jeweiligen Hierarchiestufe, die eine wesentlich vereinfachte Verwaltung des ganzen Netzwerkes liefert.

2.2 TCP/IP – das Protokollsystem des Internets

Auch dem Internet liegt das Schichtungs- und Hierarchienprinzip zugrunde. Die Kommunikationsprotokolle TCP („Transmission Control Protocol“) und IP („Internet Protocol“) bilden das Kernstück des Verbindungsaufbaus zwischen Host und Client. Obschon eigentlich keine Normierung der Internetarchitektur, respektive deren Schichtung besteht, lässt sich diese sehr einfach darstellen (Fig. 2). Die Schichtung der Abläufe ist dabei wie ein Stack vertikal angeordnet.

- Die innerste Schicht enthält die Applikationsschicht, sie organisiert die zu transferierende Informationen, dass heisst, in ihr befinden sich die verschiedenen Anwendungen und Dienste. Zu ihr zählen zum Beispiel FTP, telnet oder das DNS („Domain Name System.
- Unterhalb der Applikationsschicht liegt der Transport-Layer, welcher dafür verantwortlich ist, dass die Informationen an ihr Ziel kommen. Ausserdem sorgt er für die komplette Übertragung der Datenpakete. Das mit Abstand wichtigste Protokoll dieser Schicht ist das TCP.
- Das primäre Protokoll der nächsten Schicht ist das Internet Protocol IP. Es sorgt dafür, dass die Daten den Weg durch das Netzwerk finden und zum Zielrechner gelangen. Dieser Vorgang wird „forwarding“ genannt. Um dies zu garantieren, muss das Protokoll Kenntnis über die Topologie des Netzwerkes haben. „Routing“ wird der Prozess genannt, diese Informationen zur Verfügung zu stellen.

- Die unterste Schicht des TCP/IP Stacks bildet die Netzwerktechnologie an sich. Sie bewerkstelligt die Verbindung zwischen den verschieden Netzwerken, die das Internet bilden.

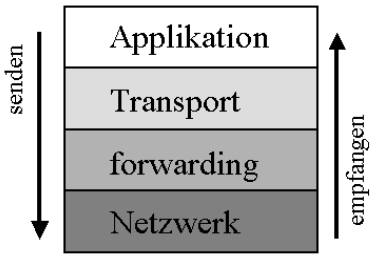

Fig. 2 Aufbau des TCP/IP Stacks

2.2.1 Connectionless und Connection-Oriented Delivery

Connectionless Delivery (C) und Connetion-Oriented Delivery (CO) bezeichnen die Art einer Verbindung zwischen einem Host und einem Client.

- Connetionless Delivery ist die einfachste Art einer Verbindung. Ein Datenpaket wird vom Sender ohne vorherige Absprache mit dem Empfänger abgeschickt. Jedes Datenpaket wird isoliert betrachtet, eine Antwort oder Bestätigung vom Empfänger bleibt aus. Zu vergleichen ist diese Verbindungsart mit der herkömmlichen Briefpost.
- Connection-Oriented Delivery setzt eine Interaktion zwischen Sender und Empfänger voraus. Ähnlich einer Kommunikation zwischen zwei Faxgeräten wird vor dem Versenden des eigentlichen Datenpakets eine Verbindung aufgebaut. Nach der Transmission kann der Empfänger die Vollständigkeit des Pakets bestätigen oder eine Neuübertragung der Daten anfordern.

Für das TCP/IP-Protokollsystem kommen beide Formen zum Einsatz. TCP macht sich die Eigenschaften der Connection-Oriented Delivery zu Nutze, um sich von der Vollständigkeit der Datenübertragung überzeugen zu können. IP hingegen liefert die Datenpakete per Connetionless Delivery aus. Es kommt jedoch während einer Datenübertragung mehrmals und in beiden Richtungen zum Einsatz.

3 Geschichte des Internet

Die Geburtsstunde des Internets lässt sich nicht genau bestimmen. Vielmehr ist es das Produkt einer Reihe von Ideen und Entwicklungen, die das Internet zu dem Technologiestand gebracht hat, den es heute inne hat. Einen grossen Anteil an der Verwirklichung der Idee eines Netzwerks mit Computern ist der Advanced Research Projects Agency (ARPA), einem Institut des amerikanischen Verteidigungsministeriums zuzumessen. In ihrem Auftrag entwickelte die Firma Bolt Beranek and Newman BBN (führende Firmen wie IBM lehnten die Idee als unrealisierbar ab) den Interface Message Processor IMP. Der erste IMP wurde im September 1969 in einem Büro an der Universität von Kalifornien in Los Angeles in Betrieb genommen. Seine Aufgabe bestand darin, Daten zu senden und zu empfangen, den Empfang zu überprüfen und das Senden im Falle eines Fehlers zu wiederholen. Drei weitere IMPs wurden in Stanford, Santa Barbara und Salt Lake City aufgestellt. Am 10. Oktober 1969 war es dann soweit: erstmals wurde ein Datentransfer zwischen zwei unterschiedlichen Computersystem vollzogen (wenn das System auch nach dem dritten

übertragenen Buchstabe abstürzte). Als das Projekt 1971 unter dem Namen ARPA-Net der Öffentlichkeit vorgestellt wurde, waren bereits 15 IMP-Rechner miteinander verbunden. Bald darauf wurde die Idee wach, mittels eines einheitlichen Protokollsystems die Einbindung weiterer Netzwerke mit anderen Standards zu fördern. Schliesslich wurde 1983 das TCP/IP-Protokollpaket als offiziellen Standard eingeführt. Dies führte zu einem raschen Anwachsen der angeschlossenen Rechnern. Als 1989 die Grenze von 100'000 Hostrechnern erreicht wurde, trennte man den militärischen Teil des Netzes ab und schaltete das ARPA-Net ab, das NSFNET wird zum leistungsfähigen Nachfolger. Der eigentliche kommerzielle Durchbruch gelang mit der Vorstellung des World Wide Web, das 1991 vom CERN (European Organization for Nuclear Research) in Genf entwickelt wurde. Innert kürzester Zeit wuchs die Anzahl der Hostrechner auf heute über 147 Millionen (WWW-Sites 38'118’962).

4 Schlussfolgerung

Quellennachweis

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Netzwerktechnologien für das Internet

Philippe Wüger
wuegerp@ee.ethz.ch
26. April 2002

1 Einführung

Im Folgenden soll darüber berichtet werden, auf welchen Netzwerktechnologien das Internet aufbaut und welche Vor- und Nachteile die verschiedenen Systeme mit sich bringen.

1.1 Anforderungen und Protokolle

Die Anforderungen, die das Internet an die Netztechnologien stellt, sind sehr gering: Sie sollen einzig die Möglichkeit bieten, Datenpakete zu übermitteln. Restliche wichtige Faktoren für eine Netzwerkverbindung, wie Verzögerung, Zuverlässigkeit, Durchsatz oder Reihenfolgeerhaltung der einzelnen Pakete, werden durch die sogenannten Protokolle (z.B. IP - Internet Protocol) garantiert.

1.2 Lokale Netze und Weitverkehrsnetze

Grundsätzlich unterscheidet man zwischen zwei Netzwerktypen: Den lokalen Netzen ("Local Area Networks": LANs), welche einzelne Computer innerhalb von Räumen oder Büros verbinden, und den Weitverkehrsnetzen ("Wide Area Networks": WANs), welche sich über Distanzen von mehreren Kilometern erstrecken und die einzelnen LANs verbinden.

2 Local Area Networks

Wie bereits erwähnt, basiert das Internet auf verschiedenen Netzwerksystemen. So gibt es auch unter den LANs verschiedene Technologien, wie z.B. Ethernet, Token Ring oder Wireless LAN. Alle diese haben gemeinsam, dass sie die Rechner innerhalb von Gebäuden lokal vernetzen. Meist geschieht dies mit relativ hohen Übertragungsraten von einigen Mbits/s bis zu 1 Gigabit/s und kurzen Verzögerungszeiten.

2.1 Ethernet

Ein weit verbreiteter Typ eines Local Area Network stellt das Ethernet dar. Die Idee des Ethernet baut auf einer gemeinsam genutzten Datenleitung (auch Bus genannt) auf, an welche alle Computer angeschlossen und somit vernetzt sind, wie Abbildung 1 zeigt.

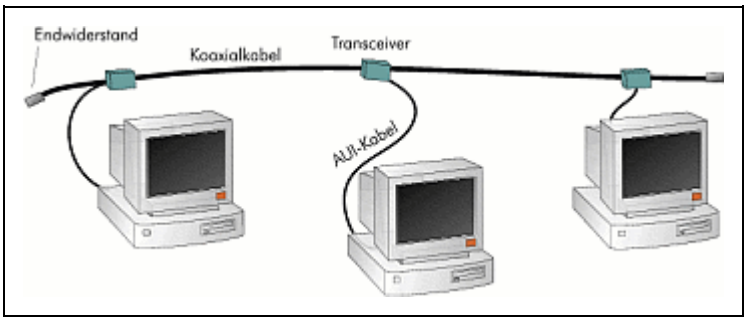

Fig. 1 Gemeinsamer Datenbus eines Ethernets.

2.1.1 Grundprinzip CSMA/CD

Um ein Datenpaket zu übertragen, muss der sendende Rechner diese gemeinsame Leitung zuerst "abhören", um zu sehen, ob nicht bereits eine Datenübertragung stattfindet. Ist die Leitung frei, so schickt der Sender sein Paket. Er überwacht die Leitung jedoch weiterhin, um zu sehen, ob nicht vielleicht ein weiterer Computer gleichzeitig damit begonnen hat, Daten zu übermitteln. In diesem Fall würde man von einer Kollision sprechen. Beide Sender würden die Übermittlung sofort beenden und nach einer zufälligen Zeit einen neuen Versuch untenehmen. Führt dies zu einer erneuten Kollision, so erhöht sich die zufällige Wartezeit jeweils. Dieses Prinzip wird mit dem Namen Carrier Sense (Abhören der Datenleitung) Multiple Access / Collision detection (Kollisionserkennung), also kurz CSMA/CD zusammengefasst.

2.1.2 Einschränkungen

Das beschriebene Prinzip birgt die Einschränkung, dass "die Ausdehnung des Mediums nur so gross sein darf, dass zwei gleichzeitig zu senden beginnende Endsysteme, die jeweils am Ende des Mediums lokalisiert sind, die Übertragung eines Pakets mit minimaler Länge noch nicht eingestellt haben, wenn die vom jeweils anderen Endsystem erzeugten Signale eintreffen."[2] Für das Ethernet resultiert dies in einer maximalen Ausdehnung von ca. 2,5 km bei einer minimalen Paketlänge von 64 Bytes.

2.1.3 Entwicklung des Ethernet

Die heute wahrscheinlich am meisten verbreitete Variante des Ethernet trägt den Namen "10BaseT" oder "100BaseT", je nach Übertragungsgeschwindigkeit. Im Gegensatz zu den Vorgängern "Thick Ethernet" und "10Base2" wird das als Datenbus verwendete Coaxialkabel in einem Hub zusammengefasst, an welches alle Computer wie in Abbildung 2 sternförmig angeschlossen werden. So wird die Verbindung auch noch garantiert, falls an irgendeiner Komponente eine Störung auftritt. Die älteren Systeme waren viel fehleranfälliger: Wurde eine Verbindung unterbrochen, so war der Bus für alle Computer nicht mehr nutzbar, da sie alle seriell an der Leitung angeschlossen waren.

Eine weiter Form des Ethernets stellt das Gigabit Ethernet dar, welches Übertragungsraten bis zu 1 Gbit/s erlaubt. Dieses wird vor allem als Backbone-Verbindung zwischen Netzwerken eingesetzt und benötigt speziell abgeschirmte Kabel oder Glasfasern.

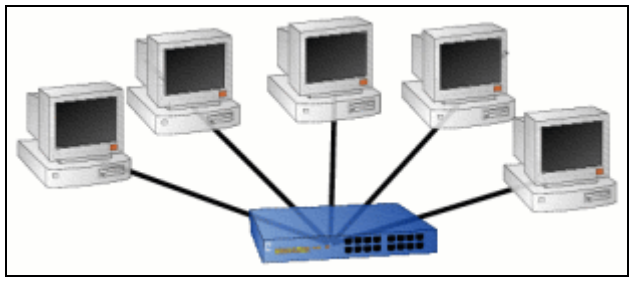

Fig. 2 "10BaseT" / "100BaseT"-Ethernet mit Verbindungen über ein Hub.

2.1.4 MAC-Datenformat

Die Datenpakete im Ethernet (und allgemein in LANs) sind entsprechend dem MAC-Datenformat (siehe Abbildung 3) aufgebaut: Ein Medien-spezifisches MAC-Control-Feld macht den Anfang, darauf folgen Quell- und Zieladresse, die Länge des Paketes, dann die eigentlichen Daten und schliesslich noch ein Prüffeld. Die Adressen jedes Systems sind einmalig, damit es eindeutig identifizierbar ist. Sie werden der Hardware durch den Hersteller fest zugewiesen.

MAC Control	Source Address	Destination Address	Length	Daten + Padding	Frame Check Sequence
variabel	48	48	16 (Ethernet)		bit

Fig. 3 Das MAC-Datenformat

2.2 Token Ring

Beim CSMA/CD-System des Ethernets, ist man kaum in der Lage vorauszusagen, wielange die Übertragung eines Datenpakets dauern wird, da immer wieder Kollisionen auftreten können. Beim "Token Ring" umgeht man dieses Problem, indem man ein sogenanntes Token (im Prinzip auch ein Datenpaket) in dem kreisförmigen Netz (Abbildung 4) herumgibt. Mithilfe dieses Tokens wird das Senderrecht geregelt. Ein Computer darf nur senden, falls dieser ein freies Token empfängt, da man immer nur ein Datenpaket im Ring haben will. Der Sender kann das freie Token auf "besetzt" setzen und sein Datenpaket verschicken. Der Empfänger macht sich eine Kopie der Daten, die dann vom Sender wieder aus dem Ring genommen werden. Danach schickt der Sender ein freies Token ins Netz, welches vom nächsten Computer aufgenommen wird. So können die einzelnen Stationen abwechslungsweise ihre Daten versenden. Das Übermitteln eines Paketes dauert höchstens solange, wie ein Durchlauf eines Tokens.

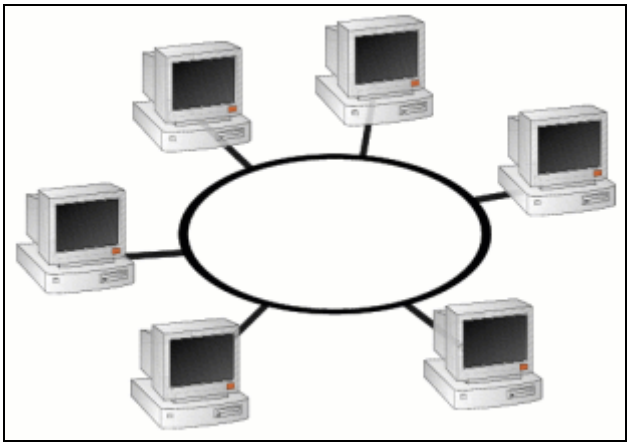

Fig. 4 Token Ring

2.3 FDDI

Das Fiber Distributed Data Interface, kurz FDDI, ist wie Token Ring auch ringförmig aufgebaut. Anstatt über Kabel, sendet es die Daten über Glasfaserleitungen, welche nicht durch elektromagnetische Felder gestört werden können und auch höhere Übertragungsraten erlauben. Dazu besteht ein zweiter Ring, welcher genutzt wird, falls irgendwo eine Hardwarestörung auftritt. Die benachbarten Stationen des defekten Ringabschnitts registrieren, dass keine Pakete mehr an die fehlerhafte Hardware gesendet werden können und leitet den Datenverkehr automatisch auf den zweiten Ring um, wie Abbildung 5 rechts zeigt.

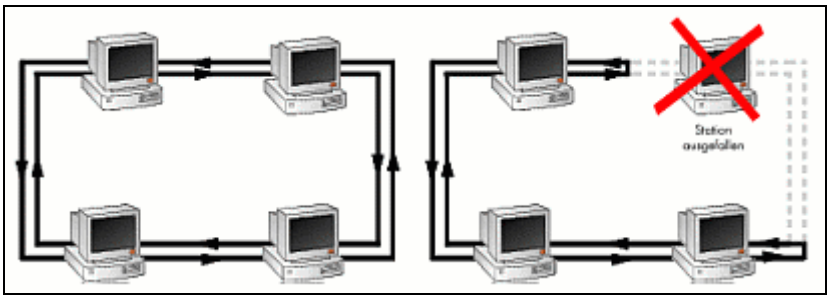

Fig. 5 FDDI - Netz. Rechts mit Umleitung auf den zweiten Ring

Ähnlich wie beim Token Ring, werden die Senderechte mit Tokens verwaltet. Diese bilden aber nun Teil eines Datenpakets. Somit wird es möglich, gleichzeitig mehrere Pakete auf dem Ring zu haben.

3 Wide Area Networks

Im Gegensatz zu den LANs operieren die Weitverkehrsnetze verbindungsorientiert. Das bedeutet, dass vom Sender zum Empfänger eine feste Verbindung hergestellt wird, also eine genau definierte "Datenroute". Bei den LANs werden die Pakete meistens einfach über das ganze Netzwerk "geflutet", so dass sie ihren Bestimmungsort eher zufällig erreichen.

3.1 X.25

X.25 benützt zur Datenübertragung virtuelle Kanäle, welche zwischen zwei Datenendeinrichtungen (DEE) in der Datenübertragungseinrichtung (DÜE) erstellt werden. Dazu müssen vor der eigentlichen Datenübertragung bestimmte Verbindungsaufbaunachrichten gesendet werden. Ist der Kanal erstellt, so können die Daten im Vollduplexmodus ausgetauscht werden, also gleichzeitig in beide Richtungen. Die Übertragung wird dabei durch Fluss- und Fehlerkontrollfunktionen überwacht. Geht ein Paket verloren, so wird es einfach nochmals übertragen.

Die Datenpakete enthalten neben den Sende- und Empfangsfolgenummern auch die Kanalnummern, welche die genaue Verbindung beschreiben, das heisst es enthält Informationen über den vordefinierten Weg.

3.2 Integrated Services Digital Network (ISDN)

ISDN ist heutzutage eine sehr weit verbreitete Technologie. Die analogen Telefonverbindungen werden durch digitale ersetzt, welche mit einer höheren Geschwindigkeit, Zuverlässigkeit und Flexibilität aufwarten können, was vor allem dem Übermitteln von Daten zugute kommt. Insbesondere für den Datenaustausch über TCP/IP bildet ISDN eine gute Grundlage. Im Folgenden wird auf das "Narrowband ISDN" eingegangen, welches vor allem auf dem bestehenden Telefonnetzwerk aufbaut und sich vom Breitband-ISDN (ATM genannt) unterscheidet.

3.2.1 Grundprinzip

Im Unterschied zu X.25, arbeitet ISDN nicht paketvermittelnd, sonder leitungsvermittelnd. Zwischen Sender und Empfänger wird eine Verbindung mit einer festen Bitrate von 64 kbit/s erstellt. Für Datenübermittlung und Kontrolle, bzw. Signalisierung, stehen zwei verschiedene Kanal-Typen zur Verfügung. Der Datenaustausch findet über den sogenannten B-Kanal statt, wobei der D-Kanal die Signalisierungsfunktionen übernimmt. Diese beinhalten an erster Stelle den Verbindungsauf- und Abbau, also zum Beispiel das Übermitteln der Telefonnummer.

Für den Endnutzer stehen zwei verschiedene ISDN-Setups zur Verfügung (Abbildung 6). Das Basic Rate Interface (BRI), welches 2 B-Kanäle und einen D-Kanal enthält, und das Primary Rate Interface (PRI) mit 30 B-Kanälen und einem D-Kanal. Durch Kanalbündelung können so Übertragungsraten von bis zu 128 kbit/s (BRI) und 1.92 Mbit/s (PRI) erzielt werden.

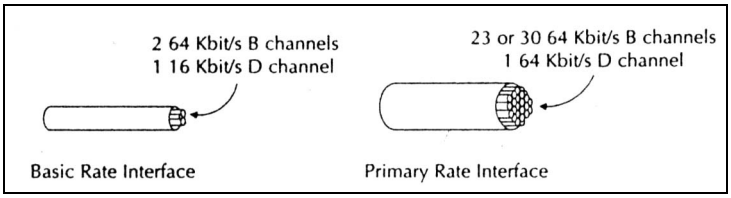

Fig. 6 Basic Rate Interface und Primary Rate Interface

3.2.2 Point-to-Point Protocol (PPP)

Arbeitet Schmalband-ISDN in Verbindung mit einem TCP/IP Netzwerk, so kommt das PPP-Protokoll zum Einsatz. Dieses besteht aus mehreren Unterprotokollen, welche alle für verschiedene Phasen einer PPP-Verbindung stehen:

- Link Control Protocol: Erstellt und Verwaltet eine Verbindung.
- Authentication Protocol: Erlaubt die Verifizierung der Identität des anderen Systems.
- Challenge Handshake Authentication Protocol: Höhere Sicherheit bei der Systemidentifizierung durch Verschlüsselung.
- Weitere Protokolle zur Kontrolle des Netzwerks.

Die verschiedenen Phasen eines Verbindungsverlaufs mit PPP illustriert folgende Abbildung 7:

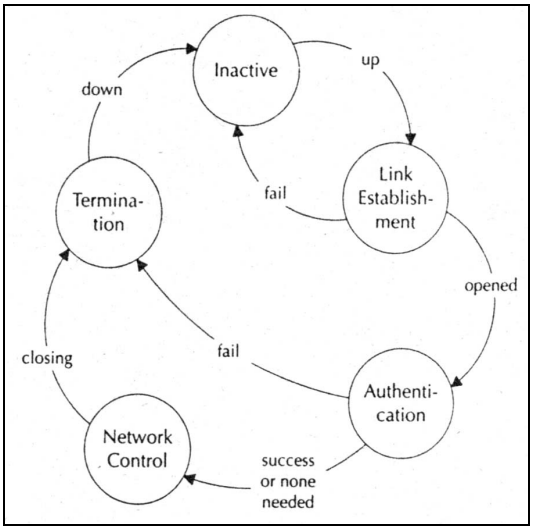

Fig. 7 Verlauf einer PPP-Verbindung

3.3 Asynchronous Transfer Mode (ATM)

Ursprünglich wurde ATM für Breitband-ISDN, also für Weitverkehrsnetze konzipiert. Heutzutage ist es aber auch im lokalen Bereich weit verbreitet, wo es Bitraten von bis zu 622 Mbit/s erreicht. Grundsätzlich ist das Funktionsprinzip ähnlich wie bei Schmalband-ISDN, es arbeitet auch verbindungsorientiert. Je nach verwendeter Netzwerktechnologie können aber viel höhere Transferraten (bis Gbit/s) erzielt werden.

3.3.1 ATM-Zelle

Die Daten werden bei ATM in einzelne Zellen aufgeteilt und anschliessend übertragen. Um eine möglichst hohe Verarbeitungsgeschwindigkeit in den Switches (den Knoten des Netzwerks) zu gewährleisten, ist die Länge der Zelle fest definiert und immer gleich: Insgesamt 53 Bytes umfasst ein solches Paket, wobei 5 Byte Kontrollinformationen enthalten und die restlichen 48 Bytes die zu übertragenden Daten beinhalten. Im Header der Zelle befinden sich Informationen wie Kanalkennung, Zugriffssteuerung und Prüfsumme.

3.3.2 Dienste

Ein wesentlicher Unterschied zum Schmalband-ISDN äussert sich in der grösseren Flexibilität bezüglich Nutzung des Netzes. Je nach Anwendung kann von den Kommunikationsteilnehmern eine "massgeschneiderte" Verbindung erstellt werden. So wird keine Bandbreite verschwendet, falls wie beim Schmalband-ISDN zwar eine feste Bitrate reserviert ist, aber nur ein kleiner Teil davon auch genutzt wird. Die fünf folgenden Dienstkategorien stehen für eine Verbindung zur Verfügung:

- Konstante Bitrate, z.B. für ISDN-Telefonie mit 64 kbit/s.
- Variable Bitrate für Realzeitdaten, z.B. für die Übertragung von MPEG-komprimierten Videodatenströmen, welche starke Schwankungen in der Bitrate aufweisen.
- Variable Bitrate für Nicht-Realzeitdaten, z.B. für Reservierungssysteme oder Transaktionen.
- Unspezifizierte Bitrate, z.B. für LANs und Dateitransfer.
- Verfügbare Bitrate. Die minimale Bitrate wird zwecks besserer Verbindungsqualität festgelegt. Z.B. auch für LANs und Dateitransfer.

4 Schlussbemerkung

Der Wunsch nach einer möglichst flexiblen Nutzung eines Datennetzes resultiert in einer Vielzahl von Protokollen und Technologien. Diese erlauben eine auf die spezifische Anwendung zugeschnittene, optimale Lösung. Bestimmende Faktoren sind hier Übermittlungsgeschwindigkeit, Verzögerung, Reihenfolgeerhaltung der Datenpakete, Fehleranfälligkeit, Flexibilität bezüglich Änderungen in der Netzwerktopologie, Verbindungsmöglichkeiten mit anderen Systemen und nicht zuletzt auch die Kosten. Die wichtigsten Netzwerktechnologien werden hier noch einmal aufgelistet:

- Je nach Distanz unterscheidet man zwischen Local Area Networks (LANs) und Wide Area Networks (WANs).
- Zu den LANs gehören: Ethernet, Token Ring und Fiber Distributed Data Interface (FDDI). Daneben gibt es auch noch weitere Technologien, wie zum Beispiel das Wireless Lan (WLAN).
- Als WANs kommen hauptsächlich Integrated Services Digital Network (ISDN) und Asynchronous Transfer Mode (ATM) in Frage, wobei ATM grössere Flexibilität und Geschwindigkeit anbietet und so oft auch in LANs eingesetzt wird.

Quellenangabe

1. S.Thomas: Ipng and the TCP/IP Protocols. J.Wiley & Sons Inc. New York.
2. T.Braun: Ipng - Neue Internet Dienste und virtuelle Netze. dpunkt Verlag, Heidelberg, Deutschland, 1999.
3. www.physio.mu-luebeck.de/schulung/_private/netzwerk

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Internet Protocol (IP), Adressierung und Routing

Martin Pfister
martpfis@ee.ethz.ch
17. Mai 2002

1 Einleitung

Eine der wichtigsten und grundlegendsten Fragen über Netzwerke ist jene, wie denn überhaupt die Daten von einem Computer zum anderen gelangen. Wie werden die Daten übermittelt? Warum kommen sie am richtigen Ort an? Welchen Weg nehmen sie durch die Netzwerkstrassen und was geschieht wenn ein „Unfall“ passiert? Wie können verschiedenste Rechner-Arten miteinander kommunizieren?

Wir wissen, dass im Internet – unserem Beispiel-Netzwerk – versandte Daten tatsächlich ankommen und die obigen Fragen irgendwie beantwortet werden können. Es muss demnach etwas geben das diese Hürden überwindet und es ermöglicht ein Netzwerk zu betreiben.

Dieses „etwas“ heisst Internet Protocol oder kurz IP. Die aktuelle IP-Version ist die Version 4 (IPv4). IP kommt bei allen Internet-Anwendungen zum Zuge, also z.B. beim aufrufen von Websites mit einem Browser, beim E-Mail-Verkehr oder auch FTP-Downloads usw. IP wird meistens im Zusammenhang mit TCP (Transfer Control Protocol) genannt.

2 Internet Protocol – IPv4

IP ermöglicht es nicht nur innerhalb von kleinen Netzwerken für den Datentransport zu sorgen. Erst durch IP war es möglich, verschiedene Netzwerke zusammenzuschliessen und damit das Internet zu verwirklichen. Es stellt deshalb sozusagen die Basis des Internets dar, etwas das alle Teilnehmer im Netz verstehen. Durch das Internet Protocol lassen sich folgende Anforderungen erfüllen:

- Segmentierung und Reassemblierung
- Adressierung
- Routing
- Löschen von abgelaufenen Datenpaketen
- Fehlererkennung

Eine wichtige Eigenschaft vom IP stellt die verbindungslose Übertragung dar. Die Datenpakete werden nacheinander losgeschickt ohne die Gegenstelle zu verifizieren oder eine direkte Verbindung aufzubauen. Da jedes Paket (Datagramm) einzeln geroutet wird, kommen die Pakete vielfach nicht in der Reihenfolge an, wie sie abgeschickt wurden. Von Vorteil ist aber, dass während der Übertragung die Pakete schnell und flexibel auf Ausfälle oder Überlastungen im Netzwerk reagieren können, egal welchen Weg die Vorangehenden bereits genommen haben.

2.1 Segmentierung / Reassemblierung

Für die Übertragung von Daten werden diese Segmentiert und in kleineren Einheiten verschickt. Dies ist bedingt durch die Grenzen, welche von der Software- oder Hardware gegeben sind so wie auch durch verwendete Standards oder Überlegungen betreffend der Übertragungsqualität bzw. Fehlerquote. Es gibt deshalb auch eine maximale Länge für IP-Datenpakete. Die Daten werden demnach beim Sender segmentiert und losgeschickt und erst beim Empfänger wieder reassembliert, d.h. wieder zusammengesetzt.

Feldbezeichnung	**Bedeutung**
version	Version des Protokolls
precedence	Priorität
total length	Gesamtlänge

fragment offset	Wert zur korrekten Reassemblierung
time-to-live	Lebenszeit
protocol	Art des Transport-Protokolls (z.B. TCP)
header checksum	Prüfsumme des Headers
source adress	Startadresse das Datagramms
destination adress	Zieladresse des Datagramms
options	Optionen

Tab. 1 Die wichtigsten Informationen in einem Header (unvollständig).

Jedes Datagramm enthält neben den Daten einen so genannten Header („Kopf“), welcher die Informationen enthält um das Paket korrekt zu routen (vgl. Tab. 2). Es sind diese Header-Daten, die das Internet Protocol eigentlich ausmachen. Darin ist z.B. die Zieladresse, die Quelladresse oder die Lebenszeit zu finden.

2.2 Lebenszeitkontrolle

Wie bereits beschrieben handelt es sich bei IP um ein verbindungsloses Protokoll. Es kann deshalb vorkommen, das Datagramme ihr Ziel gar nie erreichen aber nach wie vor im Netz „herumirren“. Solche Pakete sind unerwünscht und belasten das Netzwerk unnötig. Es braucht deshalb einen Mechanismus, um solche Datagramme aus dem Netz zu entfernen: Im time-to-live-Feld im Header eines IP-Paketes wird eine Zahl festgelegt. Diese definiert die maximale Anzahl Router, die das Paket während der Reise durch das Netz passieren darf (Hop-Count). Nun wird in jedem durchlaufenen Router die Zahl um eins dekrementiert. Sobald die ttl-Zahl auf Null ist, wird das Datagramm gelöscht und es wird über ICMP (Internet Control Message Protocol) eine Fehlermeldung abgesetzt.

2.3 Fehlererkennung

Die Fehlererkennung ist bei IP sehr beschränkt, was es zusammen mit dem Prinzip der Verbindungslosigkeit ein relativ unsicheres Protokoll macht. Wie man aus Tab. 1 entnehmen kann (header checksum), ist die Erkennung von Fehlern auf den Header begrenzt, die eigentlichen Daten werden also nicht berücksichtigt. Da sich der Header auf dem Weg z.B. durch den time-to-live-Wert oder durch Weginformationen ständig verändert, muss die Prüfsumme in jedem Knoten auch wieder neu berechnet werden. Mankos in diesem Bereich werden bei IPng ein Thema sein.

3 Adressierung

Im obigen Kapitel war schon von Adressen die Rede. Doch was hat es mit diesen Adressen auf sich? Es soll hier verständlich gemacht werden, wie die unzähligen Rechner im Netz alle einzeln erreichbar werden und wie das Internet strukturiert ist.

3.1 Das IP-Adresssystem

In einem Computer-Netzwerk muss jeder Rechner eindeutig identifizierbar sein, damit die Zustellung der Datenpakete sichergestellt ist. Wie jedes Haus durch Ort, Strasse und Nummer definiert ist, besitzt jeder Rechner im Netz eine IP-Adresse. Bei IPv4 hat eine IP-Adresse die Länge von 32 Bit oder vier Byte. Man stellt IP-Adressen so dar, indem man jedes der vier Bytes als Dezimalzahl schreibt und die vier Zahlen durch Punkte trennt, wie z.B. 129.132.2.198.

Klasse	Beginn (binär)	Adressierung	Anzahl Netze	Anzahl Rechner
A	0…	**xxx**.xxx.xxx.xxx	128	16777216
B	10…	**xxx.xxx**.xxx.xxx	16384	65536
C	110…	**xxx.xxx.xxx**.xxx	2097152	256

Tab. 2 Die drei wichtigsten Klassen von IP-Adressen.

Wie ein Brief durch die Postleitzahl erst mal grob sortiert wird, bezeichnet der erste Teil einer IP-Adresse das Netz, in welchem das Datagramm „abgeliefert“ werden soll. Die restlichen Bits bezeichnen den Rechner („Strasse/Hausnummer“). Um den verschieden Anforderungen und Grössen von Netzen gerecht zu werden, wurden verschiedene Klassen von IP-Adressen eingeführt (vgl. Tab. 2). Der fett gedruckte Teil symbolisiert den Bereich, der für die Netzerkennung reserviert ist, der normal gedruckte die Rechneradresse. Um die Adressen unterscheiden zu können sind ihnen unterschiedliche Zahlenbereiche zugeordnet worden (vgl. zweite Spalte). Klasse A-Netze kann es nur 126 geben (erstes Byte zwischen 1 und 126), dafür kann ein solches Netz fast 17 Millionen Rechner enthalten. Das lohnt sich natürlich nur für riesige Organisationen wie das z.B. für das US-Militär. Analog kann man sich das für die anderen Klassen überlegen. Es ist zu beachten, dass die Zahlen in den letzen zwei Spalten theoretische Werte sind, denn gewisse Adressen sind reserviert und somit nicht zur Vergabe verwendbar. Weiter gibt es auch noch Klasse D- und E-Adressen, die für Multicast bzw. Testzwecke benötigt werden.

Ein Problem bereitet die beschränkte Anzahl Adressen, denn das Internet wächst ständig und durch die inflexible und zum Teil unpraktische Einteilung in Klassen gehen Adressen für die Nutzung verloren. Es gibt jedoch Methoden zur besseren Ausnutzung der Adressen, wie zum Beispiel CIDR (Classless Inter-Domain Routing). So kann man z.B. für ein Unternehmen mit 1000 Rechnern vier zusammenhängende Klasse C-Blöcke verwenden (1024 Adressen), anstatt eine Klasse B-Adresse zu „verschwenden“. Eine definitive Lösung zur Knappheit der IP-Adressen kann aber wohl erst mit IPng behoben werden.

3.2 Subnetze

Grössere Institutionen besitzen vielfach auch grosse Netze, die schwer zu warten sind. Um Verantwortlichkeiten aufzuteilen und die Verwaltung zu erleichtern besitzt man die Möglichkeit Subnetze einzurichten. Subnetze sind lokale Unternetze in einem bestehenden grossen Netz. Realisiert wird ein Subnetz mit einer so genannten Subnetzmaske. Diese wird dann mit den IP-Adressen mittels einer logischen UND-Funktion verknüpft. Beschreibt z.B. 9.0.0.0 ein Klasse A-Netz, dann wäre die Adresse 9.228.0.0 mit der Subnetzmaske 255.255.0.0 ein Subnetz des Klasse A-Netzes. Für die Adressierung wurde in diesem Fall das ganze zweite Byte verwendet. Es ist auch möglich, Subnetze in Subnetzen zu erstellen.

3.3 DHCP

DHCP steht für Dynamic Host Configuration Protocol und existiert auch nicht zuletzt wegen der Knappheit an IP-Adressen. Jedoch bietet DHCP auch viele andere Vorteile. Durch DHCP lassen sich in einem Netzwerk Hostadressen dynamisch verteilen sowie Zugriffsrechte und Kompatibilitäten regeln. Dockt man z.B. mit einem Laptop an einem Subnetz der ETH an, bekommt man dynamisch eine Adresse zugeteilt, die während der Dauer der Sitzung aber nicht wechselt. Bei der nächsten Sitzung kann aber irgendeine andere Adresse zugeteilt werden. In grossen Netzwerken bedeutet dies auch eine beachtliche Verringerung des Verwaltungsaufwands, da die Adressen nicht manuell gemanagt werden müssen.

3.4 Domain Name Service (DNS)

Der Domain Name Service bietet in Zusammenhang mit IP-Adressen einen wichtigen Dienst. Als Mensch würde man sich beim merken oder aufschreiben von IP-Adressen wohl ziemlich schwer tun, darum ordnet DNS den IP-Nummern Textadressen zu, welche für uns leichter zu handhaben sind. Das abrufen von Homepages ist ja bekanntlich eine Hauptapplikation des Internets und auf diese Weise ist es uns möglich, in einem Browser Text statt Zahlen einzugeben. Die DNS-Adressen sind in einer hierarchischen Baumstruktur organisiert. Die so genannten Top-Level-Domains wie „ch“ werden von eigenen Organisationen verwaltet, welche selbständig die Subdomains vergeben. Die Subdomains bezeichnen häufig Namen von Firmen oder Organisationen. Es ist auch hier möglich, Subdomains weitere Subdomains zuzuordnen (z.B. www.ee.ethz.ch). Auf der untersten Ebene befindet sich dann der Name des Rechners.

4 Routing

Wie ein Netzwerk aufgebaut ist und welche Funktionen Internet Protocol bereitstellt wissen wir nun. Jedoch ist immer noch die Frage offen, wie die Daten zum Ziel gelangen. Sobald nämlich Datagramme an eine IP-Adresse ausserhalb des eigenen Subnetzes verschickt werden sollen, kennt man den Empfänger ja vorerst nicht. Dann tritt der Router auf den Plan, denn er verbindet als Gateway ein Subnetz mit anderen Netzen. Das Weiterleiten von Daten durch verschiedene Subnetze nennt man Routing.

4.1 Routing-Prinzipien

Hinsichtlich der Routing-Protokolle lassen sich zwei grundsätzliche Verfahren unterscheiden.

4.1.1 Distanz-Vektor-Routing

Jeder Router weiss über die Distanz zu jedem anderen Router bescheid und streut seine Informationen an alle benachbarten Router.

4.1.2 Link-State-Routing

Jeder Router besitzt Informationen über jeden Link der Domain und berechnet dann die gesamte Netztopologie dieser Domain.

4.2 Routing-Protokolle

Um IP-Pakete intelligent weiterleiten zu können benötigen die Router das Wissen, an welche benachbarten Router die unterschiedlichen Datagramme gesendet werden müssen. Dazu hat jeder Router eine Routing-Tabelle, die man z.B. auch manuell konfigurieren kann. Jedoch verändert sich das Netz im Internet ja fortlaufend, und somit müssen die Router irgendwie auf den neusten Stand gebracht werden. Dies geschieht durch Routing-Protokolle, durch welche die Router in zyklischen Zeitabständen Informationen austauschen.

4.2.1 Routing Information Protocol (RIP)

RIP arbeitet nach dem Prinzip des Distanz-Vektor-Routings und stellt ein einfaches, weit verbreitetes Protokoll dar. Jeder Router sendet zyklisch seine Informationen über angeschlossene Links an die benachbarten Router und meldet ihnen die Distanz zu anderen Netzen. So kann ein Router bei einer Veränderung des Netzes seine Routing-Tabellen anpassen und jeweils die kürzesten Wege zu anderen Routern bestimmen. RIP birgt jedoch auch Nachteile. So dauert es eine ganze Weile bis sich Zustandsinformationen über einige Hops weiterverbreitet haben.

4.2.2 Open shortest path first (OSPF)

Im Gegensatz zu RIP ist OSPF ein wichtiger Vertreter der Link-State-Routing-Protokolle. Bei diesem Prinzip tauschen die Router Beschreibungen ihrer direkt angeschlossenen Links aus. Dazu werden in kurzen Abständen (< 30 Sek.) so genannte Hello-Nachrichten über die angeschlossenen Links gesandt. So können die Router die ganze Struktur des Netzwerkes errechnen. Diese Informationen über die Topologie werden dann jeweils weitergegeben, bis sozusagen alle Router über die ganze Domain bescheid wissen.

4.2.3 Border Gateway Protocol (BGP)

Innerhalb von Domains wird üblicherweise mit einem einzigen Routing-Protokoll wie RIP oder OSPF (so genannten Interior Gateway Routing Protocols) gearbeitet. Die verschiedenen Domains mit ihren verschiedenen internen Protokollen sind wiederum über Edge-Router miteinander verbunden. Auf diese Weise kann das auf einer höheren Ebene liegende BGP durchgeführt werden, welches neben der Wegfindung auch viele administrative Funktionen zu erledigen hat.

5 Transportprotokolle

Es soll hier nur kurz der Begriff der Transportprotokolle genannt werden, da ja gerade IP praktisch immer im Zusammenhang mit TCP vorkommt. TCP bedeutet Transfer Control Protocol und wie der Name schon sagt sichert es vor allem die Korrektheit der Übertragung und kann bei Problemen z.B. eine erneute Übertragung anfordern. In der Schichten-Architektur liegt TCP ein Layer höher (Transportschicht) als IP (Internetschicht). TCP ist verbindungsorientiert, d.h. zwischen Client und Server muss eine Verbindung bestehen und Daten werden nicht ohne Verifizierung einfach gesendet. Somit kann es die zwei grössten Schwachstellen von IP gut beheben, und die beiden Protokolle ergänzen sich optimal. Mehr zu TCP und anderen Transportprotokollen ist den entsprechenden Arbeiten zu entnehmen.

6 Schlusswort

IP hat wohl zusammen mit TCP die Welt verändert, denn ohne dieses “Team” könnte das Internet in der heutigen Form wohl kaum existieren. Durch die Tatsache, dass IP einfach und übersichtlich ist und somit einen „kleinsten gemeinsamen Nenner“ darstellt, hat es die globale Vernetzung ermöglicht.

Wie wir sehen hat sich das Internet aber rasant entwickelt und die Ansprüche, vor allem im Bezug auf die Dimensionen und die Sicherheit, sind enorm gestiegen. Diesen Ansprüchen wird die jetzige Version vom IP zum Teil nicht mehr gerecht, es wird Zeit für eine neue Generation. Es ist beachtlich, dass diese Erfindung es so weit gebracht hat, denn sie stammt aus den Anfängen des Internets (US-Militär), und die Idee von IP wird auch in neueren Versionen weiterleben.

7 Referenzen

1. Ausgegebene Dokumente
2. http://www.freesoft.org/CIE/Topics/81.htm
3. http://www.ruhr-uni-bochum.de/~rothamcw/Lokale.Netze/tcpip.html
4. http://www.selfnet.de/

5. http://home.t-online.de/home/TschiTschi/ip_adressierung.htm

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Internet Protokoll Version 6

Simon Solenthaler
simonso@ee.ethz.ch
17. Mai 2002

1 Das Internet-Protokoll Version 6

Das Internet Protokoll (IP) ist das Protokoll, das den Transport von Daten über das Internet ermöglicht. Die klare Abkapselung des IP's gegenüber höher- und tiefergelegenen Layern gestattet ein hardware-unabhängiges Funktionieren. Das Protokoll wird zurzeit noch in der Version 4 genutzt, die neu überarbeitete Version 6 steht kurz vor dem Einsatz. Die Eigenschaften des neusten Internetprotokolls (IPv6) sollen hier durch eine kleine Übersicht hervorgehoben werden.

1.1 IPv6 im Überblick

Im Umgang mit Datenübertragung durch Protokolle spricht man von Datenpaketen (engl. datagrams), die zwischen Sender und Empfänger ausgetauscht werden. Jedes Paket besitzt ein typisches Format , das durch den Header des IPv6 vorgegeben ist:

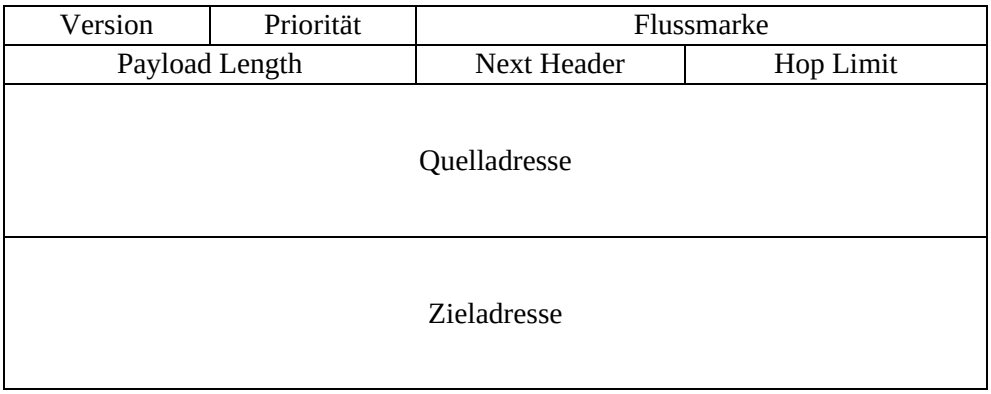

Fig. 1 IPv6 Basic-Header

Jedes Kästchen steht für Informationen, welche zur Verarbeitung und richtigen Verteilung der Datenpakete beitragen. Auf die Bedeutung der einzelnen Felder wird in den folgenden Kapiteln eingegangen.

1.1.1 Unterschiede zwischen IPv4 und IPv6

Der vorrangige Grund überhaupt ein neues Protokoll zu kreieren war sicherlich der begrenzte Adressraum des IPv4. Die Adressgrösse wurde bei IPv6 verzehnfacht. Als weitere Gründe können die mangelhafte Eignung des IPv4 für moderne Internet-Anwendungen und Technologien wie Video on Demand, Web-TV oder E-Commerce aufgeführt werden.

Eine grundlegende Änderung erfuhr der Basic-Header. In der Version 4 enthielt er 13 Felder, da alle Optionen im Header integriert sind. Das IPv6 kommt mit nur 7 Feldern aus und lagert Optionen in Erweiterungs-Header aus (Kap. 1.2.3 Erweiterungs-Header). Die Arbeit für einen Router wird somit bedeutend kleiner, da nicht mehr jede Option analysiert werden muss, bevor sie bearbeitet werden kann. Eine weitere Erleichterung für eine beschleunigte Verarbeitung bietet die festgelegte Header-Länge. Bei IPv4 war die Länge variabel.

Die manuelle Nummerierung von Geräten gehört der Vergangenheit an. Ein Gerät kann an das Internet angeschlossen werden und erhält automatisch eine Nummer zugeteilt. Ob das ein heimischer PC, ein Router oder sonst ein Gerät ist, spielt keine Rolle.

Eine grundlegende Neuerung besteht in der Möglichkeit manipulations- und abhörsichere Übertragung auf jeder Verbindung zwischen zwei IPv6-Rechnern zu gewährleisten. Dazu gehört ein Verschlüsselungsmechanismus und eine Echtheitsüberprüfung von Adressat und Absender. Schliesslich bildet das IPv6 kein vollständig abgeschlossenes Protokoll wie das IPv4. Es lässt sich mit beliebigen Optionen erweitern.

1.2 Das allgemeine Paketformat

1.2.1 Der Basic-Header

Wie bereits angetönt, lässt besitzt der Header verschiedene kleine Sektoren welche die Informationen für die Verkehrsregelung enthalten. Wird ein Datenpaket verschickt, gelangt es zu einem Router, welcher über den weiteren Weg des Pakets entscheiden muss. Die im Protokoll enthaltenen Daten helfen im das Paket richtig weiterzuleiten.

Version

Die ersten vier Bits des Pakets enthalten die Versionsnummer des Protokolls. In diesem Fall ist das Version 6.

Priorität

In den nächsten vier Bits kann die Priorität des Pakets im Vergleich zu andern Paketen bestimmt werden. Besondere Beachtung wird dabei dem Verkehrsfluss geschenkt. Ein System kann diesen mit Prioritätsstufen beeinflussen und so ein überlastetes Netz vor dem Zusammenbruch bewaren.

Flussmarken (Flow Labels)

Die Flussmarken sind eine grosse Neuerung des IPv6. Sie bestehen aus 24 Bit und werden zufällig ausgewählt. Die Idee ist, dass man den Fluss der Pakete von einer Quelle zu einem speziellen Ziel steuern kann und ihnen eine besondere Behandlung durch IPv6-Router zukommen lassen will. Normalerweise muss der Router mehrere Behandlungsschritte durchführen, bis er das Paket weiterleiten kann. Durch eine Flussmarke wird dieser Vorgang beschleunigt, da er sich nur das Label des Paketes und dessen Quell- und Zieladresse merken muss und das nächste Paket mit gleicher Marke sofort weiterleiten kann. Verschiedene solche Flüsse können gleichzeitig stattfinden. Um einer Verwirrung durch gleiche Flussmarken vorzubeugen, merkt sich der Router eine Marke nur während sechs Sekunden. Damit eine Quelle in diesem Zeitraum nicht durch Zufall ein neues Paket mit gleicher Flussmarke an einen neuen Empfänger sendet, wird der Versand auf sechs Sekunden hinausgezögert. Wofür braucht es diesen steuerbaren Fluss? Neben einem Geschwindigkeitsgewinn können durch Flussmarken Ressourcen (Bandbreite, Speicher, etc.) in einem Netzwerk bereitgestellt werden. Als Beispiel sei hier die digitale Videoübertragung erwähnt. Es ist allerdings nicht möglich jeden Fluss zu steuern, da dadurch bei zu kleinem Cache-Speicher des Routers die Suchzeit nach den richtigen Flussmarken zu lange dauern würde. Die Flussmarke kann auch auf null gesetzt werden und ihre Funktion wird somit deaktiviert.

Payload Length

Diese Bitfolge zeigt die Länge des Datenpakets an. Durch die reservierten 16 Bits ist die Paketgrösse auf 65535 Bytes (2^{16}) beschränkt. Falls diese Menge nicht genügt, kann diese Einschränkung durch die Jumbo Payload Option (Kap. Hop-by-Hop-Optionen) umgangen werden.

Next Header

In IPv6 werden optionale Informationen in separaten Erweiterungs-Headern kodiert. Diese Header (es können jedoch auch Protokolle eines höheren Layers sein) werden zwischen dem Basic-Header des IPv6 und den zu übermittelnden Daten platziert. Der Next Header (8 Bit) gibt an, welche Header dem Basic-Header folgen. Als besonders zukunftsträchtig erscheint die Möglichkeit optionale Header zu entwickeln, falls solche verlangt werden.

Hop Limit

Um die Lebensdauer eines Paketes zu beschränken, erhält jedes Paket sein eigenes Hop Limit. Dieser Wert wird bei jedem frequentierten Router um eins dekrementiert. Erreicht das Datenpaket den Wert null bevor es sein Ziel erreicht hat, verfällt es. Man möchte damit verhindern, dass verirrte Pakete endlos in den Netzen zirkulieren. Der zweite Vorteil besteht in der Möglichkeit den Aufwand einer Netzdurchsuchung zu minimieren und die Belastung tief zu halten. Ein Host sendet einen Suchauftrag an eine Gruppe von ähnlichen Servern. Er setzt den Hop Limit Wert auf eins, was bedeutet, dass das Netz bis zur „Tiefe“ eins durchsucht wird. Hat er damit kein Erfolg, versucht er es mit zwei, dann drei, usw. Irgendwann findet er den nächstgelegenen Server, der die gesuchten Informationen bereit hält. Die andern Server werden dadurch nicht belastet.

Quell – und Zieladresse

Die Quell -und die Zieladresse bestehen aus je 128 Bits. Sie sind weltweit eindeutig identifizierbar.

1.2.2 Die Adressierung

Eine wichtige Erweiterung von IPv6 ist die erwähnte Vergrösserung der Adresse auf 128 Bits. Theoretisch sind nun pro Quadratmeter Erdoberfläche ca. $6.65 * 10^{23}$ Adressen möglich. Es wird zwischen drei Adresstypen unterschieden:

- Unicast-Adresse:
  Diese Adresse identifiziert eine einzige Schnittstelle, zum Beispiel ein Router.
- Anycast-Adresse:
  Die Anycast-Adresse kennzeichnet eine Menge von Schnittstellen, die verschiedenen Zwischensystemen angehören. Ein Paket, das an eine solche Adresse gesendet wird, gelangt an das nächst gelegene Interface gemäss der Routing-Metrik.
- Multicast-Adresse:
  IP-Multicast erlaubt, dass ein Sender ein Paket an eine Multicast-Adresse (definiert eine Gruppe) sendet, und das Paket von allen Mitgliedern empfangen wird.

1.2.3 Die Erweiterungs-Header (Extension Header)

Die Erweiterungs-Header stehen für die Vielseitigkeit des IPv6. Für die konventionelle Datenübertragen werden sie nicht gebraucht, der Basic-Header reicht dafür aus. Braucht es für die Übertragung zusätzliche Angaben, greift man zu den Erweiterungs-Headern.
Beispiele:

- Hop-by-Hop-Options-Header
- Routing-Header
- Fragment-Header
- Authentication-Header
- No Next Header
- Destination-Options-Header

Die Erweiterungs-Header sind dem Basic-Header angefügt und enthalten jeweils einen Verweis auf den Typ der nachfolgenden Erweiterungs-Header. Der letzte Header zeigt auf das Protokoll des nächst höheren Layers, in diesem Fall auf den TCP-Header.

IPv6-Header	Routing-Header	Fragment-Header	TCP-Header + Nutzdaten
next header = Routing	next header = Fragment	next header = TCP	(Fragment)

Fig. 2 Erweiterungs-Header

Die Reihenfolge der Header ist zwar wichtig um den Routern, die diese zusätzlichen Informationen benötigen, die Prozessausführung zu vereinfachen, wird aber nicht verlangt. Die folgenden Ausführungen beziehen sich nur auf die wichtigsten Erweiterungs-Header und deren grundlegenden Aufgaben.

Hop-by-Hop-Optionen

Die Hop-by-Hop-Optionen (Hop bedeutet Schaltstelle, bei der Übertragung eines IP-Paketes von einem Router-Eingang zum Router-Ausgang wird genau ein "Hop" passiert) werden für den Austausch optionaler Informationen verwendet. Diese werden in jedem Knoten zwischen Quelle und Ziel ausgewertet. Der Hop-by-Hop-Optionen-Header kann dabei beliebig viele Optionen umfassen. Jede Option wird durch einen Optionstyp und durch eine Längenangabe gekennzeichnet.

Eine wichtige Hop-by-Hop-Option ist die Jumbo-Payload-Option. Sie ermöglicht die durch die Paketlänge auf 65535 Bytes begrenzte Datenmenge zu umgehen. Dabei wird die Payload-Funktion des Basic-Headers auf null gesetzt. Die Grenze der grösstmöglichen Paketgrösse liegt bei ca. 4 GB. Allerdings liegen Übertragungen von Daten dieser Grössenordnung noch jenseits der allgemeinen hardware-technischen Anwendbarkeit.

Routing-Header

Wie der Name schon sagt, besteht durch den Routing-Header die Möglichkeit, den Weg des Datenpaketes zu beeinflussen. So kann verhindert werden, dass das Paket Knoten kreuzt, die nicht erwünscht sind.

Fragment-Header

Was geschieht, wenn ein Paket wegen seiner Grösse nicht durch ein Netzwerk geleitet werden kann? Der Fragment-Header löst dieses Problem. Er erlaubt dem Quellknoten das Paket aufzusplitten und die einzelnen Stücke zu senden. Der Zielknoten setzt das Paket wieder zusammen.

Authentication-Header

Der Authentification-Header (Authentifizierung) überprüft, ob die erhaltenen Pakete authentisch sind. Dieser Überprüfungsmechanismus ist ein Teil der erweiterten Sicherheitsfeatures des IPv6.

Destination Options

Diese Option enthält Angaben über den Zielknoten.

1.3 Wie steht es um die Sicherheit?

Das IP hat eine hohe Popularität erreicht ohne bedeutende Sicherheitsfunktionen für die übermittelnden Daten anzubieten. Solange die Datenübermittlung funktionierte, wurde der Frage der Sicherheit wurde kaum Beachtung geschenkt. Mit IPv6 soll dieser Missstand behoben werden. Die definierten IP-Sicherheitsfunktionen umfassen Erweiterungen zur Unterstützung von Authentifizierungen und vertraulicher Kommunikation. Für IPv4 sind diese Erweiterungen optional, IPv6-Implementierungen müssen zumindest eine Basismenge der Funktionalitäten unterstützen. Damit diese Vorgaben eingehalten werden können, müssen sich Sender und Empfänger auf eine Menge von Parametern einigen, zum Beispiel:

- Schlüssel für Authentifizierung/Verschlüsselung

- Algorithmen zur Verschlüsselung/Authentifizierung
- Lebenszeit der Schlüssel
- Lebenszeit der Security-Assoziation
- Sicherheitsstufe der Kommunikation (vertraulich, unklassifiziert, etc.)

1.3.1 Schlüsselverwaltung

Beinahe alle Authentifizierungs- und Verschlüsselungsalgorithmen basieren auf Schlüsseln. Der Schlüssel ist ein geheimer Bestandteil, welcher nur Sender und Empfänger kennen. Fällt er in falsche Hände, ist die Sicherheit einer Kommunikation mit diesem Schlüssel gefährdet. Da das übermitteln von Schlüsseln stets ein gewisses Risiko beinhaltet, wird in diesem Bereich nach sicheren Lösungen geforscht.

1.3.2 Authentifizierung

Wer Daten sendet, möchte, dass diese in ihrer Ursprungsform ankommen. Zur Überprüfung wird aus dem zu sendenden Paket ein 128-Bit Code ermittelt und in das Authentifizierungsfeld des Headers eingetragen. Dies geschieht durch den MD5-Algorithmus (MD: Message Digest) und einem von beiden Seiten vereinbarten Schlüssel. Der Empfänger errechnet auf die gleiche Weise die Authentifizierungsdaten des erhaltenen Paketes. Sind die beiden Werte gleich, darf er davon ausgehen, dass die Daten ohne Verfälschung übertragen wurden.

1.3.3 Verschlüsselung

Durch den Authentifizierungs-Mechanismus wird es unwahrscheinlich, dass Daten verändert werden. Ein vertraulicher Datenaustausch ist dennoch nicht möglich, da die Informationen auf der Reise durch das Netz von Unberechtigten herausgefiltert werden können. Darum stellt IPv6 ein System zur Verschlüsselung bereit; das Encapsulating Security Payload (ESP). Es wird zwischen zwei Systemen unterschieden. Beim Tunnel-Modus wird das gesamte IP-Paket verschlüsselt und ein neuer unverschlüsselter IP-Header erzeugt, während beim Transport-Modus nur die Nutzlast verschlüsselt wird. Somit kann ein unsicheres Netzwerk sicher überbrückt werden.

1.4 Schlusswort

Durch die kompakte Grundstruktur des Basic-IP-Headers wird eine schnelle und effiziente Übertragung von Daten gewährleistet. Für komplexere Aufgaben bietet das IPv6 verschiedene Erweiterungs-Header, welche erhöhte Sicherheit sicherstellen, Fragmentierung erlauben, den Weg durch das Netzwerk festlegen und weitere Funktionen ermöglichen. Zudem ist das Protokoll noch nicht vervollständigt worden. Neue Erweiterungs-Header können implementiert werden. Für die Zukunft lässt das viel Spielraum offen. Der Adressierraum ist gross genug um über längere Zeit genügend Anschlussmöglichkeiten zu bieten. Das IPv6 bietet ein solides Fundament an, das seine Existenz rechtfertigt.

Referenzen

1. http://page.to/ipv6
2. http://www.ipv6-net.de/faq.html
3. Autor unbekannt, Chapter 4, Internet Protocol Version 6
4. Autor unbekannt, Kapitel 3, Das Internet-Protokoll der nächsten Generation
5. http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Mobile IP

Benjamin Marti
bmarti@ee.ethz.ch
24. Mai 2002

1 Einführung

Obwohl das Internet Zugang zu Informationsquellen weltweit bietet, kann normalerweise nur von bestimmten Zugangsorten davon profitiert werden: im Büro, in der Schule oder zuhause. Die rasant wachsende Auswahl an mobilen IP-fähigen Geräten wie PDAs und Mobiltelefonen verändert die Möglichkeiten des Internets.

Mobile Computing und Networking sollte nicht mit dem Portable Computing und Networking verwechselt werden, das heute schon weit verbreitet ist. Beim Mobile Networking werden keine Computer-Prozesse unterbrochen, wenn der Ort des Internetzugangs gewechselt wird. Das Aufbauen der Verbindung geschieht völlig automatisch.

Wirkliches Mobile Computing bietet viele Vorteile. Zuverlässiger Internetzugang überall und zu jeder Zeit stellt den statischen Arbeitsplatz im Büro in ein völlig anderes Licht. Man bedenke nur was die Mobiltelefonie in den letzten Jahren ausgelöst hat. Die Möglichkeit, ein vollständiges Computersystem ständig zur Verfügung zu haben fördert nicht nur die Flexibilität sondern hat auch das Potential, die gesamte Arbeitsethik grundlegend zu verändern.

1.1 Motivation

Im weltweiten Internet werden Daten in Form von IP-Paketen von Endsystem zu Endsystem transportiert. Die Vermittlung geschieht mit Hilfe von Routern basierend auf der IP-Quell- und Zieladresse im Paketkopf. Aufgrund des am besten passenden Netzwerkpräfixes legt der Router fest, wie ein Paket weitergeleitet wird. Der letzte Router vor dem Zielsystem legt schlussendlich das physikalische Subnetz des empfangenden Rechners fest. So liegt zum Beispiel ein Rechner mit der IP-Adresse 129.12.31.145 im physikalischen Subnetz 129.12.31 nach klassischer Adressierung.

Probleme stellen sich, wenn ein Endsystem von einem physikalischen Subnetz in ein anderes gebracht wird. Das kann durch eine geänderte Funkanbindung in einem drahtlosen LAN oder auch ganz simpel durch Umstecken einer herkömmlichen Verbindung geschehen. Nach dem Wechsel in ein anderes Subnetz kann ein solches System ohne Zusatzmassnahmen keine Datenpakete mehr empfangen. Das liegt daran, dass es sich mit seiner IP-Adresse im „falschen“ Subnetz, das heisst in einem Subnetz mit einem anderen Präfix befindet.

Ohne ein zusätzliches Protokoll gäbe es zwei Lösungen für dieses Problem.

1.1.1 Wechseln der IP-Adresse

Je nach dem aktuellen Aufenthaltsort eines Endgeräts könnte seine IP-Adresse immer wieder so geändert werden, dass sie zu dem jeweiligen Präfix des Subnetzes passt. Dann müssten allerdings auch sämtliche DNS-Einträge für das Endsystem angepasst werden, da erst diese den Rückschluss von einer logischen Adresse auf die IP-Adresse ermöglichen. Diese Änderungen der DNS-Einträge dauern aber zu lange. Ausserdem sind aus Stabilitäts- und Sicherheitsgründen nicht alle Endsysteme dazu berechtigt. Auch ein ganz praktischer Grund spricht gegen die Änderung der IP-Adresse: viele Endsysteme müssen nach einer Änderung ihrer IP-Adresse neu gestartet werden. Das verunmöglicht einen nahtlosen Übergang zwischen Subnetzen.

1.1.2 Spezifische Wege zum Endsystem

Zu jedem mobilen Endsystem könnte in Routern ein spezieller Eintrag vorhanden sein (z.B. 129.12.31.145). Dies wäre theoretisch möglich, da in einem Router stets versucht wird, den am besten passenden Präfix zu finden, in diesem Fall sogar die ganze Adresse. Dieser Ansatz muss allerdings verworfen werden, da für jedes Gerät ein spezieller Eintrag in einem Router vorhanden sein müsste, diese aber schon jetzt aufgrund der grossen Anzahl von Einträgen an die Grenzen ihrer Kapazität stossen.

Aus diesen Gründen musste eine Erweiterung des bestehenden IP-Protokolls entworfen werden.

1.2 Anforderungen an Mobile IP

Folgende Anforderungen wurden beim Entwurf von Mobile IP gestellt:

- Transparenz: Die Mobilität soll für die höheren Schichten und die Anwendungsprogramme nicht sichtbar sein. Bei einem Wechsel des Aufenthaltsortes eines mobilen Endgerätes sollen die höheren Schichten ohne Unterbruch weiterarbeiten. Die IP-Adresse des Endgerätes soll immer dieselbe sein.
- Kompatibilität: Mobile IP muss mit dem bestehenden IP vollständig kompatibel sein. Mobile Endgeräte sollen mit festen und umgekehrt kommunizieren können. Es dürfen keine Änderungen an bisherigen Endsystemen und Routern notwendig werden.
- Sicherheit: Mit der Mobilität kommt immer die Frage nach der Sicherheit: Fremde, nicht autorisierte Geräte könnten sich in ein Netzwerk einbinden. Deshalb müssen alle Registrierungsnachrichten, d.h. die Nachrichten zur Integration des mobilen Endsystems ins Netz, authentifiziert werden.
- Effizienz und Skalierbarkeit: Mobile Endgeräte sind häufig nur über kabellose schmalbandige Anbindungen zu erreichen. Deshalb sollten möglichst wenig zusätzliche Daten ausgetauscht werden müssen. Ausserdem sollten internetweit möglichst viele mobile Endsysteme unterstützt werden.

1.3 Definitionen

- Mobile Node: Als Mobile Node wird das Endsystem bezeichnet, das den Punkt seines Netzanschlusses wechseln kann, ohne seine IP-Adresse zu wechseln.
- Home Agent: Der Home Agent ist eine Einheit (typischerweise auf einem Router) im „Heimatnetz“ des Mobile Node, die den Aufenthaltsort des Mobile Node verwaltet und die Datenpakete zur Care of Address weiterleitet („tunnelt“). Als Heimatnetz wird das Subnetz verstanden, zu dem der Mobile Node laut seiner IP-Adresse gehört.
- Foreign Agent: Der Foreign Agent ist eine Einheit (typischerweise ebenfalls auf einem Router), welche die vom Home Agent empfangenen Pakete an den Mobile Node weiterleitet. Meist ist der Foreign Agent auch der Standard-Router für den Mobile Node und er stellt ausserdem die Care of Address für den Mobile Node zur Verfügung.
- Care of Address: Die Care of Address stellt die Adresse des für den Mobile Node aktuell gültigen Tunnelendpunkt und für andere Endsysteme im Internet den aktuellen Aufenthaltsort des Mobile Node dar. Die Care of Address kann sich auf dem Foreign Agent oder dem Mobile Node selber (co-located) befinden.

2 Funktionsprinzip

In Bild 1 wird der Ablauf des Datenverkehrs zwischen einem fixen Endsystem und einem Mobile Node aufgezeigt. Das sendende Endsystem muss den aktuellen Aufenthaltsort des empfangenden Mobile Node nicht kennen. Es sendet in Schritt 1 das Datenpaket ganz normal an die ihm bekannte IP-Adresse des Mobile Node. Die Router im Internet auf dem Weg zum Heimatnetz leiten das Datenpaket ganz normal weiter bis sie zum Home Agent kommen. Falls der Mobile Node in diesem Moment im Heimatnetz ist, leitet er sie diesem weiter. Wenn der Mobile Node nicht im Heimatnetz ist, wird das Datenpaket gekapselt und in einem Tunnel an die Care of Address des Mobile Node weitergeleitet (Schritt 2). Im gezeigten Beispiel liegen der Tunnelendpunkt und der Foreign Agent im Router für das aktuelle Fremdnetz des Mobile Node. Der Foreign Agent entkapselt das Datenpaket wieder und leitet es in Schritt 3 an den Mobile Node weiter.

Der Mobile Node kann Daten direkt über den Foreign Agent als Standardrouter und über die Router im Internet an irgendein Zielendsystem senden.

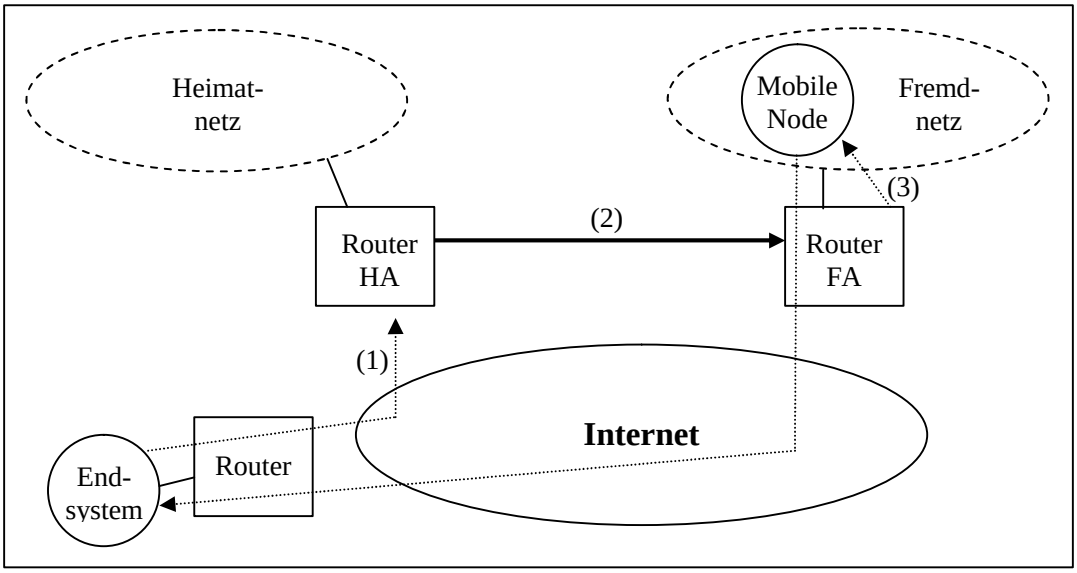

Fig. 1 Ablauf des Datenverkehrs zwischen irgendeinem Endsystem und einem Mobile Node.

Für den Datenverkehr durch den Tunnel wird die sogenannte IP-in-IP-Kapselung angewendet. Dabei wird über den ursprünglichen IP-Kopf einfach noch ein zweiter gelegt. Die Originaldaten (Sender- und Empfängeradresse) bleiben dabei erhalten.

2.1 Netzintegration

Bei Mobile IP stellen sich auch Fragen zur Sicherheit. Damit sich keine unbefugten Endsysteme in ein Netz integrieren können, müssen bestimmte Aktionen zur Integration authentisiert werden.

2.1.1 Agent Advertisement

Home Agent und Foreign Agent senden periodisch Nachrichten in ihre jeweiligen Subnetze. Ein Mobile Node hört diese Nachrichten und kann entscheiden, ob er sich in seinem Heimatnetz oder in einem Fremdnetz befindet. Falls er sich in seinem Heimatnetz befindet sind keine weiteren Schritte mehr nötig.

Befindet er sich in einem Fremdnetz, kann er aus diesen Nachrichten die Care of Address ablesen.

2.1.2 Registrierung

Als nächsten Schritt meldet sich der Mobile Node beim Foreign Agent und via diesen beim Home Agent an. Diese Aktionen müssen über Authentisierungsverfahren abgesichert werden.

2.1.3 Bekanntmachung

In einer letzten Phase kann der Home Agent andere Router darüber informieren, dass der Mobile Node über diesen zu erreichen ist. Dies ist zum Beispiel bei der Integration eines bisher noch unbekannten Rechners sinnvoll. Pakete an den Mobile Node können ab jetzt an den Home Agent geschickt werden. Allfällige Änderungen des Foreign Agents oder der Care of Address haben darauf keinen Einfluss, da der Home Agent immer weiss, wohin er die Pakete weiterleiten muss.

3 Optimierungsmöglichkeiten

3.1 Effizienz

Das Senden von Paketen via Home Agent zum Mobile Node, das auch als Triangular Routing bezeichnet wird, stellt einen Umweg dar. Es besteht die Möglichkeit, dass ein Sender den aktuellen Aufenthaltsort des Mobile Node lernt nachdem ihm dieser vom Home Agent mitgeteilt wurde. Er kann die zu sendenden Pakete anschliessend direkt zur aktuellen Care of Address tunneln.

Eine weitere Optimierung betrifft Pakete, die noch zur alten Care of Address unterwegs sind, während dem der Mobile Node schon eine neue Care of Address bekommen hat. Damit keine Daten verloren gehen kann der neue Foreign Agent den Alten benachrichtigen damit dieser die Pakete weiterleitet.

3.2 IPv6

Mobile IP wurde zwar ursprünglich für IP Version 4 entwickelt, die neue Version 6 erleichtert aber einiges. Zahlreiche Sicherheitsmechanismen sind in IPv6 standardmässig integriert.

Bei IPv6 wird auch kein Foreign Agent mehr benötigt. Alle Router beherrschen das Router Advertisement, das anstelle des Agent Advertisement eingesetzt werden kann. Ein Mobile Node kann einen Sender direkt über seine Care of Address informieren. Bei einem Wechsel zwischen Subnetzen kann der Mobile Node dem bisherigen Router ausserdem direkt seine neue Care of Address mitteilen damit dieser allfällige noch an ihn gesendete Datenpakete weiterleiten kann.

3.3 Reverse Tunneling

Die Möglichkeit des Mobile Nodes, direkt Daten über den Foreign Agent an irgendeinen Empfänger senden zu können, birgt einen gewaltigen Nachteil: Diese Pakete haben eine topologisch falsche Absenderadresse, d.h. die Pakete werden aus einem physikalischen Subnetz gesendet, dessen Präfix nicht zur Sendeadresse passt. Aus Sicherheitsgründen besitzen die meisten Router Firewall-Funktionalität, die genau solche topologisch falschen Pakete herausfiltert und verwirft. Das bedeutet, dass Mobile IP in seiner ursprünglichen Form in realen Netzen mit Firewalls nicht funktionieren kann.

Nachdem dies erkannt wurde, wurde das Reverse Tunneling entwickelt: Auch für den Weg der Daten vom Mobile Node zu einem Empfänger wird zwischen Foreign Agent und Home Agent ein Tunneln mit Kapselung der Pakete eingesetzt.

4 Probleme, Ausblick

Es gibt verschiedene Probleme im Zusammenhang mit Mobile IP, die noch nicht befriedigend gelöst werden konnten. Die Sicherheit ist bei Mobilität naturgemäss ein grosser Schwachpunkt. Die beschriebene Authentifizierung kann in der Realität schwierig sein, da der Foreign Agent typischerweise einer fremden Organisation untersteht. Bei der Verteilung und Verwaltung von Schlüsseln gibt es noch keine Standardisierung.

Beim Umgehen des Triangular Routing sind unter Umständen viele zusätzliche Authentisierungen nötig, da nicht nur zwischen Foreign Agent und Home, sondern zwischen dem Foreign Agent und jedem Sender Agent eine Sicherheitsverbindung besteht.

Auch das Reverse Tunneling löst nicht alle Probleme, da ein nur schwer zu kontrollierender Tunnel aus dem physikalische Subnetz ins unsichere Internet besteht, was Firmen häufig nicht zulassen.

Sobald aber all diese Probleme einmal gelöst sein werden und für die Internetnutzung nicht mehr ein fester Netzwerkanschluss benötigt wird, wird Mobile IP das Arbeits- und auch das Freizeitverhalten unserer Gesellschaft nachhaltig verändern.

Referenzen

1. C. Perkins: RFC 2002; IBM, 1996
2. http://www.computer.org/internet/v2n1/perkins.htm
3. http://www.telematik.informatik.uni-karlsruhe.de/lehre/alt/vorlesungen/Tele1-Folien_WS9697/K12-Mobi/sld014.htm
4. weitere abgegebene Unterlagen

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

TCP und UDP

Miskovic Sascha
sasam@ee.ethz.ch
17. Mai 2002

1 Internet-Transportprotokolle

Die Transportschicht des Internets besitzt zwei Protokollarten, eine verbindungslose und eine verbindungsorientierte. Die verbindungslose beinhaltet das User Datagram Protocol (kurz UDP) und die verbindungsorientierte das Transmission Control Protocol (TCP). Schauen wir nun die grundlegende Funktionsweise beider Protokolle an.

1.1 Transmission Control Protocol

TCP wurde entwickelt um einen Bytestrom zuverlässig in einem unzuverlässigen Netzverbund zu übertragen, wobei wir in einem Netzverbund unterschiedliche Topologien auffinden. Beim TCP werden Unterschiede in Verbundnetzen dynamisch angepasst, und somit wird das gesamte Gebilde robuster. Die Dateneinheit des Transmission Protocol heisst **TCP-Segment.**

1.1.1 Eigenschaften des TCP

Die typische TCP-Zuverlässigkeit basiert auf folgenden Eigenschaften:

- Eine virtuelle, vollduplexe Punkt-zu-Punkt Verbindung wird aufgebaut, vollduplex bedeutet, dass die Daten gleichzeitig unabhängig voneinander in beide Richtungen gesendet werden können und Punkt-zu-Punkt, dass jede Verbindung zwei Endpunkte hat. Nach erfolgreichem Datenaustausch wird die Verbindung abgebaut.
- Der Bytestrom wird beim Übertragen weder verändert noch interpretiert.
- Die Grösse des Puffers, auf dem zwischengespeichert wird bestimmt durch die Grösse der zu sendenden Segmente. Für einen reibungslosen Ablauf ist die Flusskontrolle verantwortlich, welche hierfür ein Window-Feld benützt, auf das später genauer eingegangen wird.
- Jedem zu sendenden Segment wird eine Sequenznummer zugewiesen. Mit einem Acknowledgement bestätigt der Empfänger jedes übertragene Segment, wird eine Sequenznummer nicht bestätigt so wird das entsprechende Paket noch mal gesendet.

1.1.2 Der TCP-Segment-Header

Jedes Segment wird initialisiert durch ein 20-Byte-Header, welches ein festes Format aufweist. Dem Heder folgen normalerweise Optionen und denen wiederum können bis zu 65.495 Datenbyte folgen. Hier sieht man den Aufbau des 20-Byte-langen TCP-Headers:

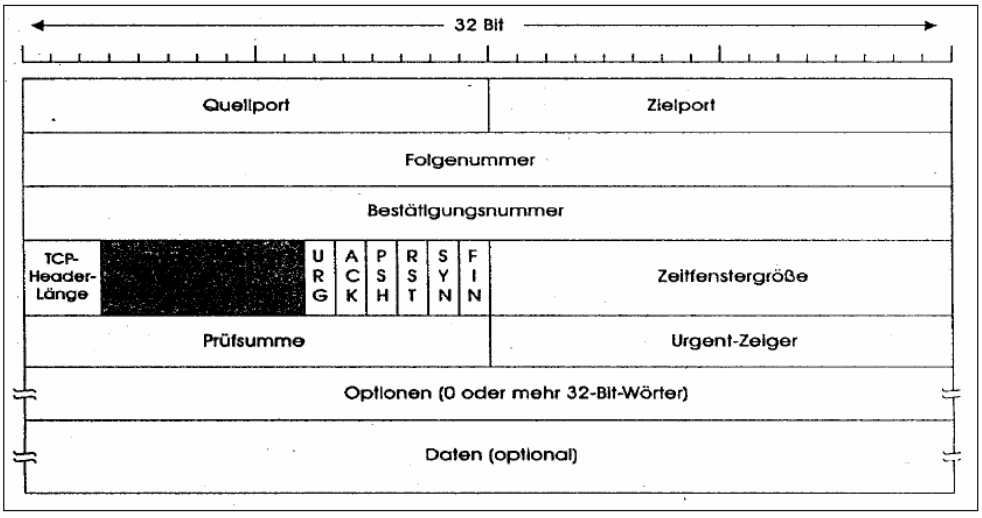

Fig. 1 Der TCP-Header

- Quellport (Source Port) und Zielport (Destination Port): Wie der Name bereits andeutet, sind diese zwei Ports für die Identifizierung der lokalen Endpunkte der Verbindung verantwortlich. Die Ports ab 256 können vom Host selbst zugewiesen werden, die Ports unter 256 sind für spezielle Anwendungen definiert, z.B. Telnet usw.
- Folgenummer (Sequence Number) und Bestätigungsnummer (Acknowledgement Number): Der Sender gibt beim Folgenummer-Feld die Nummer des nächst zu sendenden Segments! Der Empfänger hingegen gibt bei Acknowledgement Number gibt die Bestätigung für erfolgreiches Empfangen des Segments.
- TCP-Header-Länge (TCP Header Length): Da das Options-Feld eine variable Länge hat, muss die Länge des Headers angegeben werden und dieses Feld macht das in 32-Bit-Worten.
- Reservierte Bits: Die nächsten sechs Bits sind für zukünftige Optimierung frei gehalten. Sie sind alle mit 0 initialisiert.
- Sechs 1-Bit-Flags:
    - URG: Hier wird der Urgent Pointer aktiviert um Daten mit hoher Priorität zu markieren.
    - ACK: Dieses Bit ist für die Gültigkeit der Bestätigungsnummer erforderlich. Ist es mit 0 initialisiert, so bedeutet dies: fehlende Betätigung für das Segment. In diesem Fall wird das Feld Acknowledgement Number nicht beachtet.
    - PSH: Hier werden PUSH-Daten definiert. Dem Empfänger soll angezeigt werden, dass diese Daten nicht zwischengespeichert werden sollen. Sie sollen nach der Ankunft sofort bereitgestellt werden.
    - RST: Wenn der Host abstürzt oder ein anderes Problem beim Transfer auftritt, setzt dieses Bit die Verbindung zurück. Es wird auch benützt um Verbindungen bei ungültigen Aufbauversuchen abzuweisen.
    - SYN: Mit dem SYN-Bit wird eine Verbindung aufgebaut. In Kombination mit dem ACK-Bit, kann man unterscheiden zwischen dem Status CONNECTION REQUEST und CONNECTION ACCEPTED.
- Zeitfenstergrösse (Window): Dies ist ein Schiebefenster, welches für die Flussteuerung verantwortlich ist. Der Wert dieses Feldes bestimmt die Datenmenge in Bytes, welche der Sender des TCP-Segments noch empfangen kann. Der Wert 0 besagt, dass alle Bytes bis zur Acknowledgement Number empfangen wurden, und dass der Empfänger nicht mehr in der Lage ist weitere Daten zu empfangen.
- Prüfsumme (Checksum): Dies ist ein weiterer Faktor, welcher extreme Zuverlässigkeit garantieren soll. Es wird eine Prüfsumme vom Header und vom Datenteil des Segments berechnet.
- Urgent-Zeiger: Mit diesem Zeiger werden Daten, welche direkt am Header angehängt sind, mit hoher Priorität versehen.
- Optionen: Hier können im Header zusätzliche Funktionen verankert werden. Die Wichtigste ist die der Segmentgrössenoptimierung. Es ist effizient in möglichst grosse Segmente zu fragmentieren, da so weniger Header mitgeliefert und verarbeitet werden müssen. Oftmals ist die obere Grössengrenze bei Sender und Empfänger nicht äquivalent, in diesem Fall einigt man sich logischerweise auf die kleinere maximale Grösse. Die kleinste Segmentgrösse ist vordefiniert und beträgt 536-Byte (Nutzdaten) und 20 Byte (Header), also 556 Byte.

1.1.3 TCP-Verbindungsverlauf

Das Dreiwege-Handshake-Prinzip bestimmt den Verbindungsaufbau. Nach der Ausführung der Operationen LISTEN und ACCEPT wartet der Server passiv auf eine ankommende Verbindung. Der Client hingegen führt die Operation CONNECT aus und gibt die IP-Adresse und den Port an, zu der er eine Verbindung anstrebt, ausserdem gibt er noch seine maximal zu akzeptierende TCP-Segmentgrösse und wahlweise einige Benutzerdaten (Passwort, usw.) an. Durch die Ausführung der CONNECT-Operation, wird ein TCP-Segment mit gesetztem SYN- und ausgeschaltetem ACK-Bit gesendet und anschliessend wird auf die Antwort gewartet. Nach dem Ankommen des Segments prüft die TCP-Einheit, ob der Server sich im LISTEN-Zustand befindet, falls dies nicht zutrifft wird die Verbindung abgewiesen, indem mit einem gesetzten RST-Bit geantwortet wird. Hat der Server aber die LISTEN-Operation ausgeführt, so kann er die Verbindung annehmen, indem er ein Bestätigungs-segment zurückschickt. Im Normalfall sieht die TCP-Segment-Folge, wie in Fig. 2 (a) dargestellt, aus. Versuchen aber zwei Hosts gleichzeitig eine Verbindung zu den gleichen Sockets aufzubauen, so tritt der Fall (b) auf.

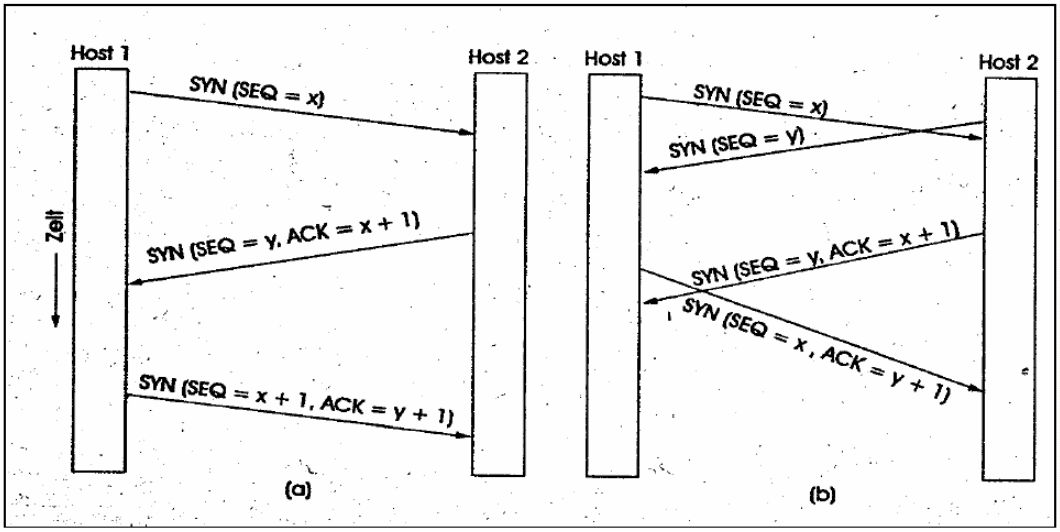

Fig. 2 (a) TCP-Verbindungsaufbau im Normalfall; (b) Kollision

Die vollduplexe Verbindung, kann man sich als ein Paar von Simplexverbindungen vorstellen. Die Simplexverbindungen werden unabhängig voneinander abgebaut und zwar folgendermassen: Die Partei, welche keine Daten mehr zu senden hat, schickt ein TCP-Segment mit gesetztem FIN-Bit, bestätigt die andere Partei das FIN-Bit, wird die eine Richtung gesperrt. Das gleiche macht man für die andere Richtung.

Die Zustände, welche beim Verbindungsaufbau, Verbindungsverlauf und Verbindungs-abbau auftreten, können wie folgt zusammengefasst werden:

Zustand	Beschreibung
CLOSED	Keine Verbindung aktiv oder anstehend
LISTEN	Der Server wartet auf eine ankommende Verbindung
SYN RCVD	Ankunft einer Verbindungsanfrage und Warten auf Bestätigung
SYN SENT	Die Anwendung hat begonnen, eine Verbindung zu öffnen
ESTABLISHED	Zustand der normalen Datenübertragung
FIN WAIT 1	Die Anwendung möchte die Übertragung beenden
FIN WAIT2	Die andere Seite ist einverstanden, die Verbindung abzubauen
TIMED WAIT	Warten, bis keine Pakete mehr kommen
CLOSING	Beide Seiten haben versucht, gleichzeitig zu beenden
CLOSE WAIT	Die Gegenseite hat den Abbau eingeleitet
LAST ACK	Warten, bis keine Pakete mehr kommen

Tab. 1 Zustände im TCP-Verbindungsmanagement

Der Verlauf des Aufbaus, der Übertragung und des Abbaus ist im untenstehenden Modell abgebildet:

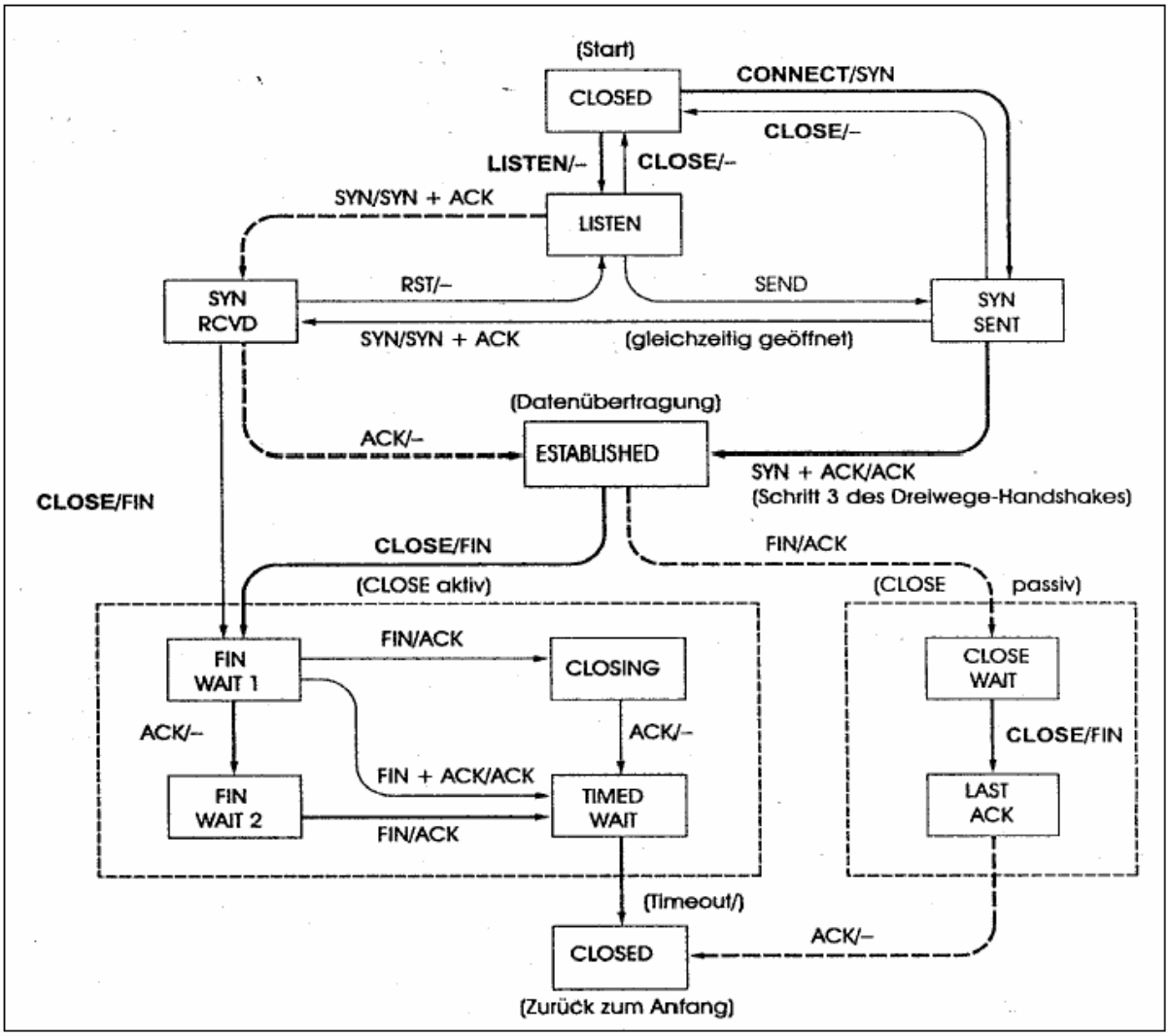

Fig. 3 TCP-Verbindungsmanagement (fette Linie = normaler Pfad des Clients; fette gestrichelte Linie = normaler Pfad des Servers; feine Linien = ungewöhnliche Ereignisse)

1.2 User Datagram Protocol

UDP ist ein verbindungsloses einfaches Transportschichtprotokoll, das heisst: Pakete werden unabhängig voneinander versendet. Dieses Protokoll übernimmt keine Garantie, ob alle Daten ankommen und ob sie in der richtigen Reihenfolge ankommen. Die Dateneinheit des User Datagram Protocol heisst **Nutzer-Datagram.** UDP dient mit wenig Protokollaufwand dem Datenaustausch zwischen Programmen. Im Gegensatz zu TCP unterstützt es Multicating und Broadcasting. Typische Anwendungen sind Konferenzanwendungen mit Übertragung von Audio- und Videodaten. Beim live-Videostreaming wäre TCP äusserst unpraktisch, da wenn ein Fehler auftreten würde, müsste das TCP das fehlgesendete Paket erneut senden und der Bildfluss wäre deutlich gestört, beim UDP würde es nur ein paar verzerrte oder fehlende Bilder geben, der Bildfluss wäre mehr oder weniger nicht gross gestört. Der UDP-Header ist nur 8-Byte lang, eine Folge dessen ist die Unzuverlässigkeit, welche man wie am oberen Beispiel wohlgemerkt nicht immer braucht. Schauen wir uns jetzt den UDP-Header an:

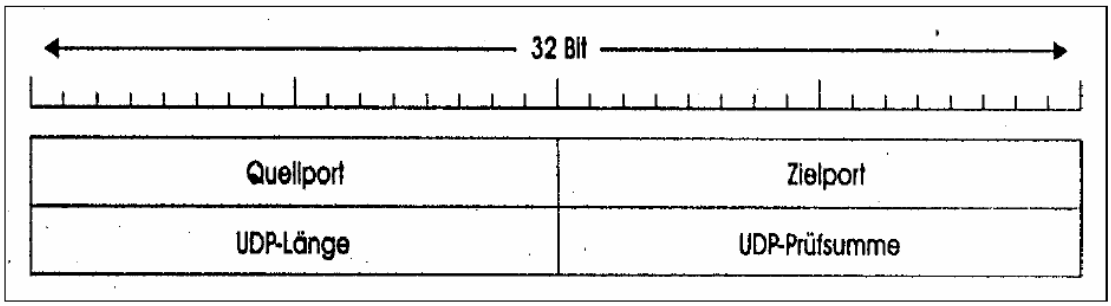

Fig. 4 Der UDP-Header

- Quellport (Source Port): und Zielport (Destination Port): Sie dienen dem gleichen Zweck wie bei TCP, indem sie die Endpunkte der Quell und Zielmaschine identifizieren.
- UDP-Länge (Length): Dieses Feld gibt die Länge des Nutzer-Datagrams inklusive des 8-Byte langen Headers.
- UDP-Prüfsumme (Checksum): Hier wird eine Prüfsumme über das User-Datagram-Header und die Nutzdaten berechnet.

1.3 Schlusswort

Beides sind Protokolle der Transportschicht, welche den Dienst des IP benutzen, jedoch aber über weitere Funktionen verfügen. TCP bietet sehr hohe Zuverlässigkeit und ist verbindungsorientiert, UDP hingegen ist flexibel, schnell und verbindungslos. Beide haben ihre spezifischen Anwendungsbereiche.

Referenzen:

1. “Abgegebenes Literaturmaterial“

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Das Hypertext Transfer Protokoll (HTTP)

Simon G. Wrann
swrann@ee.ethz.ch
24. Mai 2002

1 Einführung

Beim World Wide Web handelt es sich um ein mediales System, bei dem die Informationen in Form von Web-Seiten gespeichert sind, die über Web-Links miteinander verbunden sind.Die Links sind auch bekannt unter dem Namen URL oder URI, oder noch einfacher als Name der Web-Seite.Diese Eigenschaft des Web erfordert ein verbindliches Verfahren zwischen Kommunikationspartern, damit diese sich bei ihren gegenseitigen Anfragen verstehen können.Dieses Verfahren für den Zugriff auf die nicht lokal gespeichterte Information ist das Hypertext Transfer Protokoll (HTTP).Das Protokoll übernimmt die Mittlerfunktion zwischen zwei sich fremden Systemen, legt fest, wie die Interaktion gestartet und beendet wird, welche Informationen ausgetauscht werden dürfen und wie das System auf Anfragen zu reagieren hat.Das HTTP ist somit eine Schlüsselkomponente des WWW. Mit dem HTTP kommt man als User fast nie direkt in Berührung, nur zum Beispiel bei Fehlermeldungen, aber jeder Klick auf ein Symbol des Browsers hat unmittelbaren Einfluss auf die ausgegebene Nachrichten des HTTP.

1.1 Geschichte

Beim ursprünglichen Entwurf von HTTP, handelte es sich um ein Programm mit den Zielen:

- Einfachheit
  Leicht implementierbar, wenig Ressourcen beanspruchen
- Schnelligkeit
  weil die Datenmenge gross und Verteilung derselben gross ist

1.1.1 HTTP/0.9

Diese erste Version von HTTP unterstützte ausschliesslich eine Methode, nämlich GET. Ein Client musste die Verbindung mit dem Server Aufbauen, eine Zeile mit dem Schlüsselwort GET und dem Namen des Dokuments an diesen senden.Der Server antwortete mit dem Dokument selbst und brach die Verbindung ab um das Ende des Dokuments zu signalisieren. Nachteile:

- nur Text-Uebertragung möglich
- Client kann keine Daten an Server senden

1.1.2 heutiges HTTP/1.1

Eine Verbesserung ist im Bereich der Request/Response-Interaktion, genannt Persistant HTTP (P-HTTP).Eine für eine Interaktion aufgebaute Verbindung bleibt bestehen, bis kein Request mehr eintrifft.Das Resultat ist ein weniger häufiges aufbauen von TCP-Verbindungen.Weitere Verbesserungen sind z.B. Neue Request-Methoden; teilweise Uebertragbarkeit von einer Entity; Content Negotiation; Authentifizierungsschema und anderes mehr.

2. Was ist HTTP

HTTP ist ein Request/Response-Protokoll, das auf einem verbindungsorientierten Transferdienst aufbaut.Als solches verwendet es die Rollen Client, der den Request sendet und Server, der den Response sendet.

Neben diesen beiden grundlegenden Rollen können auch weitere Zwischenstationen in dieser Request/Response-Kette vorhanden sein, nämlich: Proxies, Gateways und Tunnels.Diese Rollen sind nicht statisch, will heissen, dass jedes Programm diese Rollen ändern und mehr als eine dieser Rollen ausüben kann.Es gibt also grundsätzlich zwei Arten der Verbindung:

1. Direkte Verbindung zwischen Client und Origin Server
2. Verbindung über einen Mittler (Zwischenstation)

Für die Mittler-Verbindung gibt es, wie schon erwähnt, drei verschiedene Arten:

1. Mittels einem Proxy.
   Ein Proxy ist ein sowohl als Client als auch als Server fungierendes Vermittlungsprogramm, das Requests entgegennimmt und dann als Client für andere Clients die Requests an den Origin-Server weitersendet.Ein Request an ein Proxy kann aber auch intern bedient werden, mittels einem Cache.(In einem Cache kann ein Programm Responses verschiedener Origin-Server speichern und bei erneuter Anfrage für den Response auf den Cache zurückgreifen, was den Transferprozess um eine Instanz verkürzt.)
2. Mittels einem Gateway
   Ein Gateway ist dem Proxy ähnlich, mit dem Unterschied, dass das Gateway diskret ist.
3. Mittels einem Tunnel
   Ein Tunnel ist ein blinde Zwischenstation.Die eingegangene Nachricht wird nicht interpretiert, sondern lediglich befördert.

2.1. Aufbau des Protokolls

Der Aufbau ist immer gleich, bedeutet dass somit für alle verbindlich.Jede HTTP-Nachricht hat folgendes Format:

generic-message =
start-line
*message-header
CRLF
(message-body)

start-line =
request-line / status-line

2.2. Der Header

Es gibt vier Headertypen:

- *General Header*: Beinhaltet generelle Informationen wie Datum, Uhrzeit und wie mit der Nachricht zu verfahren ist.Ausserdem hinterlässt jede Zwischenstation eine Spur im Header, sodass der Weg, der Nachricht nachvollzogen werden kann.
- *Entity Header*: Beinhaltet Informationen über den Inhalt wie Grösse, Codierung.Somit verrät dieser auch den Decodierungsmechanismus.Der Header gibt auch die verwendete Publikumssprache an, desweiteren die Anweisung wie mit den Einzelteilen auch das Ganze wieder zu einem Stück geformt wird und ausserdem wird der Medientyp der Information angegeben.

- *Request Header*: Beinhaltet Informationen über den Request und den Client.
- *Response Header*: Zusätzliche Information für den Client, die nicht in der Status-line angegeben werden.

2.2.1. Der Request-Header

Der Request-Header ist immer die erste Nachricht einer HTTP-Interaktion.Das Format bleibt dasselbe des allgemeinen Headers:

request =
request-line
*
*
*
request-line =
method request-URI HTTP-Version

- *method* gibt an welche Methode der Server in der Request-URI anwenden soll.
- *request-URI* gibt die Ressource an, auf die der Response anzuwenden ist.
- *HTTP-Version* gibt die verwendete HTTP-Version an.

Der Request-Header wird vom vom Client zum Uebertragen zusätzlicher Informationen über den Request und dem Client selbst an den Server verwendet.Folglich dient der Request-Header dem Abändern zur genaueren Ausführung des Requests.Eine wichtige Funktion des Headers ist die Möglichkeit der Authorization, dieses Feld wird vom Client verwendet, um sich selbst bei einem Server zu authentifizieren.
Das einzige Header-Feld, das in jedem Request enthalten sein muss, ist das Feld Host.In diesem wird der Internet-Host sowie die Port-Adresse der angeforderten Ressource angegeben.

2.2.2. Der Response-Header

Eine Response-Message ist immer die zweite Nachricht, in einer HTTP-Interaktion.Sie wird vom Server an den Client geschickt und enthält das Ergebnis der Verarbeitung des Requests durch den Server.Das Format ist erneut sehr einfach:

Response =
status-line
*
*
*
status-line =
HTTP-version status-code reason-phrase

Die wichtigen Unterschiede zum Request-Header sind:

- statuscode: Der statuscode ist der Teil, mit dem man als User direkt in Berührung kommt.Er enthält die Fehlermeldung bei nicht gelingen der Verarbeitung, oder auch die Success-Meldung bei gutem Ablaufen der Interaktion.
- reason-phrase: Dieses Feld soll dem Client in Kurzform die Bedeutung des Statuscode darlegen.

Auch im Response muss eine WWW-Authenticate enthalten sein.Vom Client wird erwartet, dass er die im Header-Feld enthaltenen Informationen zum Erstellen eines Requests verwendet, der in seinem Header-Feld Authorization die Authentifizierungsinformation enthält.

3. Der HTTP-Server

HTTP-Server sind die wichtigsten Programme für die Web-Infrastruktur.Per Definition ist ein Server ein Programm, das auf Requests von Clients wartet diese bearbeitet und das Ergebnis als Response sendet.
In praktisch allen Fällen wird ein Web-Server für einen permanenten Betrieb installiert.Der Server überwacht dauernd den Port, für den er konfiguiert wurde, und nimmt sämtliche eingehenden Verbindungen auf diesem Port an.So bearbeitet ein permanent laufender Server Requests sehr schnell.
Sehr häufig benützt man bei der Konfiguration eines Servers virtuelle Hosts, dadurch ist ein Server in der Lage, Requests an verschiedenen Hosts zu bearbeiten, die durch DNS-Namen identifiziert werden.
Die grundlegende Idee des virtuellen Host besteht darin, jeden virtuellen Host eine eigene IP-Adresse zu reservieren.Im nächsten Schritt werden all diese IP-Adressen dem Rechner zugeordet auf dem der Server läuft.Der Server überwacht dann all diese IP-Adressen und kann somit die Verbindungs-Requests der verschiedenen virtuellen Hosts beantworten.
Ein grosser Nachteil des IP-basierten virtuellen Hosts liegt darin, dass jeder einzele Host seine eigene IP-Adresse besitzt, was einer Verschwendung von IP-Adressen entspricht.Dieser Nachteil führt zum Konzept des nicht auf IP basierenden Host.

3.1.Nicht auf IP basierende virtuelle Hosts

Das Konzept beruht auf der Protokollunterstützung zur Unterscheidung verschiedener virtueller Hosts.Bei HTTP/1.1 gibt es das Header-Feld Host, welches obligatorisch ist und den Host-Namen enthält, für den der Request gesendet wurde.
In dieser Konfiguration haben alle DNS-Einträge diesselbe IP-Adresse.Der Server überwacht nur eine IP-Adresse, und weil jeder Request die Identifikation enthält, für welchen virtuellen Host der Request gesendet wurde, kann der Server diese Information nutzen und für virtuellen Host antworten.

3.2.Unterschied von URL und Absoluten URL

Die URL ist ein Synonym für den DNS-Eintrag.Eine absolute URL enthält den Hostnamen, als auch Pfadinformationen.Enthält ein Request eine absolute URL, muss der Server das Header-Feld Host ignorieren und die Informationen der absoluten URL verwenden .

4. Schlusswort

Das HTTP ist eine Schlüsselkomponente des World Wide Web.Das HTTP/1.1 wurde stark vom Ursprung weiterentwickelt.Dennoch sind bei der Version 1.1 nicht alle Probleme gelöst.So ist die Unterstützung der Erweiterungen in Version 1.1 nicht durch alle HTTP-Server gewährleistet.

Quellen:

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

HTML
Hypertext Markup Language

Mischa Demarmels
mischade@ee.ethz.ch
23.5.2002

1 Einleitung

Hypertext Markup Language (*HTML*) heisst die Sprache, in der das Internet geschrieben ist. Sie wird durch einen Browser interpretiert und meist graphisch wiedergegeben (eine andere Darstellungsart wäre zum Beispiel eine akustische Wiedergabe eines HTML-Dokumentes z.B. für Sehbehinderte)

1.1 Ziele von HTML

Die Ziele von HTML lassen sich in drei Bereiche zusammenfassen:

- *Leistungsfähigkeit*
  HTML soll in vielen Bereichen eingesetzt werden können und es sollte eine möglichst grosse Anzahl von Anwendungen unterstützt werden.
- *Einfachheit*
  Trotzdem sollte die Sprache so einfach gehalten werden, dass es jedem Autor (auch nicht Informatikern) möglich wird, seine Vorstellungen durch HTML zu verwirklichen. Jeder sollte im Stande sein, seine Informationen z.B. übers Internet zu verbreiten.
- *Zugänglichkeit und Plattformunabhängigkeit*
  Um ein möglichst grosses Publikum zu erreichen, wurde bei der Entwicklung von HTML darauf geachtet, dass sich die Sprache mehr auf den Inhalt, als auf das Aussehen konzentriert. So wurde es möglich die Sprache plattformunabhängig zu gestallten.

2 Entwicklung von HTML

Die HTML-Beschreibungssprache wurde 1990 erfunden. In der ersten Zeit verlief das Wachstum des Web so schnell, dass die Sprache nie richtig dem letzten Stand der Entwicklung entsprach. Das vor allem auch deswegen, weil die verschiedenen Hersteller der HTML-Implementierungen selbst HTML weiterentwickelt und eigene Elemente hinzugefügt haben. Mittlerweile hat sich der momentan verwendete Standard (HTML 4.0) gut durchgesetzt.

2.1 Die Geschichte von HTML

2.1.1 Die ersten HTML-Versionen

Nach der ersten Planung im Jahr 1989, wurde Ende 1990 die erste Version von HTML am *CERN (European Laboratory for Particle Physics)* verfasst. Dazu gehörte bereits ein Graphischer Browser, der die Verwendbarkeit der neuen Sprache auf verschiedenen Plattformen veranschaulichen sollte.

Diese erste Version war noch recht rudimentär, doch gehörten viele elementare Funktionen, wie *Textüberschriften* auf verschiedenen Ebenen, *Ordered* und *Unordered Lists* oder auch *Hyperlinks*, bereits damals zum Konzept.

2.1.2 HTML 2.0

Nachdem das Web immer schneller wuchs und es immer mehr verschiedene Browser gab, die ihre eigenen Funktionen zur bestehenden Sprache hinzufügten, wollte man all dies in einer einzigen HTML-Version zusammenfassen, denn HTML war ja als Plattform unabhängige Sprache konzipiert worden. So wurde 1994 HTML 2.0 als neuer Standard festgelegt.

2.1.3 HTML 3.2

Kurz danach wurde Netscape gegründet. Die vielen HTML-Elemente die von diesem Unternehmen erfunden wurden führten einerseits zu einer rasanten Entwicklung der Sprache, andererseits aber auch zur fast unmittelbaren Veraltung des soeben eingeführten HTML 2.0. Das *World Wide Web Consortium* (*W3C*), welches in der Folge die Standardisierung übernahm, hatte bereits einen Entwurf für eine Version 3.0, welche jedoch nie verwirklicht wurde, weil sie noch vor ihrer Veröffentlichung als veraltet anzusehen war. Im Januar 1997 wurde schliesslich HTML 3.2 freigegeben. Diese Version verfügte nun auch über *Tabellen, Applets, Textfluss* um Bilder und vieles mehr.

2.1.4 HTML 4.0

Schon im Dezember des selben Jahres wurde die nächste und vorläufig letzte Version (HTML 4.0) veröffentlicht. Sie ist bis jetzt als Standard erhalten und wohl auch einigermassen komplett. Mit dieser Version werden nun auch *Style Sheets*, *Frames* und die Einbindung von *Multimedia-Objekten* unterstützt.

2.2 Document Type Definitions (DTDs)

Aus der Problematik, dass sich HTML nach und nach entwickelt hat und somit sehr viele Versionen der Sprache existieren, die immer noch benutzt werden, ergab sich die Einführung des Konzeptes der *Document Type Definitions* (*DTDs*). Hierbei geht es darum, dass HTML-4.0-Implementierungen in der Lage sein sollen, die veralteten Funktionen weiterhin interpretieren zu können. Es gibt drei solche DTDs:

- *Transitional DTD*
  Diese Definition enthält neben den HTML 4.0 Elementen auch eine Vielzahl alter Funktionen die von einem Browser noch interpretiert werden sollen.
- *Strict DTD*
  Diese Definition sollte zum Erstellen eines HTML-Dokuments verwendet werden, da nur die aktuellen Elemente enthalten sind.
- *Frameset DTD*
  Diese Definition ist nötig, wenn man mit verschiedenen Frames arbeitet.

3 HTML-Dokumente

Ein HTML-Dokument wird in zwei Teile gegliedert, den *Head* und den *Body*. Die einzelnen Elemente bzw. Funktionen der Sprache werden durch *Tags* aufgerufen, die ihrerseits verschiedene Parameter aufweisen können. Ein solches Dokument könnte zum Beispiel folgendermassen aussehen:

Fig. 1 Screenshot eines HTML-Dokumentes, wie es in einem Browser angezeigt wird (ohne den Browserrahmen).

3.1 Document Head

Nach dem das HTML-Dokument (Fig. 1) mit dem <*html*> Befehl geöffnet wurde, kommt als erstes ein Deklarationsteil. Er wird als *Document Head* bezeichnet und von <*head*> und </*head*> eingerahmt. Die Daten, die hier stehen, werden im Browserfenster nicht dargestellt. Sie dienen zum Beispiel dazu Suchmaschinen oder anderen Services Informationen über die Seite zur Verfügung zu stellen.

```
<html>

<head>

<title>unofficial aliengates homepage</title>
```

Der hier angegebene Titel wird oben auf dem Rahmen des Browserfensters angezeigt.

```
<meta name="keywords" content="death metal aliengates">
```

Die hier angegebenen Schlüsselwörter können zum Beispiel von einer Suchmaschine wie Google genutzt werden.

```
</head>
```

3.2 Document Body

Der eigentliche Inhalt der Seite wird in den *Document Body* geschrieben, der durch *<body>* und *</body>* eingerahmt wird.

```
<body bgcolor="#000000" text="#F0F0F0">
```

Schon im einleitenden <body>-Tag werden Attribute zum Aussehen der Seite angegeben. In diesem Beispiel sind dies die Hintergrundfarbe mit dem Befehl *bgcolor=“000000“* (Schwarz) und die normalerweise verwendete Textfarbe mit *text=“F0F0F0“* (fast Weiss).

```
<p align="center"><b><font face="Verdana, Arial, Helvetica" size="7">dies ist die inoffizielle
    homepage von:</font></b></p>
```

Der zwischen dem Tag * * und * * stehende Text wird in einem Absatz als Fliesstext dargestellt. Zusätzlich wird mit dem Attribut *align=“...“* die Ausrichtung des folgenden Textabschnittes bestimmt. In diesem Fall hier ist dies ein zentrierter Textabschnitt.

Mit dem *<font>-Tag* kann der nachfolgende Text mit diversen Attributen versehen werden. Der Befehl *face="Verdana, Arial, Helvetica"* gibt an welche Schriftart verwendet werden soll. Da jedoch nicht auf allen Computern sämtliche Schriftarten installiert sind werden gleich mehrere Möglichkeiten angegeben, auf die der Browser gegebenenfalls ausweichen kann. In diesem Beispiel wird, sofern vorhanden, *Veranda* verwendet. Ansonsten wird *Arial* und dann *Helvitica* genommen. Mit *size="7"* wird eine Aussage über die Schriftgrösse gemacht. Es gibt bei einem HTML-Dokument sieben verschiedene Schriftgrössen, wobei 1 die kleinste und 7 die grösste ist.

Die Textpassagen, die von ** und ** eingerahmt sind werden *bolt* (fett) angezeigt.

```
<p align="center"><a href="http://www.aliengates.com"><img src="logo.jpg" width="857"
    height="210" border="0"></a> </p>
```

Mit *<a href="http://www.aliengates.com">* wird ein sogenannter *Hyperlink* eingefügt. Durch Klicken auf das vom <a>-Tag eingerahmte Objekt wird im Browser die Seite angezeigt, deren Adresse im Tag steht.

Um ein Bild in die Seite einzubetten, verwendet man die Syntax *<img src="logo.jpg" width="857" height="210" border="0">*, welche das Bild *logo.jpg*, das im gleichen Ordner wie das HTML-Dokument gespeichert ist, einfügt.

```
<p align="center"><font face="Georgia, Times New Roman, Times, serif" size="4"><b>photos vom
    letzten gig:</b></font></p>

<div align="center">

<table width="22%" height="347">
  <tr>
    <td colspan="2"><img src="s06taptab05.jpg" width="267" height="200"></td>
    <td><img src="s07saintcity01.jpg" width="326" height="200"></td>
    <td><img src="s06taptab14.jpg" width="120" height="200"></td>
    <td><img src="s06taptab11.jpg" width="117" height="200"></td>
  </tr>
  <tr>
    <td><img src="s07saintcity07.jpg" width="132" height="202"></td>
    <td><img src="s07saintcity05.jpg" width="130" height="200"></td>
    <td><img src="s07saintcity02.jpg" width="325" height="200"></td>
    <td><img src="s07saintcity04.jpg" width="121" height="200"></td>
    <td><img src="s07saintcity06.jpg" width="117" height="200"></td>
```

```
   </tr>
</table>
```

Mit <table> und </table> werden Tabellen gekennzeichnet. Dazwischen werden der Aufbau der Tabelle und die in der Tabelle enthaltenen Daten angegeben. Man beginnt mit der ersten Zeile, *<tr>-Tag*, und gibt dann mit den *<td>-Tags* von links nach rechts alle Zellen an. Danach wird die erste Zeile mit *</tr>* geschlossen und es kommt die nächste Zeile.

```
<font size="6">mail to: <a href="mailto:mi@aliengates.com"><font color="#EFEFEF">
        mi@aliengates.com </font></a></font>
```

Hier sieht man einen weiteren Link, diesmal wird aber automatisch ein E-Mail-Programm gestartet und die angegebene E-Mail-Adresse als Empfänger eingesetzt.

```
</div>

</body>

</html>
```

Am Ende des Dokumentes wird zuerst der Document Body mit </body> und dann das ganze Dokument mit </html> geschlossen.

4 Die Zukunft von HTML

HTML war mit Sicherheit einer der Gründe, weshalb das Web so schnell gewachsen ist. Durch seine einfache Handhabung ist es fast jedem möglich Informationen in Form von HTML-Dokumenten zum Beispiel über das Internet zu verbreiten. Um heute aber eine dem Stand der Zeit entsprechende Webseite zu entwerfen, genügt HTML alleine oft nicht mehr.

Viel wird sich wahrscheinlich nicht mehr ändern an der heutigen HTML 4.0 Version. Dennoch wird ständig an neuen Möglichkeiten gearbeitet, mit denen eine Webseite noch besser darstellen werden kann. Dies geschieht zum Beispiel mittels Style Sheets oder Skripts, die in ein HTML-Dokument eingebettet werden können.

Eine wichtige Rolle in der Entwicklung der Darstellung von Webseiten spielt natürlich auch die immer schnelleren Internetverbindungen, die es ermöglichen grössere Datenmengen und somit kompliziertere Anwendungen innerhalb nützlicher Frist zum Benutzer zu bringen.

Obwohl sehr vieles möglich ist mit HTML, besteht die Zukunft des Webs sicher nicht mehr nur aus HTML.

5 Quellenangaben

1 PPS-Seminarunterlagen

2 www.w3.org: Homepage des W3C

3 Dirk Chung, Robert Agular: HTML GE-PACKT; MITP-Verlag, 2001 Bonn

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

XML
Datenstrukturierungssprache

Severin Hafner
hafners@ee.ethz.ch
18. Mai 2002

1 XML - Grundlagen

Im ersten Kapitel werden kurz zwei wichtige Grundfragen zu XML besprochen. Einerseits was man sich unter XML vorstellen soll und andererseits warum XML überhaupt nötig ist.

1.1 Was ist XML?

Die Abkürzung XML bedeutet eXtensible Markup Language (auf Deutsch: erweiterbare Markup Sprache). Gemeint ist damit eine neue, vom W3-Konsortium[1] entwickelte Datenstrukturierungssprache. Sie wird vor allem im Internet benutzt, wurde aber als allgemeine Datenstrukturierungssprache konzipiert. XML ist eine Teilmenge der SGML (Standard Generalised Markup Language), der Grundlage von HTML. Da SGML sehr komplex ist, wurde eine einfachere Variante erstellt, eben XML, welche aber immer noch die ganze strukturelle Leistungsfähigkeit beinhaltet.

1.2 Warum wurde XML entwickelt?

Eine neue Sprache zu entwickeln ist mit viel Zeit und Aufwand verbunden. Da man aber mit HTML bereits eine funktionierende Alternative hat, brauchte es einen triftigen Grund, damit XML entwickelt wurde. Diese Gründe will ich hier aufzeigen. Man hatte in vielen Bereichen festgestellt, dass HTML den gestellten Anforderungen nicht mehr genügte (z. B. können mathematische oder chemische Formeln nur als Bilddatei eingefügt werden). Auch bietet HTML nicht die Möglichkeit, eigene Dokumententypdefinitionen (DTD) zu erstellen, weil sie nur einen standardisierten Dokumententyp bereitstellt.[2] Zudem ist HTML sehr beschränkt, da in einem HTML-Dokument nur Angaben über die Darstellung im Browser gemacht werden können, nicht aber zur logischen Strukturierung der Daten. Beim untenstehenden Beispiel weiss der Computer zum Beispiel nicht, dass „Fritz“ ein Name ist.

Diese Gründe sollen an einem kleinen Beispiel veranschaulicht werden. Zuerst kurz ein HTML-Dokument, in Kapitel 2 wird dann ausführlicher und schrittweise die XML-Variante erklärt.

```
<p>
<b>Fritz </b>
<em>079 776 54 32 </em>
<br>
<b>Franz </b>
franz@ee.ethz.ch
</p>
```

Im Browser erscheint dann folgende Ausgabe:

Fritz *079 776 54 32*
Franz franz@ee.ethz.ch

2 Aufbau eines XML - Dokuments

In diesem Kapitel soll der Aufbau eines XML Dokuments anhand eines Beispiels aufgezeigt werden. Der zum HTML Beispiel analoge XML Code lautet:

[1] World Wide Web Consortium (W3C)
[2] Eigentlich sind es deren drei, aber da diese fest definiert sind, werden sie häufig als eine angesehen.

```
<list>
<item><name>Fritz</name><phone>079 776 54 32</phone></item>
<item><name>Franz</name><email>franz@ee.ethz.ch</email></item>
</list>
```

Die XML Version ist strukturierter und angenehmer zum Lesen. Das hängt einerseits damit zusammen, dass die Tags verständlicher benannt wurden und andererseits, dass zu jedem Start-Tag auch ein End-Tag vorhanden sein muss. Also die Liste wird mit `<list>` begonnen und mit `</list>` auch wieder beendet. In HTML ist das nicht zwingend. Oft fehlt das End-Tag und das Dokument kann nur dank toleranter Browser, welche fehlende Tags erraten, richtig dargestellt werden.

2.1 Dokumententypdefinitionen (DTD)

Wie bereits erwähnt, hat der Anwender von XML den Vorteil eigene und damit anwendungsspezifische Dokumententypen zu definieren. Eine DTD definiert, wie das Dokument aufgebaut ist, gibt also dem Dokument eine klare Struktur. Sie wird direkt ins XML Dokument geschrieben, oder aber als eigene Datei abgespeichert, welche den gleichen Namen hat, wie die Wurzel der DTD (hier `list[...]`). Eine mögliche DTD zum obigen Beispiel könnte wiefolgt aussehen:

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

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

Die erste Zeile legt fest, dass dieses Dokument ein XML Dokument der Version 1.0 ist. Diese DTD beginnt mit der Wurzel `list[...]`, deren Zweige aus Items (`item (...)`) bestehen. Ein Item setzt sich aus einem Namen und entweder einer Telefonnummer oder einer Email-Adresse zusammen.

2.2 Darstellung im Browser

Im Beispiel wurde die Struktur und der Inhalt des Dokuments festgelegt. Somit weiss nun der Browser um was für ein Dokument es sich hierbei handelt. Hingegen hat er noch keine Ahnung, wie er es in einem Browserfenster darstellen soll. Mittels Style Sheet wird ihm vorgegeben, wie das Layout aussehen soll. Der Standardtyp des Style Sheets ist bereits von der HTML her bekannt, das Cascading Style Sheet (CSS). Um in XML auf die gleiche Ausgabe zu kommen wie beim HTML Beispiel, muss das Style Sheet etwa so aussehen:

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

Diese Zeilen werden in einem stylesheet.css file abgespeichert. Der Beispielcode wird nach der ersten Zeile mit `<?xml-stylesheet href="stylesheet.css" type="text/css"?>` ergänzt, um dem Browser zu sagen, welches Style Sheet er zur Darstellung benutzen soll. Nun erscheint im Browserfenster die gleiche Ausgabe, wie sie bereits in HTML Beispiel aussah:

Fritz *079 776 54 32*
Franz franz@ee.ethz.ch

In der folgenden Abbildung wird noch einmal aufgezeigt, welche Komponenten benötigt werden, um ein XML Dokument im Browserfenster darzustellen.

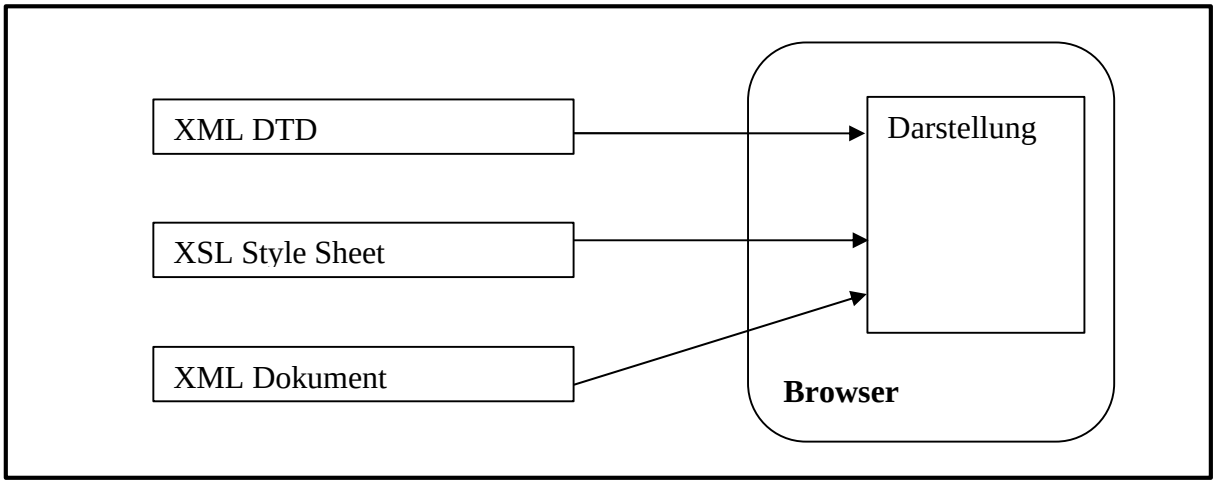

Fig. 1 Die Darstellung eines XML Dokuments im Browser

2.3 Weitere Anwendungen

Wenn man die beiden Beispiele betrachtet, stellt sich die Frage, warum XML überhaupt entworfen wurde, da man mit HTML ja viel einfacher und schneller zum Ziel kommt. Allein auf diese Beispiele bezogen ist das auch richtig. Für einfache Anwendungen lohnt sich die XML Version kaum. Für weiterführende Anwendungen hingegen erleichtert sie die Aufgabe ungemein. Bei einer Datenbank zum Beispiel ist man für die klar definierbare Struktur durch die DTD dankbar. Auch erkennt der Computer die Datenbank als Solches und es entstehen neue Anwendungsmöglichkeiten wie das Sortieren oder Durchsuchen.
Das W3C führte auch andere, auf XML basierende Beschreibungssprachen ein, wie z. B. MathML zur Darstellung mathematischer Formeln. In HTML konnten diese nur als Bilddatei eingefügt werden, sozusagen als Fremdkörper.

3 Konvertierung und Verknüpfung

Eine neue Datenstrukturierungssprache kann nicht auf einem Schlag eingeführt und durch die alte ersetzt werden. Der Datenaustausch zwischen der neuen und der bereits bestehenden Sprache muss gewährleistet sein und reibungslos funktionieren.

3.1 Konvertierung von und nach XML

3.1.1 XML und HTML

Der Datenaustausch zwischen XML und HTML ist wohl der wichtigste, weil der meist Gebrauchte. Dabei kommt es darauf an, in welche Richtung der Datentransfer stattfinden soll. Der Weg von XML nach HTML ist eher einfach, da die XML Dokumente bereits vollständig sind. Einige wenige Regeln genügen, um die XML-Tags in HTML umzuwandeln.

Einiges komplizierter ist die Gegenrichtung. Viele HTML Dateien sind fehlerhaft, z. B. fehlende End-Tags. Diese müssen zuerst durch erraten der Fehler vom Browser korrigiert werden. Anschliessend muss eine DTD für diese Dateien definiert werden, welche als XHTML bezeichnet wird.

3.1.2 Konvertierung anderer Formate nach XML

Bei der Datenumwandlung darf keine Information verloren gehen. Möchte z. B. eine Firma ihre eigenen Daten, sogenannte in-house Daten nach XML umwandeln, muss gewährleistet werden, dass bei diesem Vorgang nichts verloren geht. Dazu muss eine entsprechend geeignete, diese Bedingung erfüllende DTD erstellt werden. Ist dies gelungen, stellt die Konvertierung meist kein grosses Probleme mehr dar.

3.2 Links in XML Dokumenten

Natürlich muss XML auch die Möglichkeit bieten, verschiedene Dokumente mit einander zu verbinden, wie man das in HTML längst kennt. Auch in diesem Bereich bietet XML eine viel grössere Palette an Möglichkeiten als HTML, welche nur die Hyperlinks kennt. Diese erlauben nur eine Verknüpfung von zwei Dokumenten (z. B. zwei Homepages) und auch nur unidirektional, d.h. man kann ihnen nur in eine Richtung folgen. In XML ermöglicht die XML Linking Language (XLink) auch bidirektionale Links und Verknüpfungen zwischen vielen verschiedenen Dokumenten. Xlink definiert ein XML Umfeld, das es XML Anwendungen ermöglicht, Links innerhalb von Dokumenten zu erkennen und zu interpretieren. Xlink ermöglicht z. B. auch Links aus schreibgeschützten Dokumenten heraus.

4 Zusammenfassung und Ausblick

XML ist eine vom W3C entwickelte Datenbeschreibungssprache zur Strukturierung von Dokumenten. Sie ermöglicht die Definierung von DTD und dadurch eine selbst festgelegte Dokumentenstruktur. Dies ist der wesentlichste Unterschied zu HTML. Ein anderer besteht darin, dass in XML mit Xlink ein Umfeld definiert wurde, das im Gegensatz zum Hyperlink-Konzept von HTML bidirektionale Links erlaubt, und das Verknüpfen von mehreren Ressourcen ist ebenfalls möglich. Anders als in HTML, wo in vielen Dokumenten End-Tags fehlen, muss in XML zu jedem Start-Tag ein End-Tag vorhanden sein. So wird die Strukturierung gewährleistet. Zusammenfassend werden hier die drei wichtigsten Erneuerungen gegenüber HTML aufgelistet:

- Eigene DTD können genau auf das Dokument abgestimmt definiert werden.
- XML ist frei zugänglich und hängt nicht von der verwendeten Plattform oder Software ab.
- Xlink ermöglicht bidirektionale Links und es sind Links aus schreibgeschützten Dateien möglich.

Da XML viel mehr Möglichkeiten bietet, als HTML, ist die Sprache auch einiges komplexer. In nächster Zukunft wird wohl HTML die Oberhand behalten, denn für einfache Anwender lohnt sich der Mehraufwand nicht Es braucht immer eine gewisse Zeit, bis die potentiellen Benützer Vertrauen in eine Neuentwicklung gewonnen haben. Die Vorteile gegenüber HTML sind jedoch nicht zu übersehen und vor allem Diejenigen, welche die erweiterten Möglichkeiten benötigen, sind froh darauf zurückgreifen zu können.

5 Anhang

5.1 Quellenangaben

1. E.Wilde: World Wide Web – Technische Grundlagen; Springer Verlag, 1999 Berlin
2. www.selfhtml.org
3. www.w3c.org

5.2 Beispiel XML

Das Beispiel in XML im Überblick:

Das beispiel.xml – File:

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

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

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


<list>
<item><name>Fritz </name><phone>079 776 54 32</phone></item>
<item><name>Franz </name><email>franz@ee.ethz.ch</email></item>
</list>
```

Das stylesheet.css – File:

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

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

SSL & S-HTTP
sichere Kommunikation

Michael Schnellmann
mschnell@ee.ethz.ch
7. Juni 2002

1 Einleitung

Mit der Komplexität des Internets wächst auch die Gefahr, dass Daten von Unbefugten gelesen oder verändert werden können. Manipulationen kann man eventuell feststellen, indem man dem Datenpaket eine Prüfsumme mitgibt. Der Empfänger überprüft nun, ob die Prüfsumme mit dem Inhalt übereinstimmt.

Wird jedoch die Prüfsumme auch manipuliert, kann mit diesem Verfahren die Manipulation nicht detektiert werden. Der Datenaustausch kann auch von Drittpersonen überwacht werden, ohne den Datenfluss oder die Datenstruktur zu verändern. Um nun einen sicheren Datentransfer zu gewährleisten, müssen die Daten verschlüsselt werden. So sind sie für allfällige Lauscher die den Schlüssel nicht besitzen wertlos.

Eine sichere Kommunikation verlangt eine Datenübertragung, ohne dass Drittpersonen etwas verändern oder lesen können und Sender und Empfänger authentifiziert sind.

2 Kryptographische Verfahren

Damit eine Nachricht verschlüsselt und wieder entschlüsselt werden kann, muss der Sender und der Empfänger einen Schlüssel besitzen. Entweder brauchen beide einen geheimen Algorithmus oder einen geheimen Schlüssel.

2.1 Symmetrische Verschlüsselungsverfahren

Das symmetrische Verschlüsselungsverfahren (auch als konventionelles, Secret-Key- oder Private-Key-Verfahren bezeichnet) basiert auf einem gemeinsamen, geheimen Schlüssel. Es werden zwei Verfahren angewandt: Blockchiffren und Stromchiffren. Um heute eine genügende Sicherheit zu bieten, wird heute ein Schlüssel von mindestens 80 bit benötigt. Ansonsten ist die Gefahr gross, dass der Schlüssel innerhalb nützlicher Frist durch probieren geknackt werden kann. Doch mit zunehmender Rechenleistung müssen die Schlüssellängen nach oben korrigiert werden, um den Anforderungen an die Sicherheit zu genügen.

2.1.1 Blockchiffren

Das Blockchiffre-Verfahren arbeitet blockweise. Das Dokument wird in Blöcke (typischerweise 64 bit) aufgeteilt. Bei der Verschlüsselung spielt die Position einer Sequenz keine Rolle, das heisst, eine gleiche Sequenz ist auch nach der Verschlüsselung immer gleich.
Nach der Eingangsbehandlung (Aufteilung in Blöcke) wird die zentrale Kernfunktion angewandt. Diese ist abhängig vom Schlüssel und kann mehrmals durchlaufen werden. Dabei werden kleine Teilblöcke durch andere Bitfolgen ersetzt. In jeder Runde wird ein anderer Schlüssel, den Rundenschlüssel, verwendet, der sich vom eigentlichen Schlüssel ableitet.

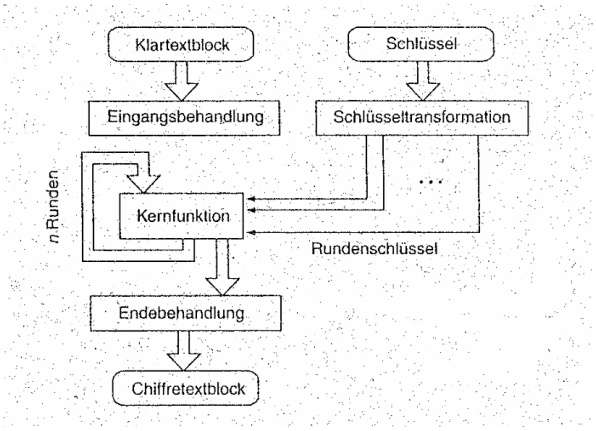

Bild 1 - Schematischer Verlauf des Blockchiffre-Verfahrens

Beispiele: DES (Data Encryption Standard), Schlüssellänge 56 bit
IDEA (International Data Encryption Algorithm), Schlüssellänge 128 bit

DES wurde in den Siebzigern entwickelt und gilt mit einer Schlüssellänge von 56 bit nicht mehr als sicher. Da aber der Algorithmus noch für gut befunden wurde, erweiterte man DES zu 3-DES. Das entspricht einem dreifachen Durchlauf mit zwei Schlüsseln (der erste und der letzte Durchlauf haben den gleichen Schlüssel). So bekommt man einen genügende Schlüssellänge von 112 bit.

2.1.2 Stromchiffren

Das Stromchiffre-Verfahren arbeitet bitweise. Mit Hilfe einer gleich langen Pseudozufalls-zahlenfolge, die vom Schlüssel und in manchen Fällen auch vom Klartext oder vom schon erzeugten Chiffretext abhängt, wird der Text mit einer einfachen Operation (normalerweise XOR) verschlüsselt. Der Vorteil der XOR-Funktion ist, dass sie auch gerade zum Entschlüsseln verwendet werden kann.

Der Hauptanteil der Verschlüsselung liegt also in der Erzeugung der Pseudozufallszahlenfolge. Verwendet wird dieses Verfahren bei RC4 (Schlüssellänge meist 128 bit) und bei A5 (54 bit). A5 wird zur Verschlüsselung von Sprachdaten in Mobiltelephonnetzen genutzt. Durch die bitweise Verschlüsselung können die Daten den Verschlüssler praktisch in Echtzeit durchlaufen. Allerdings kann nur ein Durchlauf vorgenommen werden.

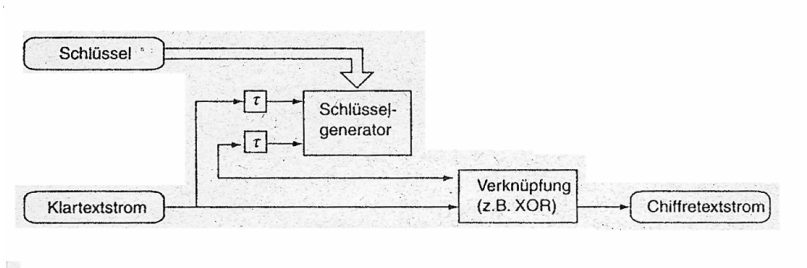

Bild 2 - Schematische Darstellung des Stromchiffre-Verfahren

2.2 Asymmetrische Verschlüsselungsverfahren

Im Gegensatz zu den symmetrischen Verfahren wird beim asymmetrischen für die Ver- und Ent-schlüsselung nicht der gleiche Schlüssel gebraucht. Diese beiden Schlüssel stehen in einem komplizierten mathematischen Zusammenhang. Es ist aber nicht möglich, den einen aus dem anderen herzuleiten.

2.2.1 Public-Key

Wenn für die Dechiffrierung einen anderen Schlüssel als für die Chiffrierung verwendet wird, so kann der Schlüssel für die Chiffrierung bekannt sein, da für die Entschlüsselung ein anderer Schlüssel gebraucht wird. Ist einer der beiden Schlüsseln öffentlich bekannt, so spricht man von einem Public-Key-Verfahren. Die Sicherheit hängt hier nicht nur von der Schlüssellänge ab (bei RSA 1024 bit), sondern hauptsächlich von den mathematischen Beziehungen der beiden Schlüssel. Deshalb können verschiedene asymmetrische Verfahren nicht nur anhand der Schlüssellänge verglichen werden.

Der zweite Schlüssel, der *private key*, darf aber auf keinen Fall in unbefugte Hände geraten. Denn dieser Schlüssel wird benötigt, um die Nachricht zu dechiffrieren. Um dies zu verhindern, schickt der Empfänger seinen öffentlichen Schlüssel dem Sender, dieser codiert das Datenpaket und der Empfänger kann nur mit seinem privaten Schlüssel die Daten wieder lesen.

Eine Möglichkeit für Unbefugte besteht dennoch, sofern sie ein eigenes Public-Private-Key-Paar besitzen. Sie fangen den öffentlichen Schlüssel des Empfängers ab und senden stattdessen den eigenen an den

Sender. Das codierte Packet wird wiederum abgefangen, decodiert und mit dem Schlüssel des Empfängers codiert. So können die Daten gelesen werden, ohne dass Sender und Empfänger etwas bemerken.

2.3 Hash-Funktion

Ein wesentliche Bestandteil von Verschlüsselungsprotokolle ist die Hash-Funktion. Sie ist vergleichbar mit einer Prüfsumme, jedoch viel höheren Anforderungen. Aus einem Text beliebiger Länge entsteht der Hash-Wert, der 128 bit (manchmal auch 160 bit) lang ist. Eine Bedingung ist, dass nicht aus zwei verschiedenen Texten der gleiche Hash-Wert resultieren kann. Die zweite Bedingung ist, dass aus dem Hash-Wert der Text nicht wieder generierbar ist. Ist die erste Bedingung nicht erfüllt, so spricht man von einer Kollision. Da aber die Länge des Hash-Wertes festgelegt ist, sind Kollisionen nicht zu vermeiden. Deshalb muss die erste Bedingung anders formuliert werden: Es muss unmöglich sein, zwei Texte mit gleichem Hash-Wert zu generieren.

Wird beim Text ein Bit verändert, so ändern sich beim Hash-Wert durchschnittlich die Hälfte aller Bits. Der Hash-Wert wird auch als Fingerabdruck bezeichnet. Sind von 2 Texten die Hash-Werte gleich, so kann man mit einer sehr hohen Wahrscheinlichkeit ausgehen, dass die Texte identisch sind. Für eine Länge von 128 bit beträgt die Wahrscheinlichkeit, dass zwei Texte den gleichen Hash-Wert besitzen 2^(-128) = 2.9*10^-39 !

3 SSL - Secure Socket Layer

SSL wurde von Netscape entwickelt, um sichere HTTP-Kommunikationskanäle zu ermöglichen. Es handelt sich dabei um ein Public-Key-Verfahren. Es befindet sich im Schichtenmodell oberhalb der Transportschicht oder wird sogar dieser zugerechnet. Die Anforderung an das Netzwerk ist ein verbindungsorientiertes Transportprotokoll. In den meisten Fällen ist das eine TCP/IP-Verbindung. Die Verschlüsselung erfolgt mit symmetrischem oder asymmetrischem Verfahren. Dabei sind vorbestimmte Kombinationen beider Methoden (*ciphersuites*) möglich. Das Protokoll selber arbeitet aber asymmetrisch und unterscheidet zwischen Server und Client. Der Verbindung wird vom Client mittels *handshake* aufgenommen. Er übermittelt dem Server die unterstützten Verschlüsselungsalgorithmen. Der Server bestimmt nun eine Verschlüsselungsart und die dazugehörigen Schlüssel. Den öffentlichen Schlüssel schickt er zusammen mit einem Zertifikat, mit welchem der Client überprüfen kann, ob wirklich der gewünschte Server antwortet, dem Client. Der Client erzeugt nun einen Sitzungsschlüssel, der als *private key* benutzt wird.

SSL ist selber in zwei Unterschichten aufgeteilt: Zu einem der Rekord Layer, der die Datenströme dechiffriert oder dechiffriert und sie dem verbindungsorientierten Transportlayer weiterleitet, zum anderen das Handshake-Protokoll, das die beiden Kommunikationspartner authentifiziert und das Codierverfahren vereinbart.

Mittlerweile dient SSL nicht nur der Verschlüsselung von HTTP-Daten, sondern wird auch für andere Dienste wie FTP oder News verwendet. Will ein Benutzer eine sichere HTTP-Verbindung mit SSL aufbauen, so ruft er die gewünschte Seite mit https://www. ... auf.

4 S-HTTP

Neben HTTPS (HTTP mit SSL-Verbindung) existiert auch noch S-HTTP. Im Gegensatz zu SSL ist S-HTTP direkt in der Anwendungschicht angesiedelt. Es stellt eine Erweiterung des HTTP-Protokoll dar. Zur Verschlüsselung verwendet S-HTTP auch Kombinationen von symmetrischen und asym-metrischen Verschlüsselungen. S-HTTP kann auch nur symmetrisch arbeiten. Das setzt jedoch voraus, dass beide Kommunikationspartner den gleichen geheimen Schlüssel besitzen.

In der Praxis hat sich S-HTTP aber noch nicht durchsetzen können. Auf Server-Seite gibt es zwar schon gute kommerzielle Programme, aber für Clients ist der Durchbruch noch nicht geschaffft, da bis jetzt weder Netscape noch Microsoft das Protokoll in ihre Browser integriert haben (Stand Mai 1997). Heute ist S-HTTP kein Standard und wird deshalb nicht mehr unterstützt. URLs mit shttp://.. gehören also der Vergangenheit an.

5 Schlusswort

Die Verschlüsselungstechnologien im Internet werden je länger je mehr gebraucht. Sei es, um eine Kreditkartennummer zu übermitteln, oder mit dem Mobiltelephon das Badwasser einlaufen lassen, die Punkte Identifikation (Authentifizierung), Abhörsicherheit sowie Schutz vor Manipulation müssen gewahrt sein. Da aber mit zunehmender Leistungsfähigkeit der Rechner Verschlüsselungen schneller geknackt werden können, müssen die kryptographischen Methoden auch immer besser werden.

6. Quellen

[1] Abgegebenes Material vom Assistenten
[2] http://141.48.65.161/html-kurs/Artikel/ssl-m.htm
[3] http://www.infoserversecurity.org/shttp.php
[4] www.klinikum.rwth-aachen.de/cbt/hypertext/shttp.htm

PPS-Seminar
Grundlagen der Internet-Technologie, SS 02

Elektronische Post Im Internet

Bernhard Wasser
bwasser@student.ethz.ch
7. Juni 2002

1 Senden und Empfangen von E-Mail

Mit dem Siegeszug des Internets hat sich auch mehr und mehr die E-Mail durchsetzten können, diese hat die Kommunikation in vielen Bereichen revolutioniert. Früher kam E-Mail meist nur in wissenschaftlichen Kreisen zum Einsatz, unterdessen hat dieses Medium jedoch auch im privaten Bereich seinen Platz eingenommen. Zusammen mit der SMS-Technologie ersetzt es Teile von älteren Formen der asynchronen Kommunikation, wie Briefe, Anrufbeantworter und Faxgeräte.

Weil E-Mail eine der ersten Internet-Anwendungen war, basiert die Technologie auf eher einfachen Modellen. Wenn man E-Mail als eine Reihe von Protokollen betrachtet, kann man eine Unterteilung in zwei grosse Bereiche vornehmen: Das Problem Nachrichten durch das Internet an eine Empfängeradresse zu senden und die Frage, wie E-Mail-Nachrichten durch Benutzer vom Mail-Server abgeholt werden können.

1.1 Senden von E-Mail

Das wohl am meisten verbreitete Protokoll für den E-Mail-Transfer ist das Simple Mail Transfer Protocol (SMTP). Davon gibt es noch eine zweite Variante namens Extended SMTP. Beide Protokolle werden in den folgenden Abschnitten erörtert.

1.1.1 Simple Mail Transfer Protocol

Das SMTP ist das Internet-Standardprotokoll zur Übertragung von E-Mail-Nachrichten und funktioniert eigentlich recht einfach. Der Rechner des Absenders nimmt mittels TCP-Verbindung Kontakt zum SMTP-Port 25 des Ziel-Rechners auf und übermittelt die Mail.

Damit die Botschaft ankommt, muss der Absender lediglich die Adresse des Empfängers kennen. Diese ist nach dem Schema username@subdomain.top-level-domain aufgebaut, wobei letztere zum Beispiel für die Schweiz „ch“ lautet. Natürlich sind noch etliche Verschachtelungen möglich, die per „.“ unterteilt werden. Das Domain Name System (DNS) sorgt dann dafür, dass dieser Adresse die richtige IP-Adresse zugeordnet wird.

Es ist nicht die Aufgabe eines Mail-Clients das ganze Mail-Routing vollständig zu beherrschen. Statt dessen gibt es eine Arbeitsteilung: Der Mail-Client, auch Mail User Agent genannt, kümmert sich um die Kommunikation mit dem Benutzer und übergibt zu versendende Mails an einen Mail Transport Agent (MTA). Die meisten User kommunizieren heute über eine Wählerverbindung mit ihrem Internet-Provider. Die Schnittstelle zwischen dem Mail-Client und den beim Provider stationierten MTA ist dann eben das SMTP. Abgehende Mail wird per SMTP beim MTA des Providers deponiert, der sich dann um alles weitere kümmert.

Das vorläufige Endziel auf dem Weg durch das Internet ist dann wiederum ein MTA. Dort bleibt die Mail so lange liegen, bis der Empfänger sie beim Client abruft.

1.1.2 Extended SMTP

Das einfache SMTP-Protokoll hat sich in einigen Anwendungsbereichen als ungenügend erwiesen. Die Ergänzung, mit der sich SMTP leicht erweitern lässt, wird Extended SMTP (ESMTP) genannt. Unterdessen wurde eine grosse Anzahl Erweiterungen definiert – die Rahmenbedingungen von ESMTP machen es Implementierungen möglich, neue Erweiterungen einzuführen, ohne die existierenden Standards ändern zu müssen. Erweiterungen machen jedoch nur Sinn, wenn sie von beiden Seiten unterstütz werden.

Ursprünglich zielte ESMTP auf einige Schwächen von SMTP. Die populärsten Erweiterungen werden in der folgenden Liste aufgeführt:

- *Nachrichten mit 8-bit-Zeichensätzen*
  SMTP erkennt nur ASCII-Zeichen, wodurch es möglich ist, dass Nicht-ASCII-Zeichen in E-Mails verfälscht werden können. In ESMTP wurde dieses Problem erfolgreich gelöst.
- *Ankündigen der Nachrichtengrösse*
  Da E-Mail-Nachrichten sehr umfangreich werden 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 in der angekündigten Grösse akzeptieren möchte. Dies ist speziell für User mit einer langsamen Internet-Leitung hilfreich.
- *Umfangreiche und binäre Nachrichten*
  Beim Senden umfangreicher Nachrichten kann es sinnvoller sein, sie in mehrere Teile zu zerlegen. Ausserdem sollte man einen Mechanismus zum Steuern dieses Prozesses haben, der es ermöglicht, das Senden eines Nachrichtenteils zu wiederholen, sollte die Übertragung fehlerhaft gewesen sein.

Zusätzlich zu den erwähnten Erweiterungen existieren natürlich noch eine ganze Reihe anderer.

1.2 Empfangen von E-Mail

Obwohl das Senden von E-Mail für eine erfolgreiche Nachrichtenübertragung ganz wichtig ist, muss auch das Problem des Empfangens gelöst werden. Früher entsprach das Lesen von E-Mail dem Lesen einer Datei, in die das SMTP-Programm alle ankommenden Nachrichten schrieb. Einen Schritt weiter geht das Definieren von Netzwerkprotokollen zum Zugriff auf E-Mails, die auf dem Mail-Server gespeichert sind. Durch ein solches Protokoll kann der Zugriff auf den Mail-Server über ein Netzwerk erfolgen, ohne dass die Notwendigkeit eines gemeinsamen Dateisystems besteht.

Die beiden populärsten Protokolle, das Post Office Protocol (POP) und das funktionell umfangreichere Internet Access Protocol (IMAP) werden in den folgenden Abschnitten vorgestellt.

1.2.1 Post Office Protocol (POP)

Nachdem eine Nachricht vom Urheber zum Mail-Server des Empfängers übertragen wurde, wird sie in dessen Postfach gespeichert, bis er sie liesst. Normalerweise nimmt der Empfänger nur gelegentlich Kontakt zum Mail-Server auf und fragt mit seinem E-Mail-Programm neue Nachrichten ab, analog zum Gang zur Post. Passend zu diesem Verhalten bekam das populärste Protokoll seinen Namen – Post Office Protocol.

Die grundlegende Funktionsweise von POP ist sehr einfach. Der POP-Server wartet auf eingehende Verbindungen auf einem bestimmten Port - der Standart-Port für POP ist Port 110 – zu dem der Client eine TCP-Verbindung aufbaut, wenn eine POP-Sitzung erwünscht wird. Die Authentifizierung kann bei POP auf verschiedene Arten erfolgen. Der einfachste Weg ist das Verwenden der POP-Befehle *USER* und *PASS.* Dabei werden allerdings Benutzername und Passwort unverschlüsselt über eine TCP-Verbindung übertragen, was ziemlich unsicher ist. Eine sicherere Methode zur Authentifizierung geschieht mittels des optionalen Befehls *AUTH.*

Nach der Authentifizierung kann der Client eine Reihe von POP-Befehlen absetzen. Die Anzahl Befehle ist sehr klein, so dass ein POP-Server oder –Client recht einfach zu implementieren ist. Die einzigen Befehle beim Zugriff auf das Postfach dienen zum Abholen oder Löschen einer Nachricht. Um mit POP zu arbeiten, muss der User also die Nachrichten erst vom Server downloaden, es werden keine Kopien auf dem Server gespeichert (Offline-Betrieb).

1.2.2 Internet Message Access Protocol (IMAP)

Da POP andere Betriebsarten als den Offline-Betrieb nur sehr begrenzt unterstüzt, wurde ein neues Nachrichtenzugriffsprotokoll entwickelt. Diese Protokoll hiess ursprünglich Interactiv Mail Access Protocol, wurde dann aber in Internet Access Protocol (IMAP) geändert.

Die wesentlichen Vorteile von IMAP gegenüber POP kann man wie folgt zusammenfassen:

- *Unterstützung verschiedener Ordner*
  IMAP bietet die Möglichkeit, neben dem Ordner für eingehende Nachrichten noch weiter Ordner anzulegen und zu bearbeiten. Funktionen wie Auflisten, Erstellen, Löschen und Umbenennen von Ordnern werden unterstützt. IMAP unterstützt ebenfalls Ordnerhierarchien, die ähnlich funktionieren wie Ordner innerhalb eines Dateiensystems.
- *Ordnerbearbeitung über das Netzwerk*
  Nachrichten können von einem Ordner in einen andern verschoben, oder als gelesen wie auch beantwortet markiert werden. Ausserdem ist es möglich, gemeinsame Ordner so zu aktualisieren, dass die anderen Benutzer von der Aktualisierung in Kenntnis gesetzt werden.
- *Optimierte Online-Performance*
  Da Emails sehr umfangreiche Teilstücke wie Bilder oder Videos enthalten können, erlaubt IMAP das einzelne Abholen von Teilstücken (MIME-Technologie). Ausserdem ist es nicht nötig die Nachrichten zu übertragen, wenn man sie durchsuchen will.

Obwohl IMAP im Vergleich zu Pop funktionell klar überlegen ist, müssen zwei Nachteile erwähnt werden. Das Protokoll ist erheblich komplexer als POP und deshalb schwerer zu Implementieren. Insbesondere bei Systemen mit sehr begrenzten Ressourcen kann diese eine wichtige Rolle spielen. Da IMAP neuer ist, und Funktionen bietet, die nicht in allen Anwendungsbereichen benötigt werden, wird IMAP derzeit in geringerem Masse unterstützt als POP. Allerdings nimmt die Verbreitung von IMAP stetig zu, und man kann davon ausgehen, dass die Unterstützung sich in Zukunft verbessern wird.

2 Sicherheit im E-Mail-Verkehr

Die E-Mail-Technologie erlaubt es Nachrichten an Personen zu schicken, die man weder kennt, noch weiss wo sie wohnen. Dabei vergisst man leicht, dass die elektronischen Nachrichten nicht unmittelbar beim Empfänger landen, sondern schon auf ihrem Weg durchs Netz abgefangen und gelesen werden können. Die Absender-Adresse einer Email muss nichts über den Absender aussagen, und ist nicht speziell vertrauenswürdig.

2.1 Verschlüsselung und Authentifizierung

Um die Sicherheit zu erhöhen kann man die Nachrichten verschlüsseln, oder digital signieren um die Authentizität zu verifizieren. Dies bereitet aber noch immer Probleme, da Standards zwar existieren, aber noch lange nicht in jeder Software implementiert sind.

2.1.1 Secure Multipurpose Internet Mail Extension

Kurz S/MIME genannt, stellt diese Technologie Funktionen zur Verfügung, die dank kryptographischen Diensten für Verschlüsselung, Authentifizierung und Vertraulichkeit sorgen. S/MIME ermöglicht Verschlüsselung und Signierung von E-Mails zwischen unterschiedlichen Mail-Plattformen und Betriebssystemen.

Der User kann bei einer Zertifizierungsstelle einen kryptographischen Schlüssel plus das dazu passende Zertifikat anfordern, eine Art digitaler Personalausweis. S/MIME Anwendungen unterstützen eine sogenannte flache Zertifierungshierarchie, das heisst, eine Zertifizierungsstelle steht an der Spitze, und die von ihr ausgestellten Userzertifikate bilden bereits die unterste Stufe der Hierarchie.

Für Nachrichten die der Absender nur signiert, aber nicht verschlüsselt, stehen nach S/MIME zwei verschiedene Formate zur Verfügung. Entweder verschickt die Software die Signatur getrennt von den signierten Daten als Anhang. Der Vorteil besteht darin, dass der Empfänger die eigentliche Nachricht auch ohne S/MIME Support lesen kann. Anderseits läuft man Gefahr, dass Formatkonvertierungen durch Mail-Server die Signatur ungültig machen, weil der empfangene Text nicht mehr mit dem ursprünglich signierten übereinstimmt. Alternativ kann man die Signatur zusammen mit der Nachricht in einen binären Anhang verpacken. Dadurch ist die Mail aber ohne S/MIME-Decoder nicht mehr lesbar.

2.1.2 OpenPGP

S/MIME und PGP (Pretty Good Privacy) verfolgen prinzipiell denselben Zweck. Auch in den verwendeten Krypto-Algorithmen weichen die beiden Standards kaum voneinander ab. Der Unterschied ist, bei S/MIME gibt es keinen Schlüssel ohne Zertifikat, und ein einheitliches Bild davon, welche Schlüssel als authentisch anzusehen sind und welche nicht. OpenPGP überlässt hingegen die Entscheidung über das Vertrauen in fremde Schlüssel klassischerweise dem Endanwender. Zudem kann jeder PGP-User selbst als Zertifizierungsinstanz auftreten. Dadurch ergibt sich ein „Web to Trust“ in dem sich verschiedene Anwender aufeinander verlassen müssen.

2.2 Gefahren

Kryptographie ist bei normalen E-Mails sicher nicht nötig. Bei Vertraulichen Daten ist es sehr ratsam dies anzuwenden, denn es ist wirklich nicht all zu schwer E-Mails abzufangen. Amerikanische Nachrichtendienste sind zum Beispiel in der Lage ein Programm bei einem Provider installieren zu lassen, das dann ohne weiteres auf alle Mails zugreifen, sie nach Stichworten durchfiltern und natürlich auch abspeichern kann. Auch User die nicht vor den Nachrichtendiensten zu verbergen haben, könnten sich daran stören. Ein E-Mail sollte ja ähnlich dem Brief als persönliche Post angesehen werden. Ein anderes ernst zu nehmendes Problem im E-Mail verkehr sind Viren und Trojaner. Gefährlich sind Viren, weil sie sich zum Teil selbständig weiterschicken können, und sich deshalb extrem verbreiten können; der Schaden der ein einzelner Virus verursacht kann in Millionenhöhe gehen. Ganze Systeme von Grossfirmen können blockiert oder vorübergehend lahmgelegt werden. Als Privat-User muss man sich vor allem vor Datenverlust fürchten. Als verantwortungsbewusster sollte man einige Sicherheitsregeln beachten:

- *Dateianhänge*
  Dateien von unbekannten Personen sollte man nicht, oder nur mit besonderer Vorsicht öffnen. Ein Virenscanner ist unbedingt nötig in solchen Fällen. Auch Dateien die von Freunden und Bekannten verschickt werden sollte man durchchecken, denn heutige Viren können sich automatisch an die ersten x Namen im Adressbuch weiterschicken.
- *Information*
  Sollte ein gerade ein gefährlicher Virus zirkulieren und Schäden in grossem Ausmass verursachen, wird dies oft in den Medien oder auf speziellen Internet-Sites erwähnt. In solchen fällen

ist erhöhte Vorsicht nötig, denn nicht nur der User selbst kann darunter leider, dank des Schneeballsystems ist es möglich dass noch etliche Andere User infiziert werden.

- *Updates*
  Anti-Viren-Programme können oft kostenlos auf den neusten Stand gebracht werden, denn neue Viren verlangen neue Abwehrsysteme. Diese Option sollte genutzt werden um optimalen Schutz zu garantieren.

3 Spam

Der Ursprung und die genaue Definition des Ausdrucks sind umstritten. Ursprünglich verstand man unter Spam einen Ausdruck für exzessives Posten desselben Artikels in Newsgroups. Heute jedoch bezeichnet man damit allgemein unverlangte Massensendungen von E-Mail. Dabei ist der Inhalt der Mail nicht von Belang. Alleine die Tatsache, dass die Mail unverlangt und in grosser Menge versandt wird, macht es zu Spam.

3.1 Folgen von Spam

Für den einzelnen User entsteht zwar kein direkter Schaden, aber dennoch sind Spam-Mails extrem nervig, sei es irgend ein nicht-abonnierter Newsletter, Werbung von irgendwelchen Firmen, Pyramidenschemen, Software zum Verschicken von Spam, Wunderkuren oder gar pornographisches Material. Dies kann unter Umständen so weit gehen, dass mehr Müll auf dem Mail-Account liegt als tatsächliche Nachrichten. E-Mail-Spam ist einzigartig in der Hinsicht, dass der Empfänger viel mehr dafür bezahlt als der Absender. Der Absender bezahlt nur ein Konto bei einem Provider und ein paar Minuten bis Stunden Telefongebühren. Der ganze Rest wird vom Empfänger bezahlt. Dabei ist nicht nur an die Zeit und Telefongebühren zum Herunterladen des Spams zu denken. Spam verstopft auch die Leitungen und die Mailserver des Providers. Spams mit ungültigen Adressen erzeugen Fehlermeldungen, die der Provider erhält. Alle diese Kosten des Providers werden natürlich über die Gebühren auf die Kunden überwälzt. Aber auch die Zeit des Empfängers ist ein Faktor. Das ist kein Problem bei einem einzigen Spam, bei mehreren wird es schnell ärgerlich und zeitraubend.

Die meisten Spammer verwenden nicht ihre eigenen Mailserver; entweder haben sie gar keine eigenen Server, oder aber ihre Server werden von anderen Providern geblockt. Sie versenden ihre Spams einfach über Server, die gar nicht ihnen gehören. So ein Missbrauch kann einen Server lahm legen, er kostet auf jeden Fall dem betreffenden Provider hunderte bis tausende Franken in verlorener Bandbreite, verschwendetem Diskplatz so wie Zeit zur Bearbeitung von Fehlermeldungen. Viele Spammer eröffnen Kontos mit dem einzigen Zweck, damit Spams zu versenden. Sobald der Spam weg ist, machen sie sich aus dem Staub - die Rechnung bezahlen sie nie. Der Provider kann dann in oft stundenlanger Arbeit aufräumen - Fehlermeldungen löschen und Reklamationen beantworten.

3.2 Schutz gegen Spam

Zuerst gilt es herauszufinden, von wem der Spam kommt. Das tönt einfach, ist es aber nicht immer. Manchmal stimmt der "From:"-Header nämlich nicht. Viele Spammer fälschen diese Adresse, damit sie nicht alle Fehlermeldungen erhalten und damit es schwieriger ist, sie zu finden. Manchmal wird sogar jemand, dem man eine auswischen will, als Absender eingesetzt - die betreffende Person wird dann mit Fehlermeldungen und Beschwerden überschwemmt. Wenn der Absender "kurios" aussieht, stimmt er wahrscheinlich nicht. Deshalb lohnt es sich den Hauptteil des Mails, den sogenannten Body, anzuschauen. Oft haben die Spammer hier ihren richtigen Absender. Oder sie machen Werbung für ihre Homepage. Sobald der Provider bekannt ist, lohnt es sich eine Mail mit allen Informationen und Kopien an jenen zu schicken. Dabei sollte nicht vergessen werden, dass der Provider wohl auch bloss Opfer und nicht Täter ist.

Welche der Standards sich im Bereich E-Mail schlussendlich durchsetzt, wird sich zeigen. SMTP und POP sind einfacher zu implementieren, und basieren auf einfacheren Systemen. ESMTP und IMAP bieten mehr Funktionen und sind benutzerfreundlicher, werden aber nicht überall unterstützt. Ein weiterer Nachteil ist, dass sie deutlich mehr Ressourcen benötigen. Wie so oft im Bereich Software liegt es wohl bei den Anbietern. Solange die neuen Technologien nicht breit unterstützt werden, ist nicht mit einer Ablösung der alten Standards zu rechen.

Die Sicherheit wird in Zukunft wohl noch wichtiger werden. Bei Zahlung per Kreditkarte oder bei Übermittlung wichtiger Unterlagen muss der User sicher sein, dass keine ungebetenen Gäste mitlesen. Vorher wird sich das Einkaufen per Mausklick nicht wirklich durchsetzen können. Natürlich braucht der Durchschnitts-User kein Verschlüsselungsprogramm. Im Bereich Wirtschaft oder Politik jedoch, wo auf Diskretion viel Wert gelegt wird, kann Verschlüsselung sicher ein Vorteil sein, denn der sichere E-Mail-Verkehr ist ein Mythos.

Gegen Viren helfen nur gute Anti-Viren-Software und Anwender die mitdenken. Jeder ist selber dafür verantwortlich, dass sein PC nicht zur Verbreitung von Viren missbraucht wird. Da die Virenprogrammierer den Anti-Viren-Entwicklern immer einen Schritt voraus sind, muss auch in Zukunft mit grossräumigem Virenbefall gerechnet werden. Weniger bedrohlich aber nicht minder nervtötend ist der Spam. Es ist zu Vermuten dass diese Schleichwerbung in Zukunft noch deutlich zunehmen wird, da Aufwand, Kosten und Gefahr bestraft zu werden sehr gering sind. Solange dies nicht gesetzlich besser geregelt wird, kann man nicht mehr tun, als den Provider um Hilfe bitten, damit dieser die Spammer verwarnen kann.

Referenzen

1. Ausgeteiltes Material
2. http://www.heise.de
3. http://www.trash.net/~sam/spam