Bundesoberstufenrealgymnasium
Innsbruck

Fachbereichsarbeit im Rahmen des
Unterrichtsgegenstandes **Informatik**:

"DIE TCP/IP-PROTOKOLLSUITE"

Eingereicht von: Markus Nolf
Klasse 8e

Eingereicht am: 20.2.2003

Benotung: Sehr Gut

Betreuende Lehrperson: Mag. Rudolf Ladinig

"Die TCP-IP Protokollsuite"
Inhaltsverzeichnis

00 Vorwort

Das Internet ist nach wie vor auf dem Vormarsch, und immer mehr Funktionen werden erfolgreich im Globalen Netz angeboten:
Aktuelle Informationen und Nachrichten sind weltweit in Sekundenschnelle abrufbar, riesige Lexika finden sich sogar schon im Internet. Die Bedeutung von Onlineshopping – 7 Tage die Woche, 24 Stunden "Zuhause einkaufen" – wächst, obwohl langsamer als ursprünglich verhofft.
Mit Internet-Banking kann man seine Finanzen von Zuhause aus regeln, der online Pizza-Service stellt die bestellten Gerichte bis vor die Haustüre zu, und mit Programmen wie ICQ (Kurzform für "I seek you") kann man gratis Kontakte auf der ganzen Welt pflegen.
Auf den Webseiten von Kinos kann man die gewünschten Kinokarten bestellen und gleich Zuhause ausdrucken.
Der gratis Email-Dienst verdrängte bereits die Telegramme und löst jetzt langsam die Briefpost ab, sogenannte Videokonferenzen ersparen oft eine Reise und sind doch persönlicher als ein simples Telefonat.
Durch Musiktauschbörsen (inzwischen eigentlich Multimediatauschbörsen – auch Videos, Bilder, Programme etc. können getauscht werden!) kann man sich einzelne Lieder oder ganze Alben auf einmal gratis herunterladen.

Der Zugang zum Internet wird immer selbstverständlicher, und die Anwendungen werden vielfältiger und noch einfacher zu handhaben.
Doch kaum jemand, der über einen Computer ins Internet einsteigt, denkt an die ungeheure Arbeit, die dahinter steckt um so ein riesiges Mega-Netzwerk aus vielen kleineren Netzwerken zum laufen zu bringen.
Die wenigsten fragen sich, wie die Kommunikation zwischen Computern eigentlich funktioniert und was zwischen dem Klicken auf einen Link und der Nachricht "Seite erfolgreich geladen" unten in der Statusleiste eines Browsers wirklich passiert.

Ich habe das Thema "Die TCP/IP Protokollsuite" für meine Fachbereichsarbeit gewählt, weil ich schon immer wissen wollte, wie das Internet auf der unteren Softwareebene arbeitet.
Ich finde es faszinierend, wie zwei Computer in einem Netzwerk Kontakt aufnehmen und eine "gemeinsame Sprache" aushandeln.
Es ist beeindruckend für mich, wie Texte, Bilder, Videos und vieles mehr von einem Punkt der Erde zu praktisch jedem Anderen in digitaler Form übertragen werden können, bei mitunter atemberaubender Qualität und moderater Geschwindigkeit.

Die folgende Arbeit soll einen Einblick in diese Funktionsweise geben. Sie soll zeigen, wie Daten verändert werden, um als binäres Signal übertragen und auf der anderen Seite schließlich wieder zusammengebaut zu werden, und sie soll die Aufgaben der wichtigsten Protokolle aufzeigen.

01 Das TCP/IP-Bündel

In diesem Kapitel möchte ich einige grundlegende Dinge von Netzwerkadressen und -klassen bis hin zur DNS festhalten.

01.1 DIE IDEE

Die TCP/IP-Protokollsuite wurde in den Sechzigerjahren des 20. Jahrhunderts vom Verteidigungsministerium der Vereinigten Staaten entwickelt. Ziel war es, ein dynamisches Netzwerk aufzubauen, das Daten von jedem beliebigen Punkt zu jedem anderen senden kann, ohne Rücksicht auf Verbindungstypen (Kabel, Glasfaser, Funk, ...). Wenn der direkte Weg zum Ziel versperrt würde, sollten sich die Datenpakete einfach einen anderen Weg aussuchen können.
Unter diesen Vorraussetzungen entstand das TCP/IP-Modell, das sich seither weit über seine eigentlichen Ziele und Erwartungen hinaus entwickelt hat. Seit den 1990ern ist TCP/IP die am weitesten verbreitete Kommunikationsform für Computer.

Heutzutage kann jeder Computer, egal von welcher Herstellermarke und unter welchem Betriebssystem, mit jedem anderen in einem Netzwerk kommunizieren.
Auch als Basis des World Wide Internet sind "TCP, IP und Co." heute nicht mehr wegzudenken. Das Internet zum Beispiel besteht aus Hunderten von verschiedenen Netzwerken, sogar verschienenen Netzwerktypen, die miteinander verbunden sind und von denen jeder einzelne Computer mit jedem anderen Informationen austauschen kann.

01.2 DIE SCHICHTENTEILUNG

Um die Komplexität von Netzwerkprotokollen möglichst auf ein Minimum zu reduzieren werden sie normalerweise in Schichten unterteilt. Jede Schicht hat dann ihre ganz spezielle Zuständigkeit.
Ein Vorteil der Unterteilung in Schichten ist zum Beispiel, dass die Benutzerprogramme nicht mit Details wie Hardwaresteuerung, Netzwerktyp, Medienart belastet werden.
Außerdem kann das Protokoll leichter verbessert und erneuert werden.

01.2.1 DAS OSI-MODELL

Das OSI Referenzmodell wurde 1984 auf das Problem der wachsenden Inkompatibilität von Netzwerken hin veröffentlicht.
Die Lösung, eine Einteilung der komplexen Netzwerkkommunikation in 7 kleinere separate Schichten, hatte einige Vorteile:

- Verringerung der Komplexität
- Spezialisierung von Entwicklern auf bestimmte Probleme innerhalb einer Schicht (ohne eine Änderung in den anderen Schichten)
- Schnellere technische Entwicklung

- Sicherung der Kompatibilität (=Interoperabilität) zwischen Netzwerkkomponenten verschiedener Hersteller

- Layer 1: Physical Layer

Die sogenannte physikalische Schicht oder Bitübertragungsschicht regelt die elektronischen (Spannungspegel, ...), mechanischen (Netzwerktyp, Medium, ...) und funktionalen (Zeitabläufe, maximale Übertragungsentfernungen, Datenraten) Anforderungen. Außerdem gibt sie Auskunft über die physikalische Netzwerktopologie (Ring, Stern, ...).
Stichworte für die 1. Schicht wären Signal und Übertragungsmedium.

- Layer 2: Data Link Layer

Die Hauptaufgabe der Sicherungsschicht ist das Erkennen und Beheben von Übertragungsfehlern.
Sie stellt als Dienst also die fehlerfreie Übertragung von Datenpaketen auf einen physikalisch an den Rechner angeschlossenen Zielrechner zur Verfügung. Außerdem übernimmt sie die Synchronisation der darunter liegenden Schicht und die Steuerung der Reihenfolge von Datenpaketen.
Außerdem beschäftigt sich die Sicherungsschicht mit der logischen Netzwerktopologie (Ethernet, Tokenring, ...), dem Zugang zum Netzwerk (network access), der Reihenfolge der Frames und der Datenfluss-Kontrolle.
Stichworte für die 2. Schicht wären Zugangskontrolle und Datenordnung.

- Layer 3: Network Layer

Die Netzwerkschicht oder Vermittlungsschicht gewährleistet Verbindungsfähigkeit und steuert den Weg des Datenpaketes vom Quell- zum Zielcomputer, die sich an geographisch verschiedenen Standorten befinden können (Routing).
Stichworte für die 3. Schicht wären Adressierung, Routing und Pfadauswahl.

- Layer 4: Transport Layer

In der Transportschicht findet die Segmentierung (=Aufteilung) bzw. der Wiederzusammenbau der Datagramme (oder Datenblöcke) in kleinere Pakete für die Netzwerkschicht statt.
Das Hauptziel ist die Entlastung der oberen Schichten von Transportdetails. Hier werden Verlässlichkeit, Fehlererkennung und -korrektur, und Verbindungstechnik behandelt.
Die Transportschicht ist die letzte der 4 unteren Schichten, die mit dem Datentransport an sich beschäftigt sind.
Stichworte für die 4. Schicht wären Verlässlichkeit und "Service-Qualität".

- Layer 5: Session Layer

Die "Sicherungsschicht" oder Sitzungsschicht ist verantwortlich für die Verwaltung von sogenannten Sitzungen zwischen zwei Hosts. Sie steuert den Datentransfer und synchronisiert Dialoge. Außerdem trifft sie Vorkehrungen für effizienten Datentransfer, "Class of Service" (Dienstgüte) und meldet Probleme der oberen Schichten.
Stichworte für die 5. Schicht wären Dialog und Konversation

- Layer 6: Presentation Layer

Die Darstellungsschicht sorgt dafür, dass die Informationen der Anwendungsschicht des Sender-Systems von der Empfänger-Anwendungsschicht verstanden werden. Falls notwendig, agiert die Darstellungsschicht als Übersetzer zwischen verschiedenen Datenformaten, um zu einer "gemeinsamen Sprache" zu kommen.

Ein Stichwort für die 6. Schicht wäre gleiches Datenformat.

✖ Layer 7: Application Layer
Die Anwendungsschicht ist dem Benutzer am nächsten. Sie gewährleistet die Netzwerkdienste für die Anwendungen "(z.B. Browser, Mailprogramme, ...), synchronisiert Prozeduren und sorgt für die Einigung auf Verfahren zur Fehlerbehebung und Steuerung der Datenintegrität.
Stichworte für die 7. Schicht wären Anwendungen, Browser.

01.2.2 DAS TCP/IP-MODELL

Obwohl das OSI-Modell ursprünglich als Nachfolger von TCP/IP gedacht war, bewährte sich die Protokollsuite bis heute.
In der TCP/IP Protokollsuite existieren nur 4 Schichten:

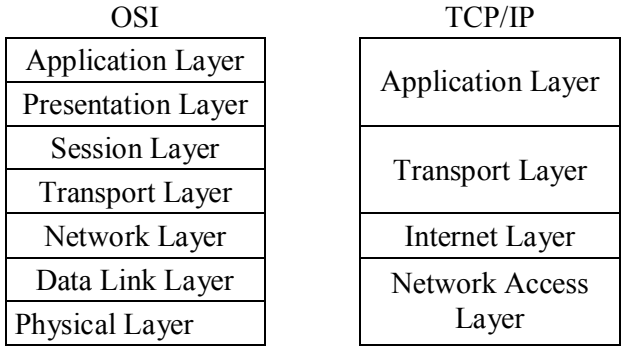
✖ Layer 1: Network Access Layer
Die sogenannte Netzzugriffsschicht ist eine Mischung von der Physikalischen Schicht und der Sicherungsschicht und beschäftigt sich mit den physikalischen Gegebenheiten.

✖ Layer 2: Internet Layer
Die Internetschicht sendet Pakete von der Quelle zum Ziel, unabhängig vom Weg und den Netzwerktypen. Das hier angesiedelte Protokoll IP steuert unter anderem die Wahl der besten Route.

✖ Layer 3: Transport Layer
Das Hauptziel ist (wie schon in der selben OSI-Schicht) die Entlastung der oberen Schichten von Transportdetails. Hier werden Segmentierung, Verlässlichkeit, Fehlererkennung und –Korrektur, und Verbindungstechnik behandelt.
Die Transportschicht ist die letzte der 4 unteren Schichten, die mit dem Datentransport an sich beschäftigt sind.
Stichworte für die 3. TCP/IP Schicht wären Verlässlichkeit und "Service-Qualität".
Das Protokoll TCP ist hier angesiedelt.

✖ Layer 4: Application Layer
Die Anwendungsschicht aus dem TCP/IP-Modell sollte nicht mit der des OSI-Modells verwechselt werden, da sie unterschiedliche Funktionen haben.
Diese 4. Schicht besteht im Grunde aus der Application- , Presentation- , und Session-Schicht des OSI-Modells. TCP verbindet alle anwendungsorientierten Probleme in dieser einen Schicht.
Stichworte für diese Schicht wären also Dialog, gleiches Datenformat und Anwendung.

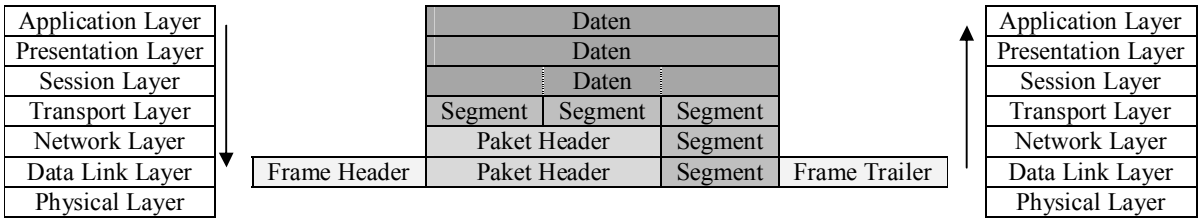
Das nachstehende Schema zeigt den Vergleich zwischen dem OSI-Modell und TCP/IP:



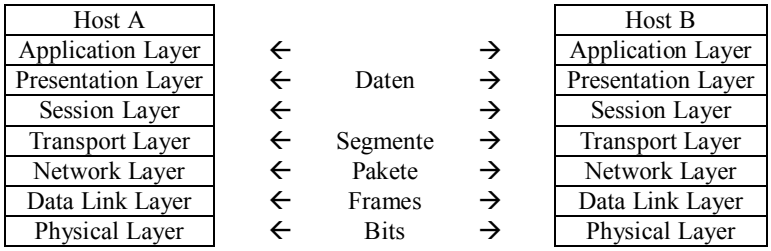
(Schema: "Vergleich OSI - TCP/IP")

01.3 ENCAPSULATION - DATENKAPSELUNG

Sollen Daten über ein Netzwerk von einem Host zu einem Anderen geschickt werden, dann müssen die Daten gegebenenfalls in kleinere Einheiten zerteilt und in Datenpakete gepackt werden. Außerdem werden die eigentlichen Daten beim Durchlaufen der Schichten noch mit Kopfdaten (Header), Endmarken (Trailer) und anderen Angaben versehen. Dieser Vorgang wird Kapselung genannt.
Jede Schicht tauscht über diese zusätzlichen Angaben mit ihrem Gegenüber zusätzliche Informationen, sogenannte PDUs (protocol data units) aus (Peer-to-Peer Communication).



(Schema "Datenkapselung")



(Schema "Peer-to-Peer")

01.3.1 VERLAUF DER KAPSELUNG

Die Kapselung wird anhand des oberen Schemas erklärt.
Nachdem die Daten von der Quelle gesendet wurden, durchlaufen sie von oben nach unten die einzelnen Schichten.
Wie in der Abbildung müssen Netzwerke die folgenden fünf Konvertierungsschritte durchlaufen, um Daten zu kapseln:

✖ Daten zusammenstellen
Wenn ein Benutzer eine E-Mail-Nachricht sendet, werden die alphanumerischen Zeichen dieser Nachricht von der Darstellungsschicht in Daten umgewandelt, die im Netzwerk übertragen werden können.

✖ Daten für den Transport zwischen den Endsystemen packen
Die Daten werden für den Transport im Netzwerk zu Datenpaketen gepackt. Mit Hilfe von Segmenten stellt die Transportschicht sicher, dass die Systeme an beiden Enden zuverlässig miteinander kommunizieren.

✖ Netzwerkadresse an den Header anhängen
Die Daten werden in ein Paket oder Datagramm gestellt, das einen Netzwerk-Header mit der logischen Adresse (IP-Adresse) des Absenders und des Empfängers enthält. Anhand der Adressen können Netzkomponenten wie z.B. Router die Datenpakete auf einem gewählten Pfad über das Netz senden.

✖ Lokale Adresse an den Header für die Sicherungsschicht anhängen
Jede Station am Netz muss das Paket in einen Frame kapseln. Der Frame ermöglicht die Verbindung zum nächsten direkt angeschlossenen Gerät auf diesem Pfad. Das Framing muss auf jedem Gerät im gewählten Netzwerkpfad durchgeführt werden, damit eine Verbindung zum nächsten Gerät hergestellt werden kann.

✖ Für die Übertragung in Bits umwandeln
Zur Übertragung über das Medium muss der Frame in ein binäres Muster umgewandelt werden. Anhand eines Taktsignals können die Geräte diese Bits bei der Übertragung über das Medium unterscheiden. Bei einem Übertragungsweg über verschiedene Netzwerkmedien werden einfach neue Header und Trailer hinzugefügt.

01.3.2 NAMEN DER DATEN IN DER JEWEILIGEN SCHICHT

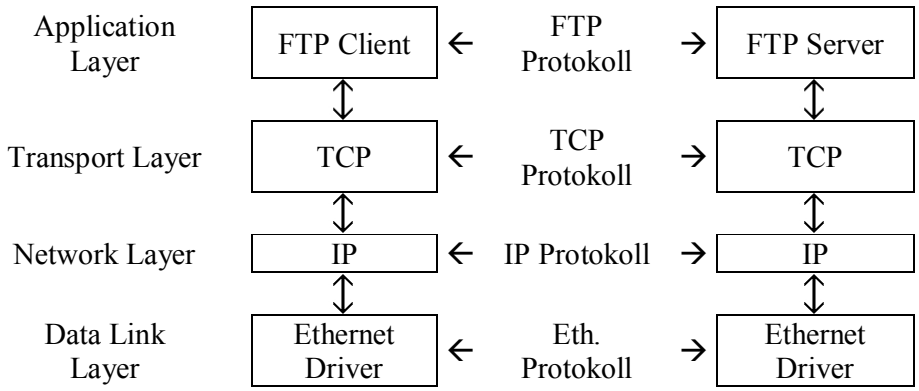
Application Layer	Daten
Presentation Layer	Daten
Session Layer	Daten
Transport Layer	Segmente
Network Layer	Pakete
Data Link Layer	Frames
Physical Layer	Bits

Datenpakete in einem Netz gehen von einer Quelle aus und werden an ein Ziel übertragen. Jede Schicht stützt sich auf die Dienstfunktion der unmittelbar darunter liegenden OSI-Schicht. Der Dienst wird bereitgestellt, indem die niedrigere Schicht die PDU der höheren Schicht mittels Kapselung in ihr Datenfeld stellt und die Header und Trailer hinzufügt, die die Schicht zur Ausführung ihrer Funktion benötigt. Während die Daten die Schichten des OSI-Modells durchlaufen, werden weitere Header und Trailer hinzugefügt. Nachdem die Schichten 7, 6 und 5 ihre Header hinzugefügt haben, fügt die Schicht 4 weitere Headerinformationen hinzu. Diese Datengruppe, die Schicht 4-PDU, wird als Segment bezeichnet.

Die Vermittlungsschicht stellt z.B. einen Dienst für die Transportschicht bereit, und die Transportschicht übergibt Daten an das Netzwerk-Subsystem. Die Aufgabe der Vermittlungsschicht besteht darin, die Daten durch das Netzwerk zu befördern. Zu diesem Zweck kapselt sie die Daten, fügt ihnen einen Header hinzu und erstellt auf diese Weise ein Paket (Schicht 3-PDU). Der Header enthält die zur Durchführung des Transports erforderlichen Daten wie z.B. logische Quell- und Zieladresse.

Die Sicherungsschicht stellt der Vermittlungsschicht einen Dienst bereit. Sie kapselt die Daten der Vermittlungsschicht in einem Frame (Schicht 2-PDU). Der Frame-Header enthält die für die Funktionen der Sicherungsschicht erforderlichen Daten (z.B. physikalische Adressen).

Die Bitübertragungsschicht stellt der Sicherungsschicht ebenfalls einen Dienst bereit. Die Bitübertragungsschicht kodiert den Frame der Sicherungsschicht für die Übertragung über das Medium in der Schicht 1 in ein Muster aus den Ziffern 1 und 0.



(Schema "FTP-Kommunikation")

Das hier dargestellte Schema zeigt, wie jede Schicht auf PC1 mit derselben auf PC2 kommuniziert. Gleichzeitig werden (hier bei PC1) nach unten hin die Daten zerstückelt, mit Kopfdaten und zusätzlichen Informationen versehen und erst auf der Netzwerkschicht übertragen. Auf PC2 geschieht das Gegenteil: Die Daten werden wieder verbunden, gegebenenfalls sortiert, "Header" und "Frames" entfernt und schließlich von der Anwendungsschicht gelesen.

Die Protokoll-Suite besteht aber nicht nur aus 4 Protokollen, sondern aus einer Vielzahl von Protokollen.

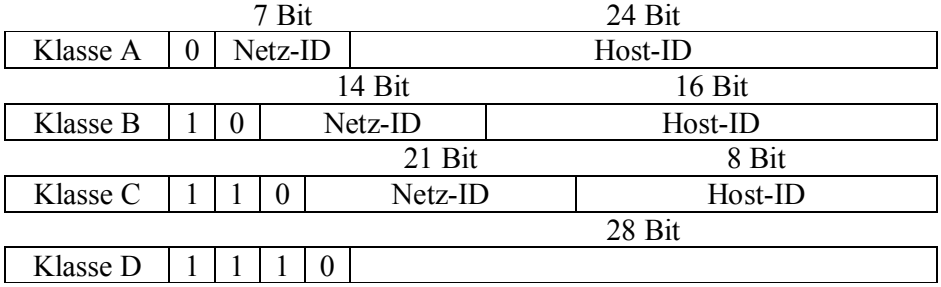
Die folgenden wichtigsten Protokolle werden in den anschließenden Kapiteln näher beschrieben: IP, TCP, UDP, Telnet, FTP
Weiters gibt es noch die weniger bekannten Protokolle ICMP, IGMP, ARP, RARP, NFS, SMTP, SNMP

01.4 IP-ADRESSEN (Schicht 2)

Für ein funktionierendes Netzwerk ist die eindeutige Bestimmung jedes Hosts mittels einer einzigartigen Identifikationsnummer unverzichtbar.
Die IP-Adressen erfüllen genau diesen Zweck. Sie werden aus 4 binären Oktetten gebildet, was eine Möglichkeit von $2^{32} = 4294967296$ Adressen ergibt.
Diese 32-Bit-Adressen werden normalerweise in 4 Dezimalzahlen (eine Zahl pro Byte) aufgeschrieben, die durch einen Punkt getrennt sind ("dotted decimal notation").
Ein Beispiel für eine solche Adresse ist 194.232.104.21 (www.orf.at) .

01.4.1 DIE NETZWERKKLASSEN (Schicht 2)

Es gibt zur Zeit 5 verschiedene Netzwerkklassen. Diese werden gebildet, indem man die ersten 1 – 5 Bit einer binären Adresse reserviert.
Ein Klasse A-Netzwerk beginnt zum Beispiel immer mit einem 0-Bit. Die nachfolgenden 7 Bits werden für die Netzwerk-ID gebraucht, und die letzten 24 für die Host-IDs.



	Bereich	mögl. Net-IDs	mögl. Host-IDs
Klasse A	0.0.0.0 – 127.255.255.255	128	16.777.216
Klasse B	128.0.0.0 – 191.255.255.255	16384	65536
Klasse C	192.0.0.0 – 223.255.255.255	2.097.172	256
Klasse D	224.0.0.0 – 239.255.255.255		
Klasse E	240.0.0.0 – 247.255.255.255		

Klasse A-Netzwerke werden nur von wenigen extrem großen Organisationen benötigt.
Klasse B-Netzwerke sind in einigen Universitäten und großen Unternehmen vorhanden.
Klasse C-Netzwerke umfassen nur 256 Hosts in jedem Netz. Die meisten Netzwerke, die mit dem Internet verbunden sind, gehören dieser Netzwerkklasse an.

Klasse D-Netzwerke werden nur für sogenanntes Multicasting verwendet. (siehe: Reservierte IP-Adressen).
Klasse E-Netzwerke wurden für "experimentelle Zwecke" reserviert.

01.4.2 SUBNETZE

Die Subnetz-Unterstützung erlaubt dem Internet-Anbieter (bzw. dem Netzadministrator), die hochwertigen Bits des vom ICANN (nichtkommerzielle Organisation zur Adressverteilung im Internet) zugewiesenen Adressbereiches mit Host-IDs für eine weitere interne Aufteilung in Subnetze zu nutzen.
Diese spalten den Teil der Host-ID noch einmal in Subnetz-ID und Host-ID auf.
Zusätzlich zur IP-Adresse muss ein Host also auch noch wissen, wie viele Bit für die Subnetz-ID und die Host-ID verwendet werden – diese Information wird mit der Subnetz-Maske übertragen.

	16 Bit		8 Bit	8 Bit	
Klasse B	Netz-ID		Subnetz-ID	Host-ID	
Subnetz-Maske	11111111	11111111	11111111	00000000	= 255.255.255.000

	16 Bit		10 Bit	6 Bit	
Klasse B	Netz-ID		Subnetz-ID	Host-ID	
Subnetz-Maske	11111111	11111111	11111111 11	000000	= 255.255.255.192

Die Unterteilung in Subnetze hat einige Vorteile.
Die interne Einteilung wird vor externen Routern sozusagen versteckt, sie können und müssen die Details nicht kennen und behandeln das Gesamte Netz als Einheit. Ein Firmeninternes Netz braucht also nur einen Router als Verbindung zum Internet.
Außerdem wird die Größe der sogenannten Routingtabellen, die später noch erklärt werden, drastisch vermindert, wenn man statt 30 eigenen Netzwerken 1 Netzwerk mit 30 Subnetzen verwendet.

01.4.3 RESERVIERTE IP-ADRESSEN

✖ Netzwerk-Adresse
Eine IP-Adresse mit der Host-ID 0 stellt die Netzwerkadresse dar. Diese Adresse spricht das gesamte Teilnetz an.

✖ Broadcast-Adresse
IP-Adressen, deren Host-ID nur aus 1en besteht sind Broadcast-Adressen. Mit dieser Adresse können alle Hosts eines Teilnetzes angesprochen werden.

✖ Multicast-Adresse
Die gesamte Netzwerkklasse D ist für Multicasting reserviert. Das ist eine spezielle Methode, Informationen von einem Server gleichzeitig an mehrere bestimmte Hosts zu übertragen, zum Beispiel für Videokonferenzen oder Internet-Fernsehübertragungen.
Der Anbieter muss vor der Übertragung seine Multicast-Adresse angeben, die dann vom Host aus kontaktiert wird.

Ein praktischer Vergleich ist das normale Rundfunkfernsehen: Ein Anbieter gibt seine Frequenz an, und das Fernsehgerät findet auf genau dieser Frequenz die Bild- und Toninformationen.
Diese spezielle Übertragungsart erfordert einen multicast-fähigen Router.

✖ Loopback-Adresse
Adressen, bei denen das erste Byte aus lauter 1en (Dezimal 127) besteht, sind sogenannte Loopback-Adressen. Pakete an solche Adressen werden nie ins Netzwerk geschickt, sondern immer lokal behandelt. Sinn einer solchen Adresse ist die Überprüfung der lokalen Netzwerkhardware.

✖ Außerdem sind bestimmte IP-Adressen für interne Netzwerke definiert:

Klasse A	10.0.0.0 – 10.255.255.255
Klasse B	172.16.0.0 – 172.31.255.255
Klasse C	192.168.0.0 – 192.168.255.255

Diese Adressen werden, als Quell- oder Zieladresse eingesetzt, von Routern im Internet nie weitergeleitet.

01.5 PORTS (Schicht 3)

Port-Nummern sind ein Teil der TCP-Kopfdaten und bestimmen, welcher Anwendungsprozess auf einem Host mit welchem Prozess auf einem anderen Host über welches Protokoll kommuniziert.
Dies ist notwendig, weil ein Server-Prozess durchaus mehrere gleichzeitige Verbindungen mit einem Host haben kann (z.B. bei DNS-Servern).
Über diese Port-Nummern erfolgt der gesamte Datenaustausch zwischen dem Protokoll und den Anwendungsprozessen.
Die Vergabe der Port-Nummern an Anwendungsprozesse geschieht dynamisch und wahlfrei. Für bestimmte, häufig benutzte Anwendungsprozesse sind jedoch beim Server feste Port-Nummern vergeben ("Assigned Numbers" oder "Well-Known Ports"). Der Sinn dieser fixen Zuteilung ist, dass gängige Dienste leichter zu finden sind.
Die Well-Known Ports reichen von 1 bis 1023 und sind in den meisten Fällen gerade, weil frühere Systeme jeweils eine Nummer zum Senden und eine zum Empfangen benötigten.
Seit einiger Zeit kann aber mit nur einem Port gesendet und empfangen werden. Ausnahmen sind aber zum Beispiel FTP (20, 21) und BOOTP (67, 68).
Zugeteilt werden die zugeteilten Nummern von IANA (Internet Assigned Numbers Authority).
Client-Hosts benötigen keine fixen Portnummern, weil der Verbindungsaufbau immer von ihnen ausgeht. Die bereitgestellten Portnummern für Clients werden "Ephemeral Ports" genannt, übersetzt bedeutet das flüchtig, kurzlebig. Diese Nummern liegen zwischen 1024 und 65535 und sind von der Anwendung frei wählbar.

01.6 DNS (Schicht 4)

Da man sich Wörter oder logische Buchstabenfolgen viel leichter merken kann als 12stellige Dezimalzahlen, gibt es im Internet einen speziellen Dienst: Domain Name System.
Computer, die diesen Dienst im Internet anbieten, übersetzen die sogenannten Domainnamen in IP-Adressen und umgekehrt.
Man kann also in einen Browser statt 194.232.104.21 auch einfach www.orf.at eingeben, und kommt ans selbe Ziel.
DNS wird später noch ausführlicher Erklärt.

02 IP (INTERNET PROTOKOLL) – Schicht 2

IP ist sozusagen das Fundament der TCP/IP-Prokoll-Suite. Jede Information von TCP, UDP, ICMP und IGMP wird als sogenanntes IP-Datagramm verpackt und verschickt.
Das Internetprotokoll selbst ist unverlässlich, es gibt keine Garantie dafür, dass ein Paket tatsächlich das Ziel erreicht.
Außerdem setzt es keine ständige Verbindung zwischen Quell- und Zielcomputer voraus, in der Informationen zur Übertragung (z.B.: Meldung über Erhalt eines Paketes) ausgetauscht werden. Wenn zwei Datenpakete von PC1 nach PC2 geschickt werden, wird jedes separat behandelt. Das kann bedeuten, dass die zweite Einheit mitunter über einen kürzeren Weg geschickt wird als die Erste.
Es liegt also an den höheren Schichten, für die Vollständigkeit und Reihenfolge der Daten zu sorgen.
Geht irgend etwas in der Übertragung schief, hat IP ein simples Prinzip: "Erhaltene Daten verwerfen und eine Fehlermeldung zurückschicken".

02.1 DER IP–HEADER

Bits

0 – 4	4 – 8	8 – 12	12 – 16	16 – 20	20 – 24	24 – 28	28 – 31
Version	Headerlänge	TOS	Gesamtlänge				
Identifikation				Flags	Fragment-Offset		
TTL		Protokoll		Header Prüfsumme			
IP-Adresse des Quellcomputers							
IP-Adresse des Zielcomputers							
Optionen						Füllzeichen	
Daten							

Die normale Header-Gesamtlänge beträgt 20 Byte, wenn keine Optionen festgelegt sind.
Die Übertragung erfolgt in der Reihenfolge der Felder in der Skizze, jeweils in 8-Bit-Paketen. Diese Reihung ist bekannt als "Big Endian Byte Ordering" und wird für praktisch alle Daten verwendet, die über ein Netzwerk geschickt werden.
Maschinen, die andere Formate verwenden, müssen Ihre Headerwerte übersetzen, bevor Daten verschickt werden können.

✖ Das erste Feld des Headers ist das Versionsfeld, ein 4 Bit langes Feld zur Kennzeichnung der Protokollversion.

✖ Die Headerlänge dient zur Kontrolle der Daten. Sie gibt die Länge des Headers in 32-Bit-Worten an. Dieses Feld ist erforderlich, weil das Options-Feld keine fixe Größe hat.

✖ Der Service-Typ (type of service, TOS) besteht eigentlich aus 2 Bereichen:
Die ersten 3 Bit werden heutzutage ignoriert. Sie wurden früher für Prioritätsangaben verwendet.
Die nächsten 4 Bit sind für die folgenden Optionen:

- ✖ TOS Minimum-Delay wird für Dienste benutzt, bei denen es wichtig ist, dass Pakete möglichst ohne Zeitverzögerung weitergeleitet werden (z.B.: FTP-Befehle, Telnet).

- ✖ TOS Maximum-Troughput wird für Dienste benutzt, bei denen es wichtig ist, dass große Datenmengen mit hoher Geschwindigkeit weitergeleitet werden (z.B.: FTP-Daten, HTTP).

- ✖ TOS Maximum-Reliability wird benutzt, wenn es wichtig ist, das man eine gewisse Sicherheit hat, dass die Daten an ihr Ziel gelangen, ohne das ein erneutes Senden nötig ist (z.B.: SNMP, DNS).

- ✖ TOS Minimum-Cost wird benutzt, wenn es wichtig ist die Kosten der Datenübertragung zu Minimieren (z.B.: NNTP, SMTP).

✖ Weil das Feld für die Gesamtlänge auf 16 Bit begrenzt ist, kann die maximale Länge 64 Kilobyte betragen. Dieses Feld dient zur Überprüfung der Vollständigkeit des gesamten Pakets.

✖ Das 16-Bit lange Identifikationsfeld dient der eindeutigen Identifikation einzelner Datenpakete. So kann auf korrekte Reihenfolge und Vollständigkeit geprüft werden.

✖ Das Flag-Feld gibt mit seinen 3 Bit an, ob Daten fragmentiert (zerstückelt) sind, und das darauf folgende Fragment-Offset-Feld gibt die genaue Position der Fragmentdaten in einem Datenblock an. Durch die festgelegte Größe des Offset-Feldes von 13 Bit ist die Anzahl der Fragmente in einem Datagramm auf 8.192 begrenzt.

✖ Um endlose Paketschleifen und somit unnötige Netzwerkbelastung zu vermeiden, wurde das TTL-Feld entwickelt. Es gibt die maximale Anzahl von Routern an, die ein Datagramm passieren kann. Diese Zahl wird von jedem Router um 1 herabgesetzt. Beträgt der Wert 0, wird das Datagramm verworfen und eine ICMP-Fehlermeldung an die Quelladresse geschickt. Der Maximalwert beträgt 255 Sekunden, das entspricht dem Durchlaufen von 255 Routern.

✖ Das folgende Protokoll-Feld bestimmt, an welches Protokoll der nächsthöheren Schicht die Daten weitergegeben werden sollen.

✖ Im Header-Prüfsummen-Feld wird der Header auf Fehler untersucht. Hier wird ein Teil der zu übertragenden Information mit einem komplizierten mathematischen Algorithmus versehen und beim Empfänger kontrolliert.

✖ Die Adresse des Absenders sowie die Zieladresse wird im jeweils zugehörigen 32-Bit-Feldern als Hexadezimal-Wert angegeben.

✖ Das Optionsfeld dient dazu, weitere Informationen für die höheren Protokolle zu liefern. So können in diesem Feld beispielsweise bestimmte Sicherheitsanforderungen festgelegt werden, die die Empfangseite erfüllen muss. Weitere Beispiele wären der "Record Route"-Befehl, bei dem jeder Router seine IP-Adresse hinzufügt, oder "Loose Source Routing" (das Datagramm muss die angegebenen IP-Adressen passieren) bzw. "Strict Source Routing" (das Datagramm darf nur die angegebenen IP-Adressen passieren).
Das Optionsfeld hat eine variable Länge und wird durch das Füllzeichen-Feld immer auf 32 Bit aufgefüllt.

02.2 IP FRAGMENTIERUNG

Jedes Übertragungsmedium besitzt seinen eigenen Wert für MTU (maximum transfer unit), also die maximale Länge eines IP-Pakets.
Sobald die IP-Schicht ein IP-Datagramm schicken soll, vergleicht sie zuerst die MTU der Schnittstelle. Ist das Datagramm größer als die Maximale Übertragungsgröße, teilt IP das Paket in mehrere Fragmente auf und versendet sie einzeln.
Datagramme können sowohl vom Quellhost als auch von jedem dazwischenliegenden Router fragmentiert werden, aber sie werden erst am Ziel wieder zusammengebaut.

Für die korrekte Bearbeitung dieser Fragmente müssen die folgenden Bedingungen für die Felder im IP-Header beachtet werden:

✖ Das Identifikationsfeld besitzt normalerweise eine eindeutige Identifikationsnummer, individuell für jedes IP-Datagramm. Bei Fragmentierung muss jedes Fragment die Nummer des ursprünglichen Paketes erhalten.

✖ Das Flag-Feld verwendet eines seiner 3 Bits um zu sagen "Es kommen noch mehr Fragmente". Dieses Bit ist bei allen Fragmenten – außer beim letzten – Fragmenten aktiviert.

✖ Das Fragment-Offset-Feld gibt die genaue Position der Fragmentdaten im Datenblock an. Weil jedes Fragment einen individuellen Weg durchlaufen kann, kann durch verschiedene Geschwindigkeiten die Reihenfolge verändert werden. Das Fragment-Offset-Feld stellt die korrekte Reihenfolge der Fragmente wieder her.

✖ Die Gesamtlänge wird bei der Fragmentierung für jedes neue Fragment neu berechnet und geändert.

Einer der Nachteile des Fragmentierungsvorganges ist, dass bei Verlust eines einzelnen Fragments das gesamte Datagramm neu übertragen werden muss. IP-Datagramme besitzen selbst kein Timeout-Feld, und da die Zerstückelung auch von einem Router durchgeführt werden kann, kann der Quellhost nicht wissen, welches Datagramm wie zerstückelt wurde.

02.3 IP ROUTING

Sind 2 Computer direkt miteinander verbunden oder am gleichen Netzwerk, werden die Daten einfach direkt von Host zu Host übertragen.
IP Routing hingegen wird eingesetzt, wenn 2 Computer physikalisch nicht miteinander verbunden sind (zum Beispiel am selben Kabel / Hub).

IP Routing funktioniert folgendermaßen:

Die IP-Schicht besitzt sogenannte Routingtabellen, die folgende Informationen enthalten:

Zieladresse	Next-Hop-Router	Flags	Schnittstelle

- Zieladresse des Pakets — Entweder eine Hostadresse oder eine Netzwerkadresse (Host-ID 0).
- IP-Adresse des sogenannten "Next-Hop-Routers" — Ein Next-Hop-Router ist ein Router auf einer direkten Verbindung. Dieser sendet die erhaltenen Pakete an den nächsten Router / die Zieladresse weiter.
- Flags — Ein Flag gibt an, ob die Zieladresse einen einzelnen Host oder ein ganzes Netzwerk bestimmt.
- Schnittstellenangabe — Router sind immer durch mehrere Schnittstellen mit mehreren Netzwerken verbunden. Daher muss bestimmt werden, an welchen Anschluss weitergeleitet wird.

Wenn ein Datenpaket an IP weitergegeben wird (egal ob von darüber liegenden Schichten oder von der darunter liegenden Netzwerkschicht), werden zuerst diese Routingtabellen durchsucht.
Ist die Zieladresse die Eigene, dann wird das Paket zum angegebenen Protokoll in die nächsthöhere Schicht weitergeleitet. Entspricht die Zieladresse weder der eigenen Adresse, noch einer Broadcast-Adresse, dann wird die Routingtabelle nach der genauen Zieladresse (Netz- und Host-ID) durchsucht. Gibt es einen passenden Eintrag in der Tabelle, wird das Paket weitergeleitet. Falls nicht, wird nur die Netzwerk-Adresse überprüft und gegebenenfalls an den Router aus der Tabelle geschickt.
Findet sich überhaupt keine Übereinstimmung zwischen Zieladresse und Routingtabelle, werden die Daten an einen voreingestellten Router, den "Default Gateway" weitergeleitet.
Falls auch kein Default Gateway definiert ist, ist das Paket unzustellbar, und eine entsprechende Fehlernachricht wird an die Quelladresse gesendet.

02.4 DIE ZUKUNFT VON IP: IPv6

Die derzeitige Version IPv4 erlaubt etwa 4 Milliarden Adressen, und nach groben Schätzungen werden diese Adressen gerade noch bis ins Jahr 2015 ausreichen.
Um diesem Problem früher auszuweichen als dem Y2K-Bug, haben Internet-Sites IPv6 bereits großflächig implementiert.
IPv4 bietet mit dem Adress-Spektrum von 000.000.000.000 bis 255.255.255.255 also etwa 4.000.000.000 Adressen.
IPv6 besteht aus acht Gruppen von 16-Bit-Zahlen (also 128 Bit), die jeweils durch einen Doppelpunkt (nicht mehr durch den einfachen Punkt) getrennt werden. Diese Zahlen werden nun nicht mehr als Dezimalzahlen angezeigt, sondern als Hexadezimalgruppen.

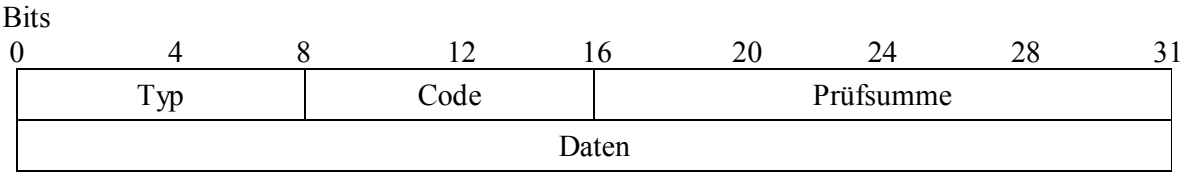
Das Spektrum reicht hier also von 0000:0000:0000:0000:0000:0000:0000:0000 bis FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF – daraus ergibt sich die astronomische Adressenvielfalt von 340.282.366.920.938.463.463.374.607.431.768.211.456 möglichen Adressen (39 Stellen!). Da diese Zahl von Normalsterblichen kaum zu fassen ist, haben sich findige Rechenkünstler einen nicht minder beeindruckenden Vergleich ausgedacht: Die Zahl reicht aus, um jeden Quadratmeter der Erdoberfläche mit 665.570.793.348.866.943 Adressen abzudecken (immer noch 18 Stellen).
Die IPv4-Adressen sind im IPv6-Adressraum beinhaltet. Die Umwandlung erfolgt einfach, indem der Anfang der Adresse mit Nullen aufgefüllt wird.
Die derzeitige IP von www.orf.at, 194.232.104.21, wird also zu folgender IPv6-Adresse:
0:0:0:0:0:0:85C5:EDF5.

Neben der Adressenvielfalt bietet IPv6 noch einige andere Vorteile, zum Beispiel verbesserte Sicherheitsfunktionen, ein vereinfachtes Headerformat, schnellere Multimedia-Kommunikation, ...

03 ICMP (Internet Control Message Protocol) – Schicht 2

ICPM dient der Steuerung und Verwaltung in Netzwerken. Es ist, wie IP, ein Layer 2 Protokoll, und ermöglicht die Kommunikation zwischen den IP-Schichten von Rechnern. Obwohl IP und ICMP auf der selben Schicht liegen, benutzt ICMP IP für das Versenden seiner Daten. Die gesamte ICMP-Nachricht steht also im Datenfeld des IP-Datagramms.
Im Großen und Ganzen wird ICMP verwendet, um Fehlermeldungen direkt zwischen den Endsystemen auszutauschen – Rechner, die nur auf der Route vom Sender zum Empfänger liegen, werten diese nicht aus.

03.1 DER ICMP-HEADER



✖ Das Typenfeld bezeichnet die Art der Nachricht. Beispiele für solche Typen wären:

0	Echo-Antwort
3	Ziel unerreichbar
4	"Source Quench"
8	Echo-Anforderung
11	TTL überschritten
30	Traceroute
33	"IPv6 Where-Are-You"
34	"IPv6 I-Am-Here"

✖ Der Fehlercode für das Datagramm in einer ICMP-Nachricht liegt im Feld "Code". Die Interpretation dieses Feldes hängt vom Nachrichtentyp ab.

✖ Das Prüfsummenfeld dient zur Überprüfung der Vollständigkeit des gesamten Pakets.

✖ Im Datenfeld steht ein Teil des betreffenden fehlerhaften IP-Datagramms.

03.2 ICMP-MELDUNGEN

Die häufigsten ICMP-Fehlermeldungen werden jetzt kurz erklärt:

03.2.1 ECHO (0, 8)

Der am häufigsten gebrauchte ICMP-Dienst ist das Testen der Erreichbarkeit eines Zielhosts und die Abfrage seines Zustandes.
Jeder Host kann jederzeit ein Echo-Paket an einen Zielcomputer aussenden, und dieser antwortet mit allen im Echo-Paket enthaltenen Daten.

03.2.2 ZIEL UNERREICHBAR (3)

Kann ein IP-Paket nicht weitergeleitet werden, dann wird eine entsprechende Fehlermeldung erzeugt. Einige mögliche Ursachen und deren Codes für den ICMP-Header sind:

0	Netz nicht erreichbar
1	Zielhost nicht erreichbar
2	Protokoll nicht erreichbar
3	Port nicht erreichbar
6	Zielnetz unbekannt
7	Zielhost unbekannt
12	Rechner für diesen Dienst nicht erreichbar

Als Beispiel für ICMP-Pakete wird hier nochmals die gesamte gesendete ICMP-Nachricht angezeigt:

Bits:	0 – 8	8 – 16	16 – 24	24 – 31
Header	Typ = 3	Code = 1	Prüfsumme	
Daten	unbenutzt (0)			
	IP-Header + 64 Bits vom betreffenden Datagramm			

03.2.3 SOURCE QUENCH (4)

Empfängt ein Router Pakete schneller, als er sie verarbeiten kann, liegt eine Pufferüberlastung vor. Die Ursache kann entweder ein einzelner Rechner sein, wenn er Pakete zu schnell erzeugt, oder die Summe des Einzelverkehrs vieler Rechner.
Wenn der Speicher eines Routers voll ist, müssen die ankommenden Pakete gelöscht werden, gleichzeitig wird eine ICMP-Nachricht an den Quellcomputer gesendet, welches um Verringerung der gesendeten Datenmenge bittet.

03.2.4 TTL ÜBERSCHRITTEN (11)

Wird die maximale Anzahl von Systemen, die ein IP-Paket passieren darf, überschritten, dann wird das Paket verworfen und die entsprechende ICMP-Fehlermeldung zurückgeschickt.

Hier gibt es 2 Codes:

0 Lebenszeit des IP-Pakets wurde überschritten. Hier wurde die maximale "Hop"-Anzahl erreicht. Nur ein Router kann diese Nachricht absenden.

1 Wartezeit für Rekonstruktion von IP-Fragmenten überschritten. Diese Fehlermeldung kann nur vom Zielhost versandt werden, wenn ein bestimmtes Fragment aus einem Datagramm noch immer ausständig ist.

03.3 ICMP-ANWENDUNGEN

03.3.1 PING

Die einfachste aller TCP/IP Anwendungen ist das Programm "Ping". Seinen Namen hat das Programm zum einen von der Unterwasser-Distanzmessung mithilfe von Schall, zum anderen ist es eine Abkürzung für "Packet InterNet Groper".
Das Programm "Ping" benutzt den Mechanismus von "Echo" und "Echo Reply" um zu prüfen, ob ein Zielhost erreichbar ist.

Die Syntax lautet in Windows 98:

ping [IP-Adresse] [-Optionen]

Folgende Optionen sind möglich:

-t	Sendet fortlaufend Ping-Signale zum angegebenen Host
-a	Wertet Adressen in Hostnamen aus
-n Anzahl	Sendet die gegebene Anzahl von Echo-Anforderungen
-l Länge	Pufferlänge senden
-f	Flag für "Don't Fragment" setzen
-i TTL	Time To Live festlegen (Standartwert =128)
-v TOS	Type Of Service festlegen
-r Anzahl	Route für [Anzahl] Abschnitte aufzeichnen
-s Anzahl	Zeiteintrag für [Anzahl] Abschnitte aufzeichnen
-w Timeout	Timeout in Millisekunden für eine Antwort

Ping ist in praktisch allen Betriebssystemen enthalten.

03.3.2 TRACEROUTE

Das Programm "Traceroute" erlaubt die Bestimmung der Route eines IP-Pakets zwischen Endsystemen.
Es sendet ein IP-Datagramm mit einer TTL von 1 zum Zielhost. Der erste Router ändert diesen Wert auf 0, verwirft das Datagramm und antwortet mit der ICMP-Meldung "TTL überschritten". Also ist der erste Router jetzt bekannt. Jetzt sendet das Programm wieder ein IP-Datagramm mit dem Wert für TTL = 2, und erhält so die Adresse des zweiten Routers.
Dieser Vorgang wird solange weitergeführt, bis das Paket tatsächlich am Ziel ankommt.

Der Befehl lautet in Windows 98:

tracert [IP-Adresse bzw. DNS-Adresse]

Auch Traceroute ist in allen Betriebssystemen vorhanden.

03.4 IGMP (Internet Group Management Protocol) – Schicht 2

IGPM wird hier nur der Vollständigkeit halber kurz angeführt.
Es ist als Hilfsprotokoll auf der Vermittlungsschicht angesiedelt und unterstützt die Gruppenkommunikation. Das IGMP-Protokoll wird für IP-Multicasting, also für das Rundsenden an mehrere Interfaces eingesetzt.

04 ARP (Adress Resolution Protocol) – Schicht 2

Das ARP-Protokoll löst IP-Adressen in MAC-Adressen auf.
Jede Ethernet-Karte besitzt eine eindeutige Hardware-Adresse, die auch als MAC-Adresse bezeichnet wird. Jedem Host im Netz wurde aber nun auch noch eine IP-Adresse zugeordnet und damit hat das Ethernet ein Problem: Mit der IP-Adresse 192.168.23.1 weiß es nichts anzufangen, mit der MAC-Adresse 08:00:00:04:72:98 allerdings schon.
Ein Übersetzer muss her: Das Adress Resolution Protocol (ARP) übersetzt die IP-Adressen in Ethernet-Adressen.
Es werden zu diesem Zweck Mapping-Tabellen angelegt, die die MAC-Adressen den Netzwerkadressen zuordnen. Vor dem Verbindungsaufbau über das Ethernet fragt IP bei ARP nach der Ethernet-Adresse der zugehörigen Ziel-Internet-Adresse an. ARP vergleicht seine Adresstabellen (oder ARP-Tabellen) mit der Anfrage.
Hat ARP keinen Eintrag in seiner Tabelle, so wird über eine Anfrage an alle Hosts (Broadcast) die Ethernet-Adresse der zugehörigen Internet-Adresse erfragt. Nur Schnittstellen mit einem Eintrag zu dieser IP-Adresse antworten auf die Anfrage. Die Antwort auf den ARP-Broadcast wird in der ARP-Adresstabelle gespeichert.

04.1 DER ARP-HEADER

Bits

0	4	8
Hardware Address Space		
Protocol Address Space		
Length of Hardware Address (n Bytes)	Length of Protocol Address (m Bytes)	
Operation code		
Hardware Address of Sender (n Bytes lang)		
Protocol Address of Sender		
Hardware Address of Target (m Bytes lang)		
Protocol Address of Target		

- Hardware Address Space:
Netzwerk-Typ, in dem das Datagramm generiert wurde (z.B. Ethernet: 1, IEEE 802 Netzwerke: 6)

- Protocol Address Space:
Protokoll-Typ, von dem die Operation angefordert wurde (z.B. IP, ARP, ...)

- Length of Hardware Address:
Länge der physikalischen Adresse in Byte (z.B. MAC-Adresse: 6)

- Length of Protocol Address:
Länge der Netzwerk-Adresse (z.B. IP-Adresse: 4)

✖ Operation Code:
Art der Operation
1 = ARP-Anforderung
2 = ARP-Antwort
3 = RARP-Anforderung
4 = RARP-Antwort

✖ Hardware Address of Sender:
Physikalische Adresse des Absenders (MAC-Adresse)

✖ Protocol Address of Sender:
Netzwerk-Adresse des Absenders (z.B. IP-Adresse)

✖ Hardware Address of Target
Physikalische Adresse des Empfängers

✖ Protocol Address of Target:
Netzwerk-Adresse des Empfängers

Will also Host A ein IP-Datagramm an Host B senden, sieht er zuerst in seinem ARP-Cache nach, ob die MAC-Adresse bekannt ist. Falls nicht, wird ein ARP-Anforderungs-Broadcast an alle Netzwerkschnittstellen gesendet. Sobald Host B seine IP-Adresse in der Anforderung erkennt, schickt er ein Antwort-Datagramm mit der eigenen MAC-Adresse. Außerdem trägt er IP- und MAC-Adresse von Host A in seinen ARP-Cache ein.
Host A erhält die Antwort, trägt IP- und MAC-Adresse von Host B in seinen ARP-Cache ein und kann nun alle folgenden IP-Datagramme an die richtige MAC-Zieladresse schicken.
Durch den ARP-Broadcast besitzen alle Hosts in diesem Netzwerk nun IP- und MAC-Adresse von Host A.

04.2 DIE BEFEHLSZEILE

Basierend auf dem Protokoll ARP gibt es bei allen Betriebssystemen ein Zeilenkommando, das ebenfalls arp heißt.

Unter Windows 98 sieht die Syntax folgendermaßen aus:

arp -s [IP-Adresse] Eth_Adr [Schnittst]
arp -d [IP-Adresse] [Schnittst]
arp -a [IP-Adresse] [-N Schnittst]

-a	Zeigt aktuelle ARP-Einträge durch Abfrage der Protokolldaten an. Falls IP_Adr angegeben wurde, werden die IP- und physikalische Adresse für den angegebenen Computer angezeigt. Wenn mehr als eine Netzwerkschnittstelle ARP verwendet, werden die Einträge für jede ARP-Tabelle angezeigt.
-g	Gleiche Funktion wie -a.
-N Schnittst	Zeigt die ARP-Einträge für die angegebene Netzwerkschnittstelle an.
-d	Löscht den durch IP_Adr angegebenen Hosteintrag. IP_Adr kann mit dem '*'-Platzhalter versehen werden, um alle Hosts zu löschen.

-s	Fügt einen Hosteintrag hinzu und ordnet die Internetadresse der physikalischen Adresse zu. Die physikalische Adresse wird durch 6 hexadezimale, durch Bindestrich getrennte Bytes angegeben. Der Eintrag ist permanent.
Eth_Adr	Gibt eine physikalische Adresse (Ethernetadresse) an.
Schnittst	Gibt, falls vorhanden, die Internetadresse der Schnittstelle an, deren Übersetzungstabelle geändert werden soll. Sonst wird die erste geeignete Schnittstelle verwendet.

04.3 RARP

Ergänzt wird ARP durch ein weiteres Protokoll, das Reverse Address Resolution Protocol (RARP). Dieses Protokoll stellt das Gegenstück zu ARP dar und ermöglicht es, zu einer bekannten physikalischen Adresse (MAC-Adresse) die zugehörige Netzwerkadresse (IP-Adresse) zu finden.
Hierbei benötigt man allerdings einen entsprechenden Server, der die gesuchten Adressinformationen bereithält.

05 UDP (User Datagramm Protocol) - Schicht 3

UDP baut direkt auf dem darunterliegenden IP-Protokoll auf und ist ein unzuverlässiges, verbindungsloses Protokoll. Unzuverlässig bedeutet, dass dieses Protokoll keinerlei Mechanismen enthält, die sichern, dass die Daten auch tatsächlich beim Zielrechner ankommen. Für solche Sicherheit - sprich Erhaltsbestätigung, Prüfung der Reihenfolge, ... - müssen also die oberen Schichten sorgen.

UDP bietet gegenüber TCP den Vorteil eines geringen Protokoll-Overheads (=Kopfdaten), und wird daher oft bei Anwendungen mit wenig Datentransfer verwendet, weil unter Umständen der Aufwand zur Herstellung einer Verbindung und einer zuverlässigen Datenübermittlung größer ist als die wiederholte Übertragung der Daten.

05.1 DER UDP-HEADER

Bits

0	4	8	12	16	20	24	28	31
Quell-Portnummer				Ziel-Portnummer				
Länge				Prüfsumme				
Daten								

- Die Quellportnummer bezeichnet die Portnummer der Anwenderschichtprotokolle, von dem die UDP-Message abgeschickt wurde.

- Die Zielportnummer gibt die Portnummer des Empfängerprotokolls auf der Anwendungsschicht an.

- Die Länge gibt Auskunft über die Länge der gesamten UDP-Message in Bytes. Bei UDP ist diese Angabe optional.

- Da die UDP-Prüfsumme (anders als die IP-Prüfsumme) auch den Datenbereich des Paketes abdeckt, können mit ihr kleinere Übertragungsfehler bemerkt werden.

- Außerdem wird vor der tatsächlichen Berechnung der Prüfsumme dem Datenpaket ein zusätzlicher Header - ein sogenannter Pseudo-Header - angefügt. Dieser wird nicht übertragen und bei der Länge nicht eingerechnet.

Der Pseudo-Header besteht aus sechs 16-bit Worten.

Bits

0	4	8	12	16	20	24	28	31
Quell-IP-Adresse								
Ziel-IP-Adresse								
Null			Protokolltyp		Gesamtlänge			

Zunächst zwei 32-bit-IP-Adressen von Sender und Empfänger, dann ein Nullbyte (für spätere Erweiterungen) gefolgt von einem Byte für die Nummer des Protokolltyps und 16 Bit für die Länge des gesamten UDP-Paketes.
Diese Daten müssen zum Testen der Prüfsumme aus dem IP-Header ausgelesen und in Form des Pseudo-Headers zum UDP-Paket hinzugefügt werden.

05.2 ÜBERLÄNGE

Die maximale UDP-Datagramm-Größe beträgt 65535 Bytes (inklusive den 20 Bytes des IP-Headers und den 8 Bytes des UDP-Headers). Trotzdem sind heute die meisten Systeme auf ein Maximum von 8192 Bytes voreingestellt. Einige UDP-Anwendungen beschränken sich sogar auf 512 Bytes oder weniger (z.B. DNS, TFTP, ...)

Normalerweise tauschen kommunizierende UDP-Anwendungen ihre maximale Datengröße aus, falls jedoch größere Pakete als erlaubt eintreffen, gibt es verschiedene Reaktionen:

- Die traditionelle Berkeley-Version schneidet das Datagramm sofort nach ihrem Paketmaximum ab und verwirft den Rest. Versionen ab 4.3BSD Reno informieren die Anwendung über die "Beschneidung".

- API-Sockets unter SVR4 beschneiden das Datagramm nicht, sondern geben den Überschuss in mehrfachen Lesevorgängen aus. Die Anwendung wird nicht über die Überlänge informiert.

- TLI API setzt ein Flag, das anzeigt, dass noch mehr Daten verfügbar sind. Die Anwendung muss durch mehrfaches Lesen die restlichen Daten auslesen.

Im nachfolgenden Kapitel wird beschrieben, dass durch den kontinuierlichen Datenstrom von TCP keine Daten durch Überlänge verloren gehen können.

06 TCP (Transmission Control Protocol) – Schicht 3

Sowohl TCP und UDP verwenden die IP-Protokollschicht, um Daten zu versenden, und trotzdem sind sie völlig verschieden.

TCP bietet verlässlichen Datentransfer durch die Verwendung eines Mechanismus, der Datenpakete / Segmente solange an einen Empfänger schickt, bis dieser mit einer Bestätigung des Empfangs antwortet.
Dabei überprüft der Empfänger auch wieder eine Prüfsumme (wie schon in der IP-Schicht), und erst wenn diese richtig ist, schickt er das Signal des Empfangs.
Auch die Fehlerkorrektur und Flusskontrolle stellen einen Vorteil gegenüber UDP dar.

Anders als bei UDP bestimmt nicht die darüber liegende Anwendung die Größe der Segmente, sondern TCP wählt selbständig die jeweils ideale Größe aus.

TCP ist verbindungsorientiert, das heißt, dass dieses Protokoll nicht einfach Daten losschickt wie UDP, sondern zuerst einen sogenannten Handshake durchführt, um sich mit dem Empfänger über dessen Bereitschaft zu synchronisieren.
Das Prinzip ist einfach, eine TCP Übertragung beginnt immer erst mit der Nachfrage ob der Empfänger bereit ist (SYN). Der Empfänger sendet ein entsprechendes Signal (SYN.ACK) und erst nachdem dieses Signal erhalten wurde startet TCP die Übertragung der Segmente.
Diese SYN.ACK-Nachricht beinhaltet außerdem die maximale Anzahl der Bytes, die in den lokalen Puffer passen. Dadurch wird ein Überlaufen des Puffers ("buffer overflow") verhindert.

Weiters ist TCP streamorientiert. Es sieht seine Daten als kontinuierlichen Datenstrom an.
Durch die Verwendung von Sequenznummern kann das empfangende TCP die Segmente wieder in richtiger Reihenfolge zusammenbauen und sie wieder zu einem Datenstrom formieren.

06.1 DER TCP-HEADER

Bits

<table>
<tr><td colspan="16">0 4 8 12 16 20 24 28 31</td></tr>
<tr><td colspan="8">Absender Portnummer</td><td colspan="8">Empfänger Portnummer</td></tr>
<tr><td colspan="16">Sequenznummer</td></tr>
<tr><td colspan="16">Bestätigungsnummer</td></tr>
<tr><td colspan="4">Offset</td><td colspan="4">Reserviert</td><td colspan="4">Flags</td><td colspan="8">Fenster</td></tr>
<tr><td colspan="8">Prüfsumme</td><td colspan="8">Dringlichkeitszeiger</td></tr>
<tr><td colspan="14">Optionen</td><td colspan="2">Füllzeichen</td></tr>
<tr><td colspan="16">Daten</td></tr>
</table>

✖ Die Absender Portnummer bezeichnet die Portnummer des Anwenderschichtprotokolls, von dem der Datenstrom abgeschickt wurde.

✖ Das Feld mit der Empfänger Portnummer bezeichnet die Portnummer des Empfängerprotokolls auf der Anwendungsschicht.

✖ Die Sequenznummer gibt die Position im Datenstrom an, an der das erste Byte dieses Segments eingefügt werden soll.

✖ Die Bestätigungsnummer dient zur Bestätigung des Empfangs eines Segments. Dieses Feld enthält immer die Nummer, die im nächsten Segment als Sequenznummer stehen soll.

✖ Das Daten-Offset-Feld nennt die Größe des Headers in 32 Bit-Worten. Ohne Optionen ist der Wert 5. Damit kann genau bestimmt werden, wo der Datenbereich des Segments beginnt.

✖ Im reservierten Feld sind 6 Bits für zukünftige Verwendung reserviert und müssen bislang auf 0 gesetzt sein.

✖ Das Feld der Flags enthält verschiedene Informationen zur Kommunikationssteuerung.
URG bedeutet, dass der Dringlichkeitszeiger in diesem Segment wichtig ist.
ACK bedeutet, dass das Bestätigungsfeld in diesem Segment wichtig ist.
PSH aktiviert während des Verbindungsaufbaus die Push Funktion. Sie bewirkt, dass die Daten von TCP an eine höhere Schicht weitergeleitet werden müssen. Bei nichtgesetztem Flag können die Daten in Warteschlangen gestellt werden.
RST trennt die Verbindung und verwirft alle anstehenden Daten.
SYN synchronisiert die Sequenznummern.
FIN signalisiert, dass keine Daten mehr vom Sender übertragen werden.

✖ Mit dem Fenster-Wert wird die genaue Größe des Puffers angezeigt, der von einem Endknoten für diese Verbindung reserviert wurde. Der sendende Host darf keinesfalls mehr Daten als die angegebene Puffergröße senden, ohne auf den Eingang einer Bestätigung zu warten.

✖ Die Prüfsumme gibt die Größe des gesamten Segments (Daten und Header) an. Dieses Feld dient zur Kontrolle der Datenintegrität, damit keine manipulierten oder beschädigten Segmente angenommen werden.

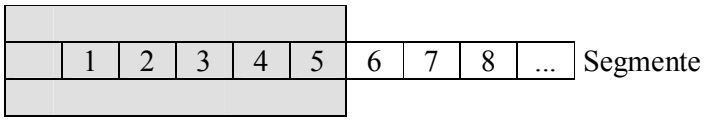
✖ Der Dringlichkeitszeiger wird bei besonders dringend zu verarbeitenden Segmenten gesetzt. Wenn gesetzt enthält dieses Feld als Wert den Bereich des Datenfeldes, das als dringlich gilt.

✖ Im Optionsfeld befinden sich verschiedene Optionen zur Kommunikation. Wie schon im IP-Header vergrößern die Füllzeichen das Optionsfeld auf 32 Bit.

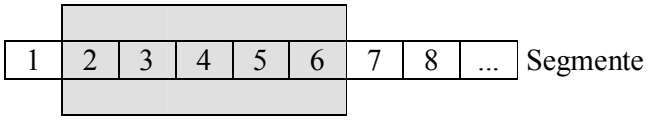
06.2 DATENÜBERTRAGUNG – DAS FENSTERPRINZIP

Einfache Transportprotokolle verwenden das folgende Prinzip: Sie schicken ein Segment, warten auf die Empfangsbestätigung und schicken dann ein weiteres. Falls die Bestätigung nach einer gewissen Zeit nicht erfolgt, wird das Segment noch einmal geschickt.
Dieser Mechanismus garantiert zwar Verlässlichkeit, aber er verwendet nur einen geringen Teil der vorhandenen Bandbreite.

Neuere Protokolle haben nun einen erweiterten Übertragungsmechanismus: Host A gruppiert seine Segmente und schickt die erste Gruppe, ohne auf ein ACK zu warten. Jedes Segment hat aber seinen eigenen Timeout-Wert. Host B sollte nun jedes Segment bestätigen, kann aber auch jeweils mehrere auf einmal bestätigen. Bei Host A rückt bei jeder erhaltenen Bestätigung das sogenannte "Fenster" weiter und die nächsten Segmente werden übertragen.



Fenster
(Schema "Fensterprinzip I")



Fenster wird verschoben →
(Schema "Fensterprinzip II")

06.2.1 SONDERFÄLLE

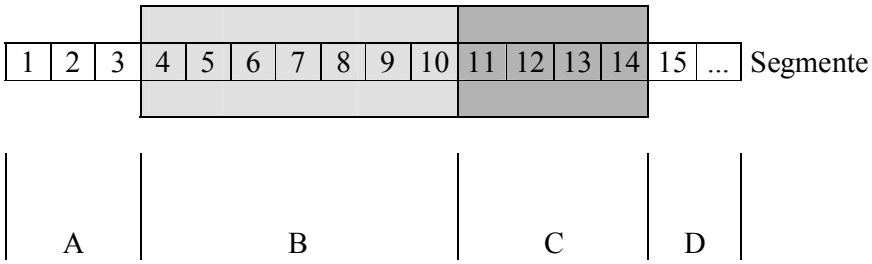
✖ Segment 2 geht verloren
Host A erhält keine Bestätigung für Segment 2, also bleibt das Fenster auf der Position 1. Da in diesem Beispiel immer 5 Segmente gesendet werden können, werden also Nummer 3, 4 und 5 immer noch mit ACK1 bestätigt, weil Segment 1 das letzte erhaltene dieses Stroms war. Nach dem Überschreiten des Timeouts wird Segment 2 neu übertragen, und Host B schickt ein ACK 5,weil die Segmente 1-5 erfolgreich übertragen wurden.

✖ Segment 2 kommt an aber ACK 2 geht verloren
Host A erhält zwar kein ACK 2, aber ACK 3. Da ACK 3 eine Bestätigung für alle Segmente von 1 bis 3 ist, kann Host A sein Fenster zu Segment 4 verschieben.

Die Verwendung des Fenster-Prinzips bedeutet verlässliche Übertragung bei besserer Nutzung der Netzwerk-Bandbreite. Außerdem wird die Datenflusskontrolle eingesetzt, weil der Empfänger auch erst ein ACK schicken kann, wenn er wieder Platz in seinem Puffer hat.

Die Fenster-Größe wird beim Verbindungsaufbau vom Empfänger bestimmt. Jeder ACK-Nachricht wird die maximale Fenstergröße zugefügt, die der Empfänger gerade bearbeiten kann. Befinden sich noch ungelesene Segmente im Puffer des Empfängers, dann wird die Größe des Fensters verkleinert.

Der Datenstrom des Senders kann also so dargestellt werden:



A Menge der Segmente, die übertragen und bestätigt wurden
B Menge der Segmente, die übertragen aber noch nicht bestätigt wurden
C Menge der Segmente, die ohne Bestätigung übertragen werden können (aber nicht müssen!)
D Menge der Segmente, die noch nicht übertragen werden können

TCP sendet seine Daten in verschieden langen Segmenten. Die Sequenznummern basieren auf Zählung der Bytes. Der dazugehörige Bestätigungswert (ACK) ist immer die Sequenznummer des als nächstes erwarteten Bytes.

Wenn ein Segment verloren geht, antwortet der Empfänger auf alle nachfolgenden Segmente mit der ACK-Nummer des fehlenden Segments. Nachdem alle Segmente eines Fensters übertragen sind, wartet der Sender auf die ACKs des Empfängers. Nach Überschreiten des Timeouts wird das fehlende Segment neu übertragen.

Ein Beispiel:

Quelle		Ziel
Sendet Segment 1 (Seq. 1000)	→	
	←	Erhält Seq 1000, bestätigt mit ACK 1500
Sendet Segment 2 (Seq. 1500)	¬	(geht verloren)
Sendet Segment 3 (Seq. 2000)	→	
Erhält ACK 1500 (Fenster wird verschoben)	←	
Sendet Segment 4 (Seq. 2500)	→	
	←	Erhält Seq 2000 und Seq. 2500, bestätigt mit ACK 1500 (Wartet noch immer auf Seq. 1500)
Erhält ACK 1500 (Fenster wird nicht verschoben)	←	
Timeout bei Segment 2 Erneute Übertragung von Segment 2 (Seq. 1500) →		
	←	Erhält Seq. 1500, bestätigt Segmente 2, 3, 4 mit ACK 3000
Erhält ACK 3000 (Fenster wird verschoben)		

06.2.3 WINDOW UPDATE

Jede Bestätigung enthält unter anderem auch den aktuellen Wert des freien Empfängerpuffers. Für den Fall, dass der Puffer voll ist, schickt der Empfänger in seinem ACK die Fenstergröße 0 mit. Sobald wieder Platz in seinem Speicher frei ist, wird noch einmal ein ACK geschickt, in dem eine neue Fenstergröße festgelegt ist.
Normalerweise werden ACKs nur als Antwort auf Segmente, die Daten enthalten, geschickt. In diesem Sonderfall jedoch wird das vorige ACK kopiert und mit dem neuen Fensterwert verschickt. Dieses ACK wird "Window Update" (Fensteraktualisierung) genannt.

Sollte nun dieses Window Update auf seinem Weg verloren gehen, dann wären beide Hosts in einer Schleife gefangen, jeder wartet auf die Nachricht vom anderen.
Hier hilft uns der sogenannte "Persist Timer" (Beharrungs-Zeitschaltuhr), der mit sogenannten "Window Probes" (Fenster-Sonden) vom Sender ausgehend in regelmäßigen Zeitabständen die Fenstergröße des Empfängers überprüft.

06.3 TIMEOUT

Um den Timeout-Wert eines Segments den Umständen im Netzwerk anzupassen, misst TCP die Zeit, in der ein Segment gesendet und dessen Bestätigung erhalten wird. Nach einigen Vorgängen wird ein Mittelwert der Übertragungsdauern errechnet, der als Timeout für die folgenden Segmente verwendet wird.

Dieser Mechanismus ist wichtig, weil verschiedene Verzögerungen auftreten können, abhängig zum Beispiel von der Belastung langsamer Netzwerke oder der Auslastung eines dazwischenliegenden Gateways.

06.4 VERBINDUNGSAUFBAU UND –TRENNUNG

Bevor Daten ausgetauscht werden können, muss eine Verbindung zwischen den Endknoten aufgebaut werden. Einer der Hosts (genannt "Server") muss auf "passive open" eingestellt sein, damit der Zweite ("Client") über einen "active open call" Kontakt aufnehmen kann.

SYN seq="n"	Host A sendet den Befehl SYN, er will also die Sequenznummern synchronisieren. Als Sequenznummer gibt er den Wert n an.
SYN ACK"n+1" seq="m"	Host B erhält die SYN-Nachricht und antwortet mit dieser Nachricht. ACK "n+1" bestätigt den Erhalt der Nachricht Nummer "n". "m" ist die Sequenznummer der Gegenanforderung zur Synchronisation.
ACK"m+1"	Host A bestätigt den Erhalt der Nachricht Nummer "m"

Die beiden Hosts sind nun verbunden und die zwei Datenströme (einer pro Richtung) wurden initiiert.
Aufgrund der Abfolge nennt man diesen Aufbau auch "3-fach Handshake".

Die Trennung dieser Verbindung erfolgt durch setzen des FIN-Flagbits. Das letzte Segment, das ein Host zu senden hat, wird mit der FIN-Flag versehen, die Verbindung wird in diese

Richtung getrennt. Der zweite Host bestätigt mit ACK, anschließend sendet er noch seine restlichen Daten und beendet die andere Richtung mit einer FIN-Flag. Erst nach Erhalt des zugehörigen ACKs ist die Verbindung endgültig getrennt.

Auch ein RST-Flag (reset connection) trennt eine bestehende Verbindung, jedoch werden hier (wie schon oben beschrieben) alle zum Versand anstehenden Daten verworfen.

06.5 CONGESTION CONTROL – VERSTOPFUNGSKONTROLLE

Ein großer Unterschied zwischen TCP und UDP ist der Verstopfungs-Kontroll-Algorithmus. TCP verhindert, dass ein Sender das Leistungsvermögen des Netzwerks überlastet.
TCP enthält 4 grundsätzliche Standards:

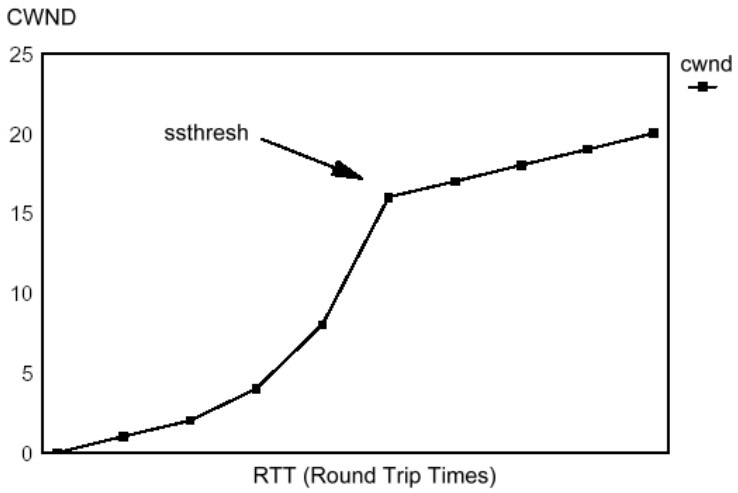
06.5.1 SLOW START

Hier wird die beste Senderate errechnet, indem der Mittelwert der Zeitspannen vom Senden eines Segments bis zum Erhalt der Antwort ermittelt wird. Dieser Wert wird rtt (round trip time; "Hin- und Rückfahrts-Zeit") genannt.
Mit Slow Start beginnt jede Verbindung mit nur 1 Segment pro Fenster (cwnd - congestion window; Verstopfungs-Fenster). Nach jedem gültigen ACK wird cwnd um 1 Segment vergrößert, solange bis die Fenstergröße mit der Angegebenen des Empfängers übereinstimmt.

Bsp.: Host A sendet 1 Segment und erhält ein ACK, cwnd wird also um 1 erhöht und Host A sendet 2 Segmente. Weil er jetzt 2 ACKs erhält, wird cwnd um 2 erhöht – auf 4 Segmente.
Durch diese Art der Erhöhung ergibt sich ein exponentielles Wachstum.

06.5.2 CONGESTION AVOIDANCE

Congestion Avoidance benötigt neben cwnd noch eine zusätzliche Variable: ssthresh (start threshold size, Anfangsgrenzwert). Beim Verbindungsaufbau setzt TCP cwnd auf 1 Segment und ssthresh auf 65535 Bytes.
Als Anfangswert gilt die vom Empfänger angegebene Fenstergröße. Bei Netzüberlastung (erkennbar durch Timeout oder doppelte ACKs) wird der aktuelle Wert der Fenstergröße halbiert (cwnd := cwnd/2) und in ssthresh gespeichert. Falls ein Timeout vorliegt, wird cwnd zusätzlich auf 1 Segment zurückgesetzt.
Sobald neue Daten mit ACKs bestätigt werden wird cwnd wie folgt erhöht:
Ist cwnd kleiner oder gleich ssthresh, wird im "Slow Start"-Modus erhöht.
Liegt der cwnd-Wert über ssthresh, geht TCP nach "Congestion Avoidance" vor und erhöht das Maximum folgendermaßen: cwnd = cwnd + Segmentgröße²/cwnd (angegeben jeweils Bytes).
cwnd wird also pro RTT um höchstens 1 Segment erhöht, das ergibt ein lineares Wachstum.



(Schema "Congestion Avoidance")

06.5.3 FAST RETRANSMIT

Auf jedes Segment antwortet TCP normalerweise mit dem dazugehörigen ACK. Doppelte ACKs können entweder den Verlust eines Segments bedeuten, oder auf falsche Reihenfolge der Segmente hindeuten.
Bei falscher Reihenfolge wären nur 1 oder 2 identische ACKs logisch.
Der Mechanismus Fast Retransmit vermeidet die Wartezeit auf ein Timeout, indem er nach 3 Duplikaten annimmt, dass ein Segment verloren ging. TCP schickt also sofort nach dem 3. gleichen ACK das wahrscheinlich fehlende Segment nach, ohne auf einen Timeout zu warten.

Quelle		Ziel
Paket 1	→	
Paket 2	→	
Paket 3 (geht verloren)	¬	(geht verloren)
	←	ACK 1
	←	ACK 2
Paket 4	→	
Paket 5	→	
Paket 6	→	
	←	ACK 2
	←	ACK 2
	←	ACK 2
Paket 3	→	
	←	ACK 6

...
(Schema "Fast Retransmit")

06.5.4 FAST RECOVERY

Nachdem Fast Retransmit das wahrscheinlich fehlende Segment noch einmal übertragen hat, wird Congestion Avoidance aktiviert (Slow Start wird also verhindert).

Das hat den Grund, dass ACKs nur gesendet werden können, wenn falsche Segmente eintreffen. 4 gleiche ACKs bedeuten also, dass 3 (in diesem Moment) falsche/unpassende Segmente erhalten wurden und noch im Puffer des Empfängers sind.

TCP will den Datenfluss nicht schlagartig mit Slow Start reduzieren. Nur ein verlorenes Segment lässt auf Verlust, nicht auf Stau schließen, also wird Congestion Avoidance eingeschaltet.

Der Wert von cwnd wird neu erstellt aus ssthresh + 3 * Segmentgröße. Bei jedem weiteren identischen ACK wird cwnd um eine Segmentgröße erhöht.
Sobald ein neues ACK empfangen wird, wird cwnd auf den Wert von ssthresh verringert.

06.6 ICMP-FEHLERMELDUNGEN

Die häufigsten ICMP-Fehlermeldungen sind "Source Quench", "Zielhost unerreichbar" und "Zielnetzwerk unerreichbar".
Die meisten Anwendungen reagieren so:

Eine erhaltene Source Quench – Nachricht setzt cwnd auf 1 Segment, belässt aber ssthresh. So wird zuerst Slow Start aktiviert, später aber zu Congestion Avoidance umgeschaltet.

"Zielhost unerreichbar" oder "Zielnetzwerk unerreichbar" werden schlicht ignoriert, da diese Fehler nur vorübergehend auftreten können. So könnte zum Beispiel ein dazwischenliegender Router ausgefallen sein, und dadurch müsste erst eine neue Route gefunden werden.
Das kann durchaus einige Minuten dauern, TCP versucht aber weiterhin, die Daten zu senden, die die Fehlermeldung ausgelöst haben.

07 DNS (Domain Name System) – Schicht 4

Da man sich Wörter oder logische Buchstabenfolgen viel leichter merken kann als 12stellige Dezimalzahlen, gibt es im Internet einen speziellen Dienst: Domain Name System.
Computer, die diesen Dienst im Internet anbieten, übersetzen die sogenannten Domainnamen in IP-Adressen und umgekehrt.
Man kann also in einen Browser statt 194.232.104.21 auch einfach www.orf.at eingeben, und kommt ans selbe Ziel.

Ursprünglich gab es nur eine Tabelle, die Datei hosts.txt , die IP-Adressen in Namen umwandelte und umgekehrt. Diese wurde vom Network Information Center (NIC) gewartet und konnte per FTP auf den eigenen Host geladen werden. Dieses Prinzip wird "flat namespace" genannt.

Durch das explosionsartige Ansteigen der IP-Adressen und Hostnamen wurde ein neuer Mechanismus nötig, das Domain Name System.
Auf speziellen Hosts laufen Programme, die die Anforderungen von einzelnen Übersetzungen bearbeiten. Dadurch ist es nicht mehr nötig, die gesamte Datenbank auf jeden einzelnen Computer zu laden.

Domains sind hierarchisch aufgebaut: Die Adresse tv.orf.at gibt den Hostnamen "tv" in der Subdomain "orf.at" an, und "orf" ist wiederum eine Subdomain von at.
Die letzte Endung einer Adresse wird als Top-Level Name (höchste Schicht) bezeichnet

07.1 FQDNs (FULLY QUALIFIED DOMAIN NAMES)

FQDNs sind absolute Domainangaben, die mit einem Punkt beendet werden. Wenn ein Domainname mit einem Punkt endet (z.B. wtscpok.itsc.pok.ibm.com.) wird die Angabe als komplett angenommen.
Wenn in einer Anfrage aber nur wtscpok.itsc steht, kann der eigene Host diese relative Angabe eventuell vervollständigen.

07.2 GENERIC DOMAINS

Als Generic Domains bezeichnet man die folgenden Top-Level Namen, die aus drei Buchstaben bestehen.
Diese Art der Namen stammt aus den Anfängen von DNS, als das Internet nur in den Vereinigten Staaten verbreitet war.

com	Komerzielle Organisation (commerical)
edu	Ausbildende Institution (educational)
gov	Regierungsinstitution (governmental)
int	Internationale Organisation (international)
mil	U.S. Militär
net	große Netzwerke
org	Non-profit Organisationen
biz	Unternehmen (business)

07.3 COUNTRY DOMAINS

Ländernamen bestehen aus 2 Buchstaben und gleichen den ISO 3166 Ländercodes. Sie reichen von .ae (Vereinigte Arabische Emirate) bis .zw (Zimbabwe).
Hier entstammen auch neuere Endungen wie .tv (Tavalonien) und .cc (Coconut Inseln).

Einige Beispiele sind:

.at	Österreich
.au	Australien
.de	Deutschland
.fr	Frankreich
.it	Italien
.nl	Niederlande

Viele Staaten haben zusätzlich ihre eigenen "Second-Level" Domainnamen wie .co.at (für Unternehmen in Österreich), .ac.at (für Ausbildungsstätten in Österreich) und .gv.at (für Regierungsseiten in Österreich).

Die Übersetzung von Domainnamen in IP-Adressen läuft auf unabhängigen, zusammenarbeitenden Systemen, genannt Name Server.
Auf jedem Name Server sind Tabellen für die Übersetzung angelegt, die von Clientanwendungen ("Name Resolver") abgefragt werden.

Eine Abfrage läuft in etwa so ab:
Ein Benutzerprogramm formuliert eine Anfrage zur Übersetzung eines Domainnamens in eine IP-Adresse und schickt diese an einen Name Server. Am Name Server wird geprüft, ob sich der Name in einer der eigenen Tabellen oder im Zwischenspeicher befindet. Falls nicht, schickt er die Anfrage weiter an hierarchisch höherliegende Server.
Das Benutzerprogramm erhält schließlich die zugehörige IP-Adresse bzw. eine Fehlermeldung falls keine Übereinstimmung gefunden wurde.

Die Übertragung funktioniert sowohl über UDP als auch über TCP. Prinzipiell muss ein Host aber zuerst via UDP versuchen, eine Übersetzung zu erhalten, weil UDP-Messages weniger Kopfdaten besitzen.
Bei oft benutzten Services wie DNS ist der Gebrauch von UDP unerlässlich.
Erst wenn keine Antwort über UDP erhalten wurde, kann eine TCP-Anfrage gestellt werden.

08 FTP (FILE TRANSFER PROTOCOL) – Schicht 4

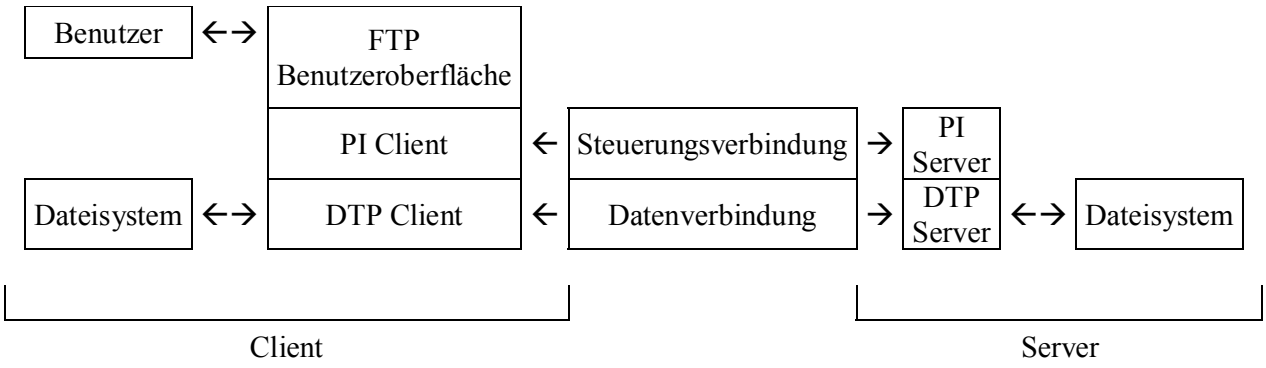
FTP ist ein Standartprotokoll, das für den Austausch von Dateien über Netzwerke entwickelt wurde.
Der Datentransfer ist in beide Richtungen möglich, einerseits um Dateien VON einem Server zu laden, andererseits um Daten AUF einen Server zu übertragen.

FTP benutzt TCP für verlässliche Verbindungen.
Die "well-known-ports" für den FTP-Server sind 20 und 21.
Weil Daten übertragen werden sollen, sollte das IP TOS-Bit "Maximum Throughput" gesetzt sein.

Im Grunde werden zwei Verbindungen benutzt: Eine zum Anmelden mit Username und Passwort, die andere, um den Datentransfer zu regeln.

Auf beiden Seiten arbeiten sogenannte PIs (protocol interpreter; Protokollübersetzer), der DTP (data transfer process; Datentransferprozess) und beim Client noch eine Benutzerschnittstelle (user interface).
Über die Benutzerschnittstelle kann der jeweilige Benutzer seine Befehle eingeben. Eine Schicht tiefer liegt der PI, der die notwendigen Daten über das eigene Dateisystem übermittelt.
Am Server startet der PI die Verbindung, und während des Datentransfers regeln die DTPs die Verwaltung der Daten.

Schema FTP-Kommunikation:



08.1 FTP BEFEHLE

Der Client benötigt Username und Passwort für ein korrektes Login beim Server, es sei denn, er verwendet "Anonymous FTP".
Um öffentlichen Zugang zu einigen Dateien zu ermöglichen, erlauben viele Server auch "Anonymous FTP" oder "Anonymous Login". Hier kann mit der Benutzerkennung "Anonymous" und irgendeinem Passwort sofortiger Zugang ohne Registrierung erlangt werden.

Über die folgenden Befehle ist das in jeder Befehlszeile möglich:

ftp	Startet das Programm FTP (in einer Befehlszeile).
open [hostadresse]	Stellt die Verbindung zum gewünschten Host her. Die Hostadresse kann entweder als IP oder als Domain Name angegeben werden.
user	Dient zur Eingabe des Benutzernamens (sofern nicht automatisch danach gefragt wird).
pass	Dient zur Eingabe des Passworts (wenn nicht automatisch gefragt wird).

Sobald die Verbindung aufgebaut ist, kann über die folgenden Standartbefehle gearbeitet werden:

cd	(change directory) Wechselt das Arbeitsverzeichnis.
dir / ls / list	Auflistung der vorhandenen Dateien anfordern.
type	Gibt die gewünschte Übertragungsart an.
ASCII	Verwendet den ASCII-Zeichensatz (oder ggf. eine ASCII-EBCDIC Übersetzung)
EBCDIC	Verwendet den EBCDIC-Zeichensatz
Image	Versendet die Daten in 8-bit-Paketen
pasv	Startet "passive mode", ein Umkehrmodus für Benutzer, die hinter einer Firewall liegen.
get / retr	Kopiert eine Datei vom Server zum Host
mget	Kopiert mehrere Dateien vom Server zum Host
put / send	Kopiert eine Datei vom Host zum Server
mput	Kopiert Mehrere Dateien vom Host zum Server
verbose	Zeigt zusätzliche Informationen während der Datenübertragung.
delete	Löscht eine Datei vom Server
mdelete	Löscht mehrere Dateien vom Server
close	Bendet die FTP-Sitzung aber lässt das FTP-Programm laufen.
quit / bye	Beendet die FTP-Sitzung und das FTP-Programm.

08.2 ANTWORTCODES

Auf jede Anweisung vom Benutzer antwortet der Server mit Antwortcodes und Benutzerfreundlichen Erläuterungen. Diese Codes bestehen aus 3 Ziffern, wobei die Erste über die Bedeutung Ausschlag gibt.

1xx	Positive Zwischenmeldung
2xx	Positive Endmeldung

3xx	Positive sofortige Meldung
4xx	Vorübergehend negative Meldung
5xx	Dauerhaft negative Meldung

Beispiele für solche Server-Nachrichten sind:

150	opening ascii mode data connection for file list
200	command ok
220	server ready
226	transfer complete
331	user name ok
425	can't open data connection
530	not logged in
550	"filename": no such file or directory

08.3 ANWENDUNGEN

Es gibt eine Vielzahl von FTP-Programmen, die durch graphische Oberfläche, Mausklicks und Drag&Drop noch einfacher zu handhaben sind. Beispiele sind WS_FTP, CoffeeCup FTP, FTP Voyager, ...

08.4 TFTP (TRIVIAL FILE TRANSFER PROTOCOL)

Das "unbedeutende Datenübertragungsprotokoll" baut auf UDP auf, hat die meisten Funktionen von FTP verloren und beschränkt sich auf Lesen und Schreiben.
Auch eine Anmeldung mit Passwort ist nicht möglich.

08.4.1 TFTP BEFEHLE

connect [hostname]	Verbindet mit dem angegebenen Host.
mode	Gibt die gewünschte Übertragungsart an.
ascii	ASCII-Daten
binary	Binäre Daten
get	Kopiert eine Datei vom Server auf den Host.
put	Kopiert eine Datei vom Host auf den Server.
verbose	Zeigt zusätzliche Informationen während der Datenübertragung.
quit	Beendet die Verbindung und TFTP.

09 HTTP (HYPER TEXT TRANSFER PROTOCOL) & WWW – Schicht 4

Das Protokoll HTTP dient im Grunde zur Übertragung von HTML (Hyper Text Markup Language) Dokumenten. HTML ist eine simple Art einer Programmiersprache für Hypertextdokumente, die durch Hyperlinks mit anderen Seiten im Internet verbunden sind. (Inzwischen werden auch viele mehr oder weniger verwandte Programmiersprachen übertragen, wie XML, JAVA & JVM, JavaScript, ...)
Solche Dokumente können neben Text auch Bilder, Graphiken und Bilder, Audio- und Videodateien und vieles mehr enthalten.

HTTP basiert auf dem Anfrage-Antwort-Prinzip und wird normalerweise über TCP verschickt. Es kann jedoch auch jedes andere verlässliche Transportprotokoll verwendet werden.
Bei den ersten HTTP-Versionen stellte ein Benutzerprogramm (Browser) des Clients eine Verbindung zum Server her und schickte zum Beispiel die Anfrage für ein HTML-Dokument. Der Server antwortete mit einer Nachricht, die die verwendete Protokollversion, einen Erfolgs- oder Fehlercode, und die Datei inklusive Informationen beinhaltete. Anschließend trennte der Server die Verbindung.
Durch diese einfache und kurze Verbindung benötigte man aber für eine Seite mit 3 Bildern bereits 4 Verbindungen – eine für jede Datei.

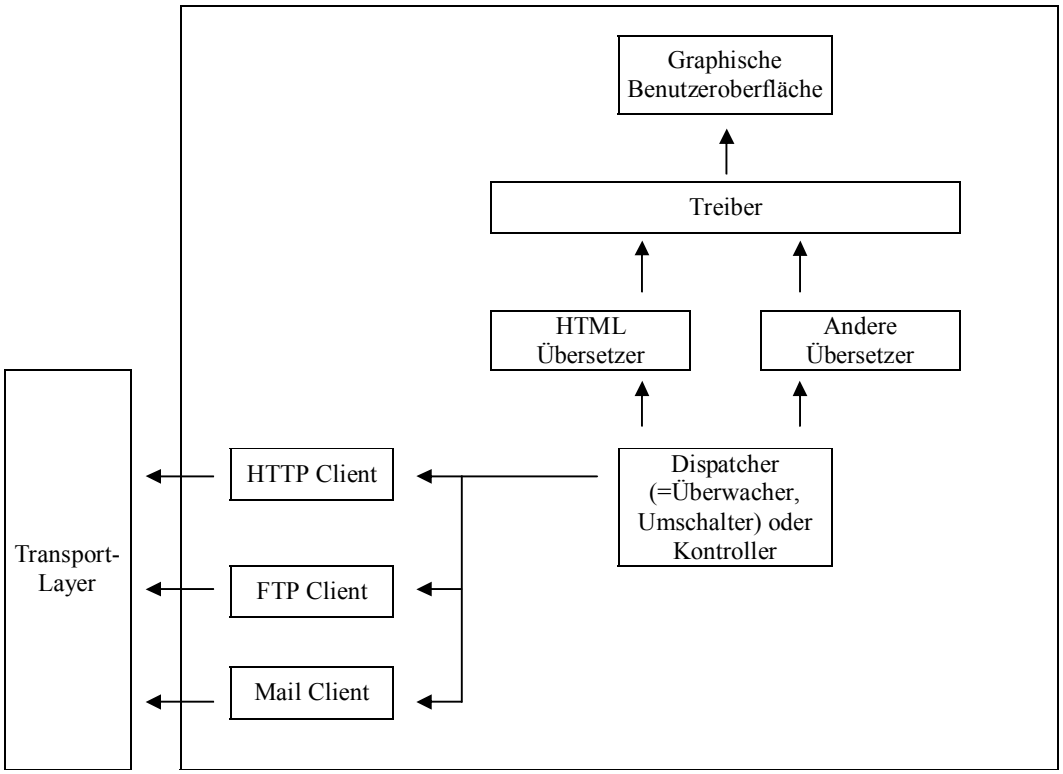
Die folglich großen Serverbelastungen wurden in der HTTP1.1-Version verringert. Wird eine Seite geladen, dann gibt es je eine längere Verbindung, über die die Gesamte Anfrage (also HTML-Seite inkl. Bilder, Sounds, ...) gesendet wird.

WWW wurde 1989 in der Schweiz von Forschern der Teilchenphysik (CERN) entwickelt, um Forschungsdaten und –dokumente leichter zu finden und für andere Forscher zugänglich zu machen.
Es dient als weltweites Hypertextsystem, dessen Benutzung 1993 durch die Entwicklung des ersten Web-Browsers "Mosaic" schlagartig anstieg. Mosaic war die erste graphische Schnittstelle zum Web und erleichterte das Durchstöbern des Internets enorm.

Heutzutage gibt es eine Vielzahl von Browsern mit integriertem FTP-Programm, e-mail-Funktion, und vielen anderen Extras, wie zum Beispiel Opera, Netscape Navigator oder MS Internet Explorer.

09.1 DIE FUNKTIONSWEISE

Die Funktionsweise und Struktur eines solchen Web-Browsers wird mit dem nachfolgenden Schema relativ gut erläutert:



09.1.1 URI

Der sogenannte "Uniform Resource Identiffier" ist ein Oberbegriff für alle gültigen Adressierungsmuster. Das gängige Adressierungsschema ist URL.

09.1.2 URL

Als URL (Uniform Resource Locator; eindeutige Quellbezeichnung) bezeichnet man die Adressen, die wir tagtäglich in unsere Browser eingeben, ohne uns nähere Gedanken zu deren Bedeutung zu machen.

Die Elemente werden am Beispiel der Website http://tv.orf.at/main/tv.html erklärt.

http:// Bezeichnet das verwendete Protokoll. Bei den gängigen Browsern muss dieses nicht extra eingegeben werden, sie stellen es automatisch vor die eingegebene Adresse.

tv Deutet auf den Host "tv" im orf.at-Unternetzwerk.
orf Deutet auf das Unternetzwerk (Subdomain) "orf" von ".at".

at AT ist der sogenannte "Country Domain Name" von Österreich.
/main Zeigt, dass die angegebene Datei im Ordner "main" zu finden ist.
tv.html Beschreibt die gesuchte Datei "tv.html".
.html oder .htm ist die gängigste Dateinamenerweiterung für Websites in der HyperText Markup Language.

09.2 HTTP-VERBINDUNGEN

Normalerweise werden Verbindungen direkt zwischen dem Client und dem Server aufgebaut.

Client ←→ Server

In manchen Fällen ist es aber nötig, über sogenannte "intermediaries" (Mittelsleute, z.B. Gateway, Proxy, Tunnel...) Kontakt zum gewünschten Host aufzunehmen. Anforderungen und Antworten werden dann zwischen den Mittelsleuten weitergegeben und an das Ziel transportiert.

Client ← → Proxy A ←→ Proxy B ←→ Server

Weil Gateways und Proxy-Server die weiterzugebenden Daten direkt verarbeiten können, ist eine Zwischenspeicherung häufig angeforderter Daten im Cache oft sinnvoll. So kann bei einer erneuten Anfrage (z.B. einer Webseite) der nächste Proxy/Gateway direkt antworten, ohne die anderen mit häufigen Anfragen zu belasten.

09.2.1 METHODEN

GET	Der GET-Befehl dient der Informationsanforderung zur Auffindung und Darstellung von Dokumenten.
HEAD	Die Methode HEAD funktioniert grundsätzlich wie GET, nur werden hier ausschließlich Headerinformationen übertragen, keine Daten.
POST	POST dient der Übermittlung von Daten vom Client zum Server, die an Datenverarbeitungsprogramme gerichtet werden (z.B. Datenbankoperationen, CGI-Skripte...)

Die drei Haupt-Methoden GET, HEAD und POST werden von jedem Server unterstützt. Die weiteren Methoden sind noch nicht so weitverbreitet und werden nicht immer verstanden.

PUT	Fordert an, dass der Datenteil der Anforderung an der angegebenen Adresse abgelegt wird.
DELETE	Fordert die Löschung der Daten einer bestimmten Adresse an.
TRACE	Wird zur Fehlersuche verwendet und fordert an, dass der gesendete Body-Abschnitt wieder intakt zurückgegeben wird.

09.2.2 STRUKTUR DER NACHRICHTEN

HTTP-Nachrichten (Anforderung/request oder Antwort/response) werden in "Status-Code", "Header" und "Body" unterteilt.

status-code Die erste Zeile zeigt die Version des HTTP-Protokolls und den Antwort-Code des Servers (mit anwenderfreundlichen Erläuterungen).

header Ab der zweiten Zeile folgen die Kopfdaten oder Headerdaten mit weiterführenden Informationen zur Nachricht. Der Header-Bereich wird mit einer Leerzeile abgeschlossen.

body Hier kommen die eigentlichen Informationen (z.B. HTML-Code der angeforderten Seite)

Die HTTP-Header bestehen aus den folgenden Feldern:

Art der Nachricht	Entweder eine Anforderung (request) vom Client oder eine Antwort (response)vom Server.	
Nachrichten Header	Die Werte für HTTP-Nachrichten-Header können in 4 Bereiche eingeteilt werden.	
✖ **Allgemeine Header**	dienen der Übermittlung grundsätzlicher Informationen.	
Beispiele wären:	cache-control	Anweisung über die Aufbewahrung der Seite im Cache eines Proxys
	connection	Spezielle Angaben zur Aufrechterhaltung der Verbindung
	upgrade	Hinweis auf verfügbares neueres Protokoll und Anfrage zum Wechseln in dieses Protokoll
✖ **Request Header**	geben in erster Linie Aufschluss über die Art und Konfiguration des Clients.	
Beispiele wären:	accept	Nennt die vom Browser bevorzugten und unterstützten Formate. [Typ]/[Untertyp] zB: text/* (jede Textart), image/gif (=Gif-Bilder)
	cookie	Dieser Wert wird automatisch gesetzt, wenn bereits in einer früheren Verbindung ein Cookie gesetzt wurde.
	referer	Adresse der Seite, deren Link auf die aktuelle Seite geführt hat.
✖ **Response Header**	werden mit den angeforderten Daten vom Server an den Client mitgeschickt.	
Beispiele wären:	age	Nennt das Alter des Dokuments in Sekunden.
	location	Gibt die neue Adresse eines Dokuments an, das verschoben wurde.
	set-cookie	Aufforderung zum Ablegen eines Cookies.
	content-type	Gibt den Medientyp der gelieferten Daten an.
✖ **Entity Header**	liefern Informationen über den Datenabschnitt in einer HTTP-Meldung.	
Beispiele wären:	content-length	Gibt die Länge des Datenabschnitts in Byte an.
	expires	Definiert den Zeitpunkt, an dem der Inhalt eines Dokuments ungültig wird. Das Dokument wird nach Ablauf dieser Deadline gelöscht oder geändert.

09.2.3 BEISPIEL FÜR KOMMUNIKATION

Der Client schickt eine GET-Anforderung:

```
--
GET /index.html HTTP/1.0                    Anforderung des Dokuments index.html, HTTP-Version 1.0
User-Agent: Mozilla/2.02Gold (WinNT; I)     Optionale Daten werden mitgeschickt (z.B. Browser-
                                            Software, Betriebssystem, ...)
Host: www.orf.at                            Angabe von Host- und Portnummer der Adresse
Accept: image/gif, image/jpeg, image/pjpeg, */*  Der Client definiert bevorzugte und unterstützte Formate
                                            Beendigung des Headers mittels Leerzeile
--
```

Der Server beantwortet die Anforderung:

```
--
HTTP/1.0 200 OK                             HTTP-Version und Bestätigung der Anforderung durch Statuscode
Date: Fri, 30 Jan 2003 08:17:58 GMT             Angabe von aktuellem Datum und aktueller Uhrzeit
Server: NCSA/1.5.2                              Angabe von Name und Versionsnummer des Servers
Last-modified: Mon, 17 Jun 2002 20:03:02        Angabe der letzten Änderung der Datei
Content-type: text/html                         Angabe des Dokument-Typs
Content-length: 2482                            Länge des übermittelten Dokuments in Byte
                                                Beendigung des Headers mittels Leerzeile
(Ab hier kommen die Daten)
--
```

09.2.4 ANTWORT- ODER STATUSCODES

Wie schon bei FTP sendet auch das HTTP-Protokoll Antwortcodes in seinen Nachrichten mit. In Browsern werden diese Codes jedoch normalerweise nicht angezeigt.

1xx	zur Information ("informational")
2xx	Erfolgreiche Meldung
3xx	Weiterleitung
4xx	Fehler beim Client
5xx	Fehler beim Server

Beispiele für solche Status-Codes sind:

101	Wechsel des Protokolls
200	OK
204	kein Inhalt
301	Datei dauerhaft verschoben
302	Datei vorübergehend verschoben
403	Zugriff verweigert
404	Datei nicht gefunden
415	nicht unterstütztes Dateiformat
500	Interner Serverfehler

503	Service derzeit nicht verfügbar
505	HTTP Version nicht unterstützt

Ein Beispiel für eine komplette HTTP-Nachricht (inkl. Daten) sieht so aus:

```
HTTP/1.1 200 OK
  Date: Sat, 17-March-01 11:45:13 GMT
  Server: Apache/1.3
  Content-type: text/html
  Content-length: 114

  <html>
    <title>ORF Startseite</title>
    <body>
      Willkommen auf www.orf.at!
    </body>
  </html>
```

10 TELNET – Schicht 4

Das Telnet-Protokoll ist eine standardisierte Schnittstelle, durch die ein Programm auf einem Host (Client) auf die Ressourcen eines anderen Hosts (Server) zugreifen kann. Telnet ist daher unabhängig von Betriebssystem, Computerhersteller etc.

10.1 AUFBAU

Die Kommunikation zwischen Telnet-Client und Telnet-Server ist folgendermaßen aufgebaut:

- NVT – Network Virtual Terminal

Beim Erstellen einer Verbindung zwischen Client und Server über Telnet bilden die beiden Computer zusammen ein NVT, ein virtuelles Terminal. Der Telnet-Client mit Monitor und Tastatur ist virtuell direkt an dem Server angeschlossen.
Die grundlegenden Merkmale von NVT sind das Verwenden des 7-bit ASCII-Zeichensatzes (Versand als 8-bit ASCII mit dem ersten Bit als 0) und Half Duplex (Wechselseitige Verwendung einer Übertragungsleitung).

- Verhandeln über Optionen

Diese Optionen dienen dazu, dass die beiden Rechner einwandfrei miteinander kommunizieren können. Ein Beispiel für eine solche Option ist die Tastaturbelegung, sie muss gegebenenfalls angepasst werden.
Mit den 4 Befehlen "Do, Don't, Will, Won't" können die Hosts die weiteren Optionen verhandeln.

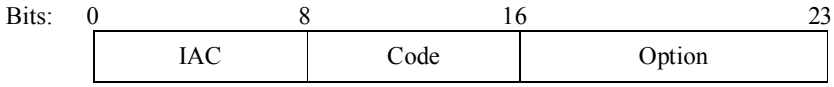
- Symmetrische Verbindung

Bei der Telnetverbindung sind beide Rechner gleichberechtigt, während man von Host A auf Host B zugreift, ist es also auch möglich, von Host B auf Host A zuzugreifen.

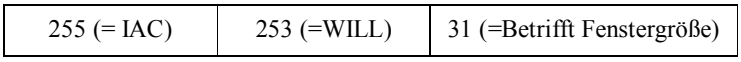
Nach erfolgreichem Aufbau einer Verbindung wird zuerst das gegenseitige Verständnis nachgeprüft. Durch diese erste Anpassung können die beiden Hosts bereits auf dem niedrigsten Level von NVT arbeiten.
Darauffolgend verhandeln sie über weitere Optionen um die Leistungsfähigkeit zu steigern und an die Fähigkeiten der realen Hardware anzunähern.
Durch das Symmetriemodell können beide Hosts weitere Optionen vorschlagen.

10.2 BEFEHLSSTRUKTUR

Die Kommunikation zwischen Client und Server besteht aus internen Befehlen, die für den Benutzer nicht zugänglich sind.
Jeder Befehl besteht aus 2 oder 3-Byte-Folgen.
Das erste Byte hat mit dem Wert 255 die Bedeutung IAC (interpret as command; Es folgt ein Befehl), das zweite Byte beschreibt den Befehlscode und das dritte Byte wird für Optionen verwendet.



zum Beispiel:



Einige Befehle:

IP	244	Prozess unterbrechen (interrupt process)
EL	248	Zeile löschen (erase line)
GA	249	Weitermachen (go ahead)
WILL	251	Akzeptiert Aufforderung zum Einschalten einer Option
WON'T	252	Akzeptiert Aufforderung zum Ausschalten einer Option
DO	253	Aufforderung, eine Option einzuschalten
DON'T	254	Aufforderung, eine Option auszuschalten
IAC	255	Es folgt ein Befehl (interpret as command)

10.2.1 TELNET OPTIONEN

Einige Optionen inklusive Codes wären:

0	Binäre Übertragungsart
1	Echo
5	Statusabfrage
18	Logout
31	Über Fenstergröße verhandeln

10.2.2 VERHANDELN ÜBER OPTIONEN

Grundlegend für jede Verhandlung ist das gemeinsame Verwenden von NVT. Jede Option kann mit den 4 Befehlscodes "Do, Don't, Will, Won't" verhandelt werden.
Weiters können mit den Befehlen SB (suboption begin) und SE (suboption end) Unteroptionen behandelt werden.

Beispiel für Verhandlung:

Anfrage	Antwort	Erklärung
DO transmit binary	WILL transmit binary	Übertragungsart auf Binär gesetzt
(Der Interne Befehl "IAC DO BIN" wird übertragen als "255 253 0")		
DO window size	WILL window size	Können wir über die Fenstergröße verhandeln?
SB window size 0 80 0 24 SE		Angabe der gewünschten Fenstergröße
DO echo	WON'T echo	Ablehnung der Anfrage für Opion "Echo"

11 STICHWORTVERZEICHNIS

12 QUELLEN

- Buch "TCP/IP Tutorial and Technical Overview"
  (Martin W. Murhammer, Orcun Atakan, Stefan Bretz, Larry R. Pugh,
  Kazunari Suzuki, David H. Wood)

- Buch "Networking with Microsoft TCP/IP"
  (Drew Heywood, Rob Scrimger)

- Buch "TCP/IP Illustrated - Volume 1: The Protocols"
  (W. Richard Stevens)

- Buch "Illustrated TCP/IP"
  (Matthew G. Naugle)

- "Einführung in TCP/IP"
  http://www.rvs.uni-bielefeld.de/~heiko/tcpip/download/tcpip.zip

- "Das OSI-Referenzmodell"
  http://karschten.freepage.de/wissen/anzeige/osi.htm

- "Basisdienste II"
  http://www.inf-wiss.uni-konstanz.de/CURR/summer01/inetd/basisdienste210601_230601.pdf

- Siemens Lexikon der Datenkommunikation
  http://w3.siemens.de/solutionprovider/_online_lexikon/start.htm