Klaus Gerhardt – Linux Seiten

Linux im Netzwerk I - Netzwerkgrundlagen

Netzwerkgrundlagen

Copyright, Nutzungsbedingungen, Haftungsausschluss

Feedback ist willkommen

Feedback ist willkommen. Feedback, Fehlermeldungen, Korrekturen, etc. bitte senden an:

k-gerhardt@gmx.de

Netzwerkgrundlagen

Dieses Kapitel darzustellen ist gar nicht so einfach. Der Stoff ist recht umfangreich, von daher ist hier die Kunst der Auslassung gefragt. Auf jeden Fall ist diese Beschreibung nicht vollständig. Sie soll jedoch den Leser in die Lage versetzten bei Bedarf eigenständig weiterzustudieren. Ausführlich beschrieben habe ich immer nur aktuelle Themen, also z.B. Ethernet. Veraltete Techniken nur am Rande erwähnt. Und noch ein Hinweis: leider lässt es sich bei einem solch komplexen und verwobenen Thema nicht vermeiden, dass manchmal Begriffe verwendet werden die erst später ausführlich erklärt werden. Ggf. muss der Leser dann im Text springen um sich an anderer Stelle zu informieren. Da es sich im Prinzip um mehrere in sich abgeschlossene Themen handelt, kann man sich natürlich auch einzelne Themen herauspicken.

Netzwerke

Was ist ein Netzwerk?

Für ein Computernetzwerk müssen mindestens 2 Computer, zum Zweck des Datenaustauschs, miteinander verbunden werden. Beide Computer müssen in der Lage sein eigenständig Programme auszuführen. Daraus folgt dann auch, dass ein zentraler Unix-Rechner mit mehreren Terminals kein Netzwerk darstellt. Denn die Terminals selbst sind nicht in der Lage Berechnungen auszuführen. Die Programme werden alle auf dem Unix-Zentralrechner ausgeführt. Eine weitere Bedingung ist, dass alle Rechner im Netzwerk über die entsprechenden Mechanismen verfügen um miteinander kommunizieren zu können.

Netzwerk-Topologien

Eine Netzwerktopologie beschreibt die Anordnung der Rechner im Netz. Wobei man hier noch zwischen physikalischen und logischen Topologien unterscheidet.

Physikalische Topologien

Die physikalische Topologie beschreibt dabei wie die Rechner mit Hilfe von Kabeln und ggf. notwendigen Netzwerkgeräten, wie z.B. Hubs oder Switches, miteinander verbunden sind.

Die Bus-Topologie

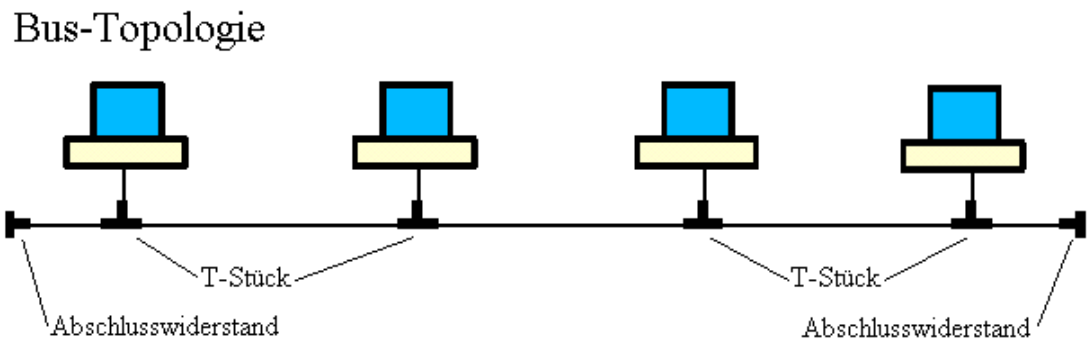
Bei der Bustopologie hängen alle Computer an einem einzigen Kabelstrang. Von den beiden Geräten am den Enden des Kabelstranges abgesehen, hat jedes Gerät 2 Nachbarn. In der Computertechnik ist diese Topologie allerdings veraltet, hat aber trotzdem noch eine gewisse Bedeutung, da sie dem Ethernetprotokoll zum Durchbruch verholfen hat. Dieses Netzwerk-Protokoll ist im LAN (Local Area Network) heute das am häufigsten verwendete.

Realisiert wird diese Netzform mit Hilfe von Koaxial-Kabeln. Diese werden auch als BNC-Kabel bezeichnet. Es gab 2 Varianten dieser Ethernet-Technologie

* 10Base5 mit den Kabeln RG 8 (Thick-Ethernet)
* 10Base2 mit den Kabeln RG 58 (Thin-Ethernet, Cheapernet)

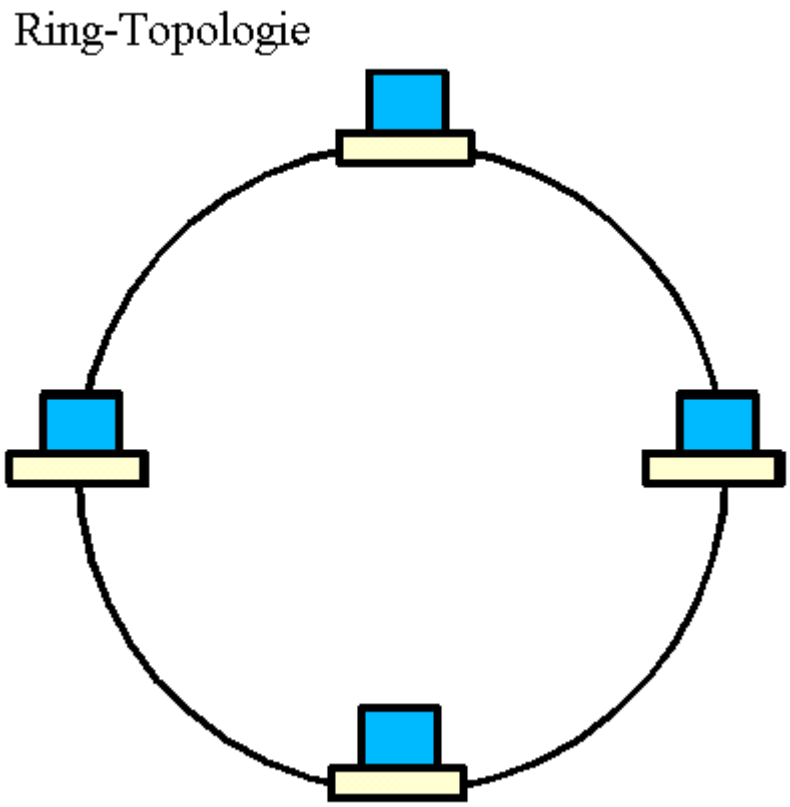
10Base2 mit dem dünneren Kabel wurde später eingeführt um ein preiswerteres Ethernet zu ermöglichen.

Die einzelnen Computer werden mit T-Stücken an das Kabel angeschlossen. Die beiden Endpunkte müssen mit Abschlusswiderständen terminiert werden. Die max. Übertragungsgeschwindigkeit ist 10 MBit/s. Der grosse Nachteil dieser Technik ist, dass eine Unterbrechung auf dem Kabel das gesamte Netzwerk lahm legt.



Die Ring-Topologie

Diese Art der Verkabelung ist heute ausgestorben. Verwendet wurde sie für Token Ring Netzwerke. Wie der Name schon sagt, handelt es sich um einen geschlossenen Ring. Jede Station im Netz hat also 2 Nachbarstationen.



Die Stern-Topologie

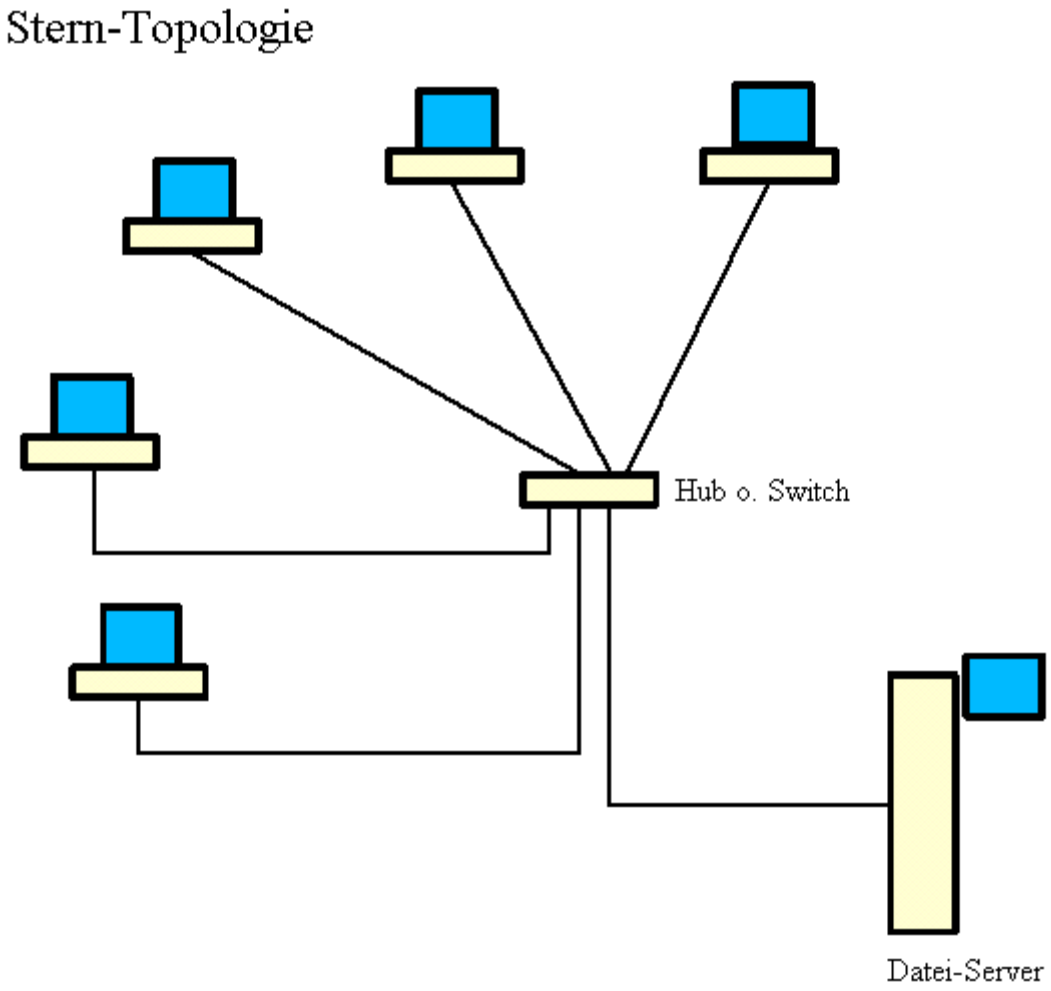
Dies ist die heute am meisten verwendete Form. An ein zentrales Netzwerkgerät werden alle Computer angschlossen. Dieser Sternpunkt heisst Hub oder Switch. Während der Hub alle Stationen miteinander verbindet, werden beim Switch immer nur 2 miteinander kommunizierende Rechner verbunden. Da diesen jetzt die Verbindung exclusiv zugeordnet ist, wird damit auch die Bandbreite der Verbindung erhöht. Wenn eine Störung auf einem Kabel auftritt, dann ist auch nur die betreffende Verbindung betroffen. Es wird also nicht das ganze Netz lahm gelegt wie bei der BusTopologie.

Für die Verkabelung werden fast ausschliesslich Twisted Pair Kabel verwendet. Diese haben 2 oder 4 Adernpaare. Die Adern eines Paares sind miteinander verdrillt und die Paare sind ebenfalls miteinander verdrillt. Diese aus der Telefonie stammende Technik dient der Unterdrückung von Störsignalen auf der Leitung. Man unterschiedet auch noch zwischen geschirmten und ungeschirmten Twisted Pair Kabeln. Auf englisch dann: Shielded bzw. Unshielded Twisted Pair.

Hier gibt es auch wieder verschiedene Ethernet-Technologien:

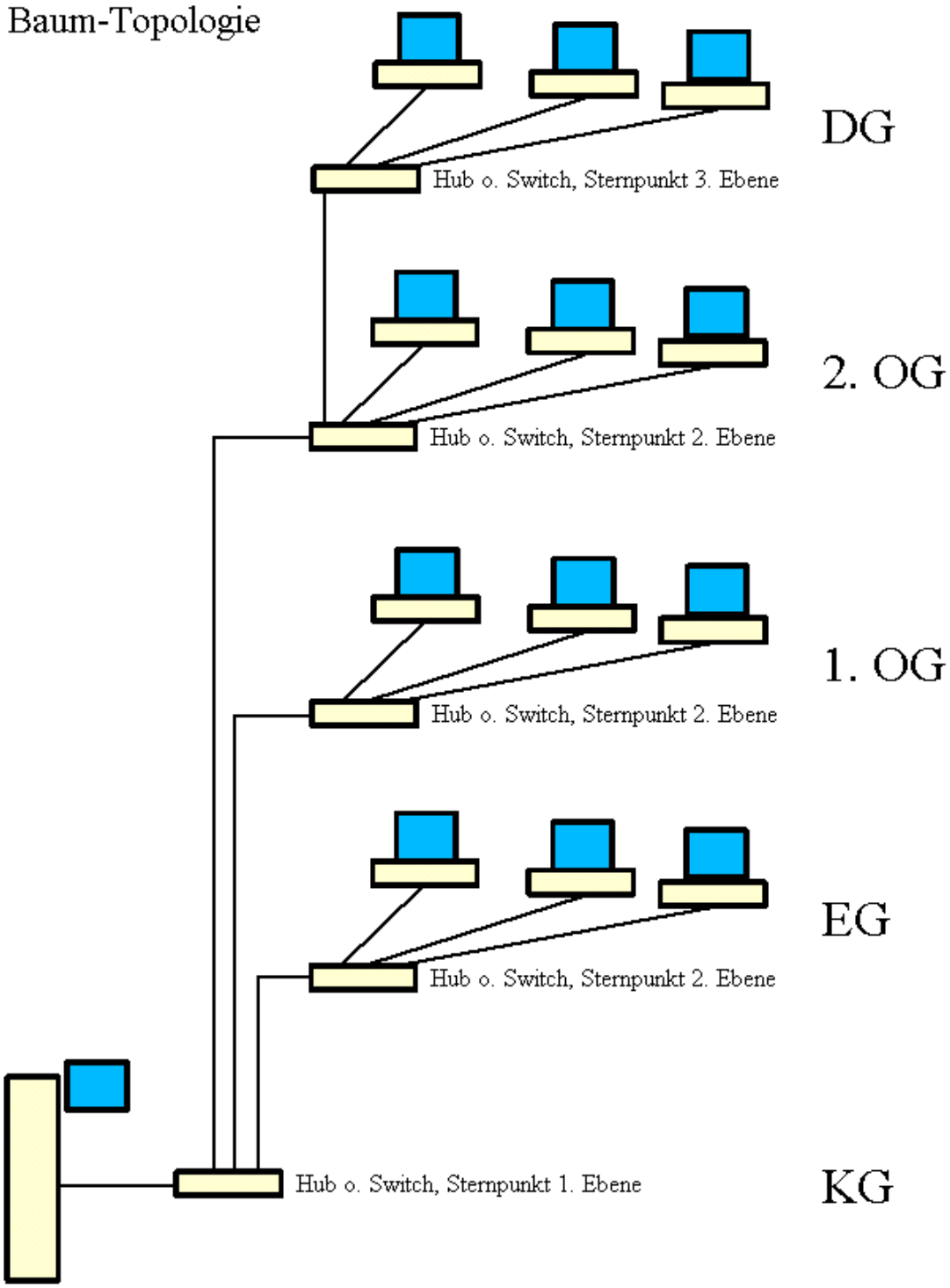
* 10BaseT mit einer Datenübertragungsrate von 10MBit/s min. CAT3
* 100BaseT mit einer Datenübertragungsrate von 100MBit/s min. CAT5
* 1000BaseT mit einer Datenübertragungsrate von 1GBit/s min. CAT5e

Zudem sind diese Kabel noch durch Kategorien klassifiziert. Höhere Kategorien erlauben höhere Datenübertragungsraten. Bis 1GBit/s kann aber immer noch Kabel der Kategorie 5 oder CAT5(e) eingesetzt werden. Allerdings werden dann 4 Adernpaare benötigt. Die Netzwerkverbindungen werden bei xBaseT mit RJ-45 Steckern und Buchsen hergestellt.



Die Baum-Topologie

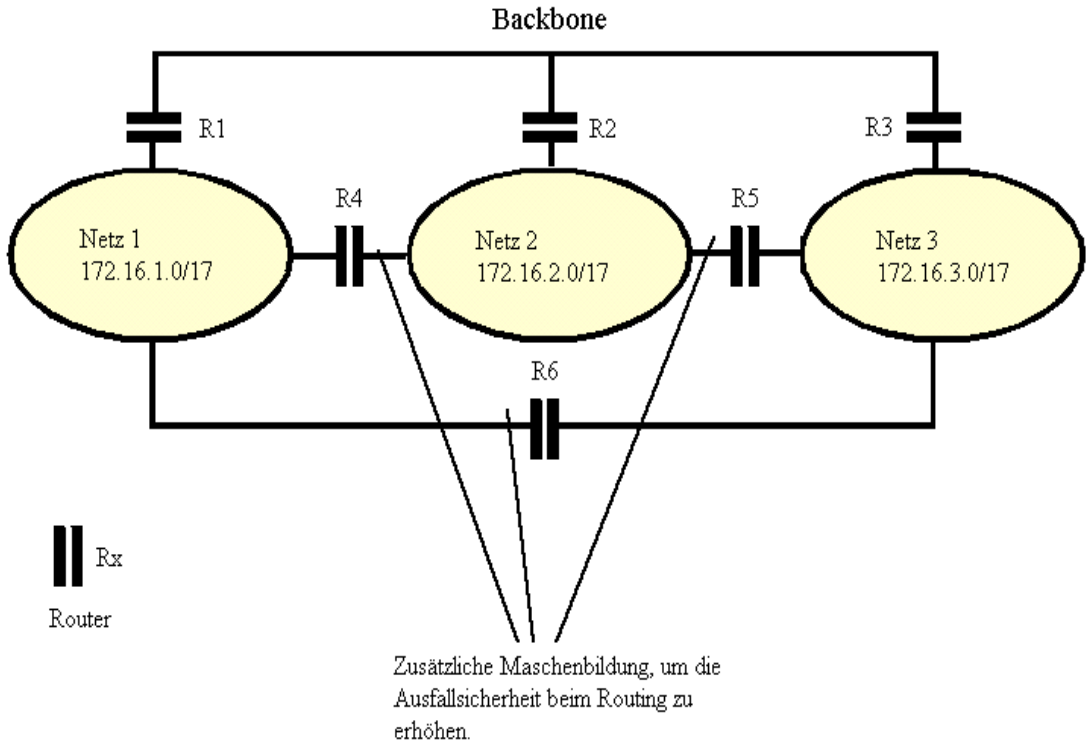
Wenn man ein grösseres Gebäude vernetzt ergibt sich eine Baumtopologie. Dies sind im Prinzip kaskadierende Sterne. Der erste Stern könnte z.B. vom Rechnerraum auf die einzelnen Geschosse führen. Auf jedem Geschoss bilden die Rechner dann einen weiteren Stern. Die Grundlagen für diese Verkabelung, man spricht auch von einer strukturierten Verkabelung, sind in der DIN EN 50173 festgelegt.



Die Maschen-Topologie

Diese findet bei der Vernetzung einzelner Computer keine Anwendung, sondern nur bei der Vermaschung mehrer Netze (Teilnetze, Subnetze). Mit Hilfe der Vermaschung können mehrere Routing-Wege erzeugt werden. Bei dem Ausfall einer Route kann dann eine andere verwendet werden. Das Internet ist ein vermaschtes Netz.

Maschen-Topologie



LAN – MAN – WAN

Den Begriffen LAN und WAN begegnet man ständig wenn es um Netzwerke geht, dem Begriff MAN eher seltener.

* LAN heist Local Area Network und bezeichnet ein Netzwerk welches auf einen Gebäudekomplex beschränkt ist. Dieser wird dann auch meist durch Grundstücksgrenzen festgelegt.
* WAN heist Wide Area Network und bezeichnet ein Netz welches grundstücküberschreitend ist, aber auch Ländergrenzen überschreiten kann.
* MAN heisst Metropolitan Area Network und bezeichnet ein Netz welches grundstückübergreifend ist, aber das alles in einer (Gross)stadt. Da es sich hier auch gleichzeitig um ein WAN handelt wird der Begriff eher selten verwendet.

Logische Topologien und Emulationen

Logische Topologien definieren wie die im Netz beteiligten Rechner das Netzwerk sehen. Oder anders gesagt: wie sind die Rechner aus der Sicht der Netzwerkkarte verkabelt. Dies kann aber immer nur ein Bus oder ein Ring sein. Denn dies sind die ursprünglichen Topologien. Das Ganze ist dann recht einfach gelöst. Die Geräte im Netz, welche für die Verbindung der Rechner zuständig sind, müssen die ursprünglichen Topologien emulieren.

* Ein Hub oder Switch verwandelt eine physikalische Stern-Struktur in eine Bus-Struktur. Dazu verbindet der Hub einfach alle angeschlossenen Rechner in einer Busstruktur. Der Switch ist intelligenter. Der Switch verbindet immer jeweils 2 kommunizierende Rechner und kann diese Verbindungen (Brücken) auch zwischen mehreren Rechnerpaaren erstellen und erstellt somit mehrere Busse. Wenn ein

weiterer Rechner auf so eine Verbindung zugreifen will, wird er abgewiesen, sodass die beiden Rechner ungestört kommunizieren können.
* Eine Multi Station Access Unit - MAU oder MSAU - verwandelt einen physikalischen Stern in einen logischen Ring.

Client-/Server Architektur

Kabel und Geräte, wie Hubs bzw. Switches, reichen natürlich nicht aus um ein Netzwerk zu bilden. Sie stellen nur die Übertragungswege zur Verfügung. Das eigentliche Netzwerk wird durch Software erzeugt. Und hier kommt die Client-/Server-Architektur (Abkürzung C/S) ins Spiel.

Client und Server treten bei der Netzwerkkommunikation immer im Paar auf. Wobei der Server Dienste und/oder Ressourcen zur Verfügung stellt und der Client diese anfordert. Mit anderen Worten:

* Auf dem Rechner der die Dienste bzw. Ressourcen zur Verfügung stellt arbeitet eine Server-Software. Diese Server-Software wird auch als Server oder als Netzwerkdienst bezeichnet.
* Auf dem Rechner der die Dienste bzw. Ressourcen anfordert arbeitet eine Client Software. Diese wird meist einfach nur Client genannt.

Wobei auf einem Rechner durchaus mehrere Server bzw. mehrere Clients arbeiten können und auf einem Rechner sowohl Server als auch Clients. Hier einige Beispiele für Netzwerkdienste:

* Dateidienste – Der Server stellt Dateien in einem Netzwerkdateisystem wie z.B. NFS oder Samba zur Verfügung.
* Druckdienste – Die Clients können über den Server auf Drucker zugreifen.
* DHCP (Dynamic Host Configuration Protocol) – Der DHCP-Server stellt IP-Adressen für die Clients im Netz zur Verfügung.
* DNS (Domain Name System) – Der Server verfügt über eine Datenbank welche die zu den Rechnernamen gehörenden IP-Adressen enthält und übermittelt diese bei Anfrage an die Clients.
* NTP (Network Time Protocol) – Der Server stellt den Clients auf Anfragte die Uhrzeit zur Verfügung.

Man unterscheidet ausserdem noch zwischen Peer-to-Peer Netzwerken und serverbasierten Netzwerken.

* Bei einem Peer-to-Peer Netzwerk können sowohl Server, als auch Clients auf jedem Rechner laufen. UNIX®, LINUX® und Microsoft® Windows® sind Peer-to-Peer Netzwerke.
* Bei einem serverbasierten Netzwerke ist ein Rechner entweder nur Server oder Client. Novell® Netware® ist eine reine Serversoftware. MS-DOS® kann nur Client sein, und dies auch nur durch spezielle Netzwerkkomponenten.

Merke! Ein Server ist immer eine Software, niemals eine Hardware!

Netzwerkprotokolle

Die oben beschriebenen Clients und Server kommunizieren dann auf der Grundlage von Netzwerkprotokollen miteinander. Diese Protokolle bestehen aus Regeln für den Nachrichten austausch. Ein einziges Protkoll reicht aber nicht aus. Vom Erzeugen der Nachricht, bis zu deren Versand über das Kabel, sind mehrere Prokolle beteiligt die jeweils eine Teilaufgabe übernehmen. Zu diesem Zweck sind die Prokolle in mehreren Schichten hierarchisch angeordnet. Jede dieser Schichten führt nur die für sie spezifischen Aufgaben aus und überlässt den Rest den Protokollen auf den anderen Schichten. Dies wird dann unten im Kapitel “OSI-Referenzmodell” genauer beschrieben.

Das Ethernet-Protokoll und die TCP/IP-Protokollfamilie haben jeweils ein eigenes Kapitel weiter unten.

RFC's (Requests For Comment)

In diesen Requests For Comment sind Standardisierungen für im Internet verwendete Verfahren und Protokolle festgehalten.

Der Link für alle RFC's ist "http://www.ietf.org/rfc.html" (Stand Dez. 2005). Die einzelnen RFC werden von mir unten nicht gesondert erwähnt. Einige RFC sind in der folgenden Liste enthalten.

RFC 0768	UDP - User Datagram Protocol
RFC 0791	IP - Internet Protocol
RFC 0792	ICMP - Internet Control Message Protocol
RFC 0793	TCP - Transmission Control Protocol
RFC 0826	ARP - Ethernet Address Resolution Protocol
RFC 0950	Internet Standard Subnetting Procedure
RFC 1011	Official Internet protocols
RFC 1112	IGMP - Host extensions for IP multicasting
RFC 1117	Internet numbers (IP-Adressen)
RFC 1118	Hitchhikers guide to the Internet
RFC 1122	Requirements for Internet Hosts - Communication Layers
RFC 1180	TCP/IP tutorial
RFC 1208	Glossary of networking terms
RFC 1517	CIDR - Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR)
RFC 1518	CIDR - An Architecture for IP Address Allocation with CIDR
RFC 1519	CIDR - Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
RFC 1597	Address Allocation for Private Internets (veraltet, ersetzt durch RFC 1918)
RFC 1883	Internet Protocol, Version 6 (IPv6) Specification (veraltet, ersetzt durch RFC 2460)
RFC 1884	IP Version 6 Addressing Architecture
RFC 1918	Address Allocation for Private Internets
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification

Das OSI-Referenzmodell (OSI-Schichtenmodell)

Der vollständige Name lautet ISO-OSI-Referenzmodell. Es wird aber meist als OSI-Schichtenmodell bezeichnet. Wobei hier ISO für "International Standards Organisation" und OSI für "Open Systems Interconnection" steht. Schichtenmodell deshalb, weil es aus 7 Schichten bzw. Ebenen besteht.

Das Modell gibt 7 Protokollebenen vor, die bestimmte Aufgaben erfüllen müssen, damit Rechner Daten miteinander tauschen können. Es ist jedoch nicht festgelegt wie es gemacht wird, sondern was gemacht wird. Die Schnittstellen zwischen den Schichten sind allerdings genau definiert. Somit kann der Inhalt einer Schicht jederzeit gegen eine andere Implimentierung bzw. ein anderes Protokoll ausgetauscht werden.

Jede Schicht bietet den angrenzenden Schichten definierte Dienste an. Beim Senden werden die Daten von den Protokollen der beteiligten Schichten jeweils zu den Protokollen der niedrigen Schichten nach unten durchgereicht. Beim Empfang dann, in umgehrter Reihenfolge, von unten nach oben durchgereicht. Wobei nicht immer alle Schichten beteiligt sind. Beim Routing z.B. sind nur Protokolle auf den untersten 3 Schichten beteiligt. Bei der Betrachtung der TCP/IP-Protokollfamilie weiter unten wird dies dann deutlicher.

Die 7 Schichten

Dies sind die 7 Schichten des Modells, jeweils in deutsch und englisch gegenübergestellt:

7	Anwendungsschicht	Application Layer
6	Darstellungsschicht	Presentation Layer
5	Sitzungs-/Kommunikationssteuerungsschicht	Session Layer
4	Transportschicht	Transport Layer
3	Vermittlungs-/Netzwerkschicht	Network Layer
2	Sicherungsschicht Logical Link Control Schicht (LLC) Media Access Control Schicht (MAC)	Data Link Layer
1	Bitübertragungsschicht	Physical Layer

1. Bitübertragungsschicht - Physical Layer

Diese Schicht ist zuständig für die Übertragung der Bits und Bytes über das Netzwerk. Die Protokolle dieser Schicht befassen sich mit den physikalischen und elektrischen Eigenschaften der Netzwerkkarte und des Übertragungsmediums (z.B. Kupferkabel oder Lichtwellenleiter). Hier findet die Modulation und Demodulation der Signale statt und die Netzwerkkarte sendet diese Signale über das Kabel an andere Teilnehmer im Netz.

2. Sicherungsschicht - Data Link Layer

Diese Schicht wird nochmals in 2 Schichten unterteilt.

Die untere Schicht heisst "Medium Access Control Schicht (MAC)" . Sie regelt den Zugriff auf das Übertragungsmedium, sprich die Netzwerkkarte. Zur Identifizierung des Mediums dient die MAC-Adresse, auch Hardwareadresse genannt.

* Die MAC-Adresse ist (im Idealfall) einmalig auf der Welt. Leider gibt es auch hin und

wieder Fälle wo dies nicht zutrifft.
* Sie besteht aus 6 Byte, die ersten 3 Byte bilden die Hersteller-ID, die letzten 3 Byte die Karten-ID. Die MAC-Adresse ist auf der Karte fest verdrahtet.
* Die MAC-Adresse steht im Datenpaket, damit auch der richtige Empfänger die Daten empfängt. Die 2. Schicht des Empfängers nimmt die Daten nur an wenn sie an seine MAC-Adresse gerichtet sind.

Die obere Schicht heisst "Logical Link Control (LLC)" . Diese hat dann die Aufgabe Datenübertragungsfehler aus der Physikalischen-Schicht zu entdecken und zu korrigieren. Diese Schicht soll eine fehlerfreie Kommunikation gewährleisten. Hier werden die Daten in sog. Frames (Rahmen) verpackt, bzw. es werden die Frames entpackt. Diese enthalten neben den Daten eine Prüfsummen sowie die Ziel- und Quelladresse der beteiligten Netzwerkkarten. Dies sind die MAC-Adressen! Mit Hilfe der Prüfsummen können fehlerhafte Informationen entdeckt und dann neu angefordert werden.

Die Aufgaben der Sicherungsschicht werden auch als MAC-Zugriff (MAC-Access) und Flusskontrolle (Flow Control) bezeichnet.

Ethernet (IEEE 802.3) ist ein Protokoll der Sicherungsschicht.

3. Vermittlungs-/Netzwerkschicht - Network Layer

Diese Schicht arbeitet nicht mit Hardwareadressen wie die 2. Schicht, sondern mit logischen Adressen. IP ist eines der Protokolle der Netzwerkschicht. Die Netzwerkschicht ist im Wesentlichen für den Transport der Pakete von einem Knotenpunkt zu einem anderen und für das Routing der Pakete verantwortlich. Die korrekte Übermittlung der Daten gehört allerdings nicht zu ihrer Aufgabe.

Ein weiteres Protokoll, welches dieser Schicht zugeordnet wird, ist IPX von Novell®.

4. Transportschicht - Transport Layer

Die Transportschicht ist für die korrekte Zustellung der Pakete zuständig. Da meist auf jedem Rechner mehrere Netzwerkverbindungen bestehen, müssen diese auch koordiniert werden. Auch diese Aufgabe übernimmt die Transportschicht.

Protokolle dieser Schicht sind:
* NetBEUI von Microsoft®
* SPX von Novell®
* TCP und UDP aus der TCP/IP-Protokollfamilie

5. Sitzungs-/Kommunikationssteuerungsschicht - Session Layer

Diese ist für die Dialogsteuerung zwischen 2. Stationen zuständig. Also z.B. für den Verbindungsaufbau und Verbindungsabbau und die Wiederaufnahme falls die Verbindung unterbrochen wurde. Die Modemverbindung gehört zur 5. Schicht, aber auch das Protokoll NetBIOS.

6. Darstellungsschicht - Presentation Layer

In dieser Schicht werden allgemeingültige im Netz verwendete Formate, z.B. für die Zeichen- oder Zahlendarstellung, in Formate übersetzt die der jeweilige Rechner versteht. Und der ganze Vorgang findet beim Senden von Daten natürlich umgekehrt statt. Beispiele:
* Formatwandlung EBCDIC / ASCII
* Datenkompression
* Kryptographie (z.B. SSL)

7. Anwendungungsschicht - Application Layer

Hierbei handelt es sich um netzwerkfähige Anwendungen die mit den Protokollen wie HTTP, FTP, TELNET, SMPT, etc. über das Netz miteinander Kommunizieren, dies ist z.B. der Teil des Browsers oder Mailprogrammes der die Schnittstelle zum Netz bildet.

Anwendungen der jeweiligen Schichten

Das oben gesagte klingt doch alles recht abstrakt. Deshalb hier ein paar Beispiele für Anwendungen der jeweiligen Schichten.

* Ein Hub arbeitet nur auf der 1. Schicht. Es findet eine reine Bitübertragung statt. Die Daten werden an alle Stationen im Netz gesendet, die an dem Hub angeschlossen sind.
* Switches (Bridges) arbeiten auf der 1. und 2. Schicht. Die Bitübertragung findet natürlich auch statt. Aber der Switch sendet die Daten nur an die MAC-Adresse für

  die die Daten bestimmt sind. Die MAC-Adresse ist jedoch Bestandteil der 2. Schicht. Der Switch baut eine Punkt zu Punkt Verbindung zwischen 2 Hosts bzw. Netzwerksegmenten auf.
* Ein Router wiederum arbeitet auf den Schichten 1, 2 und 3. Auch die Bitübertragung gehört zu seinen Aufgaben. Wenn der Router Pakete an ein Endgerät weiterreicht benötigt er dessen MAC-Adresse. Er muss also auch auf der 2. Schicht arbeiten können. Die Hauptaufgabe des Routers, nämlich die Vermittlung von Paketen an logische Adressen findet aber auf der dritten Ebene, der Vermittlungs- bzw. Netzwerkschicht statt.
* Das Protokoll IPSec arbeiten auf Schicht 3 der Netzwerkschicht und ermöglicht dadurch einen verschlüsselten Datentransport aller darüberliegenden Protokolle.
* Paketfilter-Firewalls arbeiten auf der Schicht 3 und 4 und filtern Daten anhand der Header-Informationen, die von den Protokollen IP, ICMP, TCP und UDP stammen.
* Proxy-Server, auch Application-Level-Firewalls genannt, arbeiten auf Schicht 7, der Anwendungsschicht und können dadurch auch Daten unabhängig vom verwendeten Port blockieren wenn diese zu einer bestimmten Anwendung gehören bzw. auf eine bestimmte URL verweisen.

Wofür ist das Ganze eigentlich gut?

Oft hört man die Meinung, das OSI-Schichtenmodell sei zwar eine ganz nette Theorie aber in der Praxis eigentlich überflüssig. Sozusagen ein nettes Gesprächsthema für gelangweilte Administratoren. Dies ist aber keineswegs der Fall.

* Das Verständnis des Modells hilft zum einen bei der Fehlersuche im Netzwerk.
* Wird benötigt bei Einbrüchen ins Netzwerk. Auf welcher Ebene fand der Einbruch statt, wie kann man dagegen vorgehen.
* Es wird immer wieder benötigt um Abläufe im Netzwerk zu erklären. Das Zusammenspiel von Protokollen, Anwendungen und der Hardware.

TCP/IP

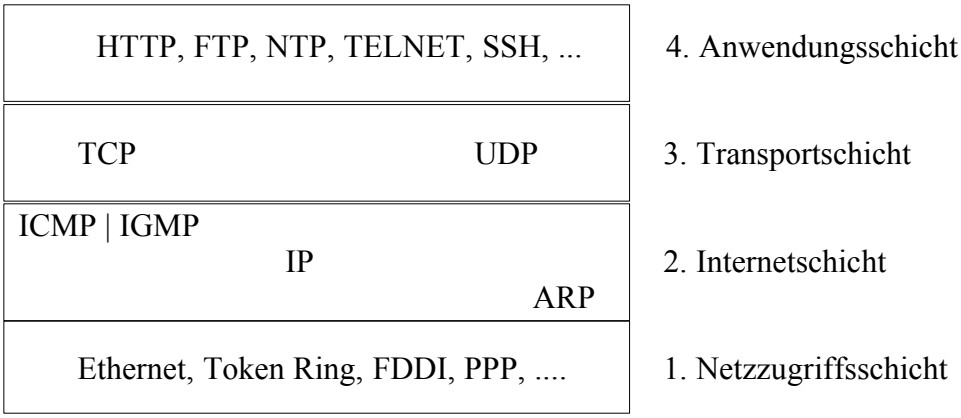
Bei TCP/IP handelt es sich um eine Protokollfamilie (Protokollsuite), nicht um ein einzelnes Protokoll. Und auch TCP/IP wird durch ein Schichtenmodell beschrieben. Es handelt sich allerdings nur um 4 Schichten. Dieses Referenz- oder Schichtenmodell ist älter als das OSI-Modell und diente für dieses als Vorlage.

Die Schichten von TCP/IP

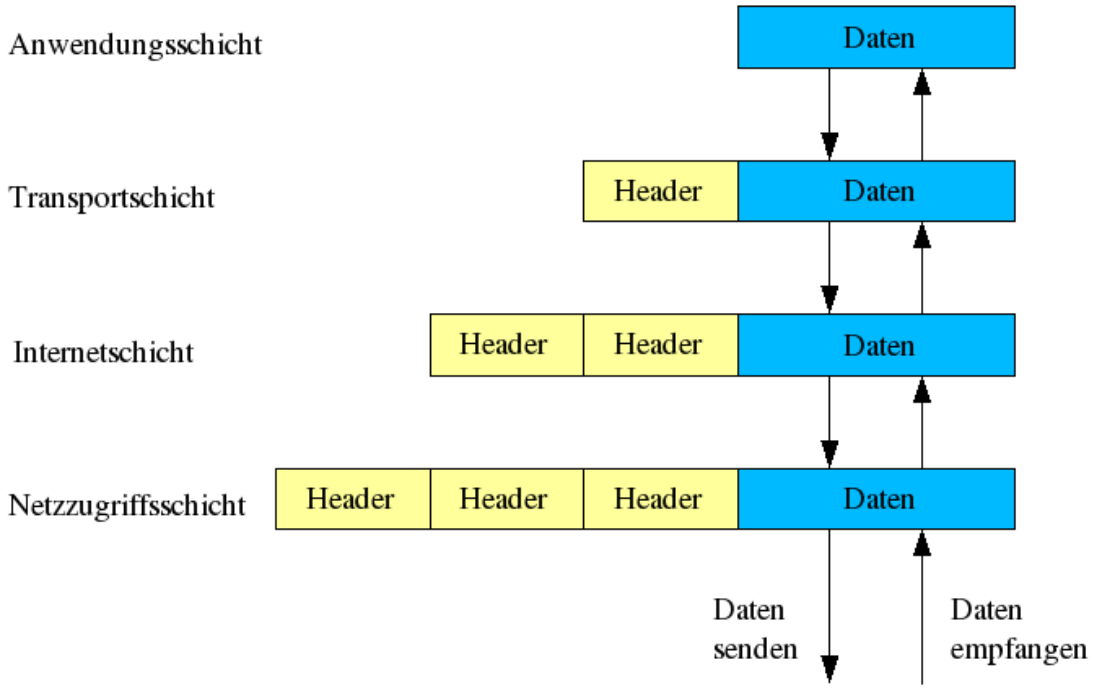
Es folgen div. Darstellungen dieser Schichten. Zuerst die Schichten mit deutscher und englischer Bezeichnung gegenübergestellt.

4	Application Layer	Anwendungsschicht
3	Transport Layer	Transportschicht
2	Internet Layer	Internetschicht
1	Network Layer	Netzzugriffs-/Netzwerkkartenschicht

Und hier jetzt die (unvollständige) Darstellung der Protokollfamilie:



Wenn nun Daten gesendet werden, werden diese Schicht für Schicht nach unten durchgereicht und dann über die Netzwerkhardware versendet.. Die höheren Schichten nehmen dabei die Dienste der darunterliegenden Schichten in Anspruch. Die Daten der Anwendungsschicht werden dabei in Pakete geteilt und erhalten dabei von jeder der darunterliegenden Schichten jeweils einen zu dieser Schicht gehörenden Header. Sie werden also quasi in jeder Schicht neu vergepackt. Beim Empfang werden die Daten auf dem umgekehrten Weg, Schicht für Schicht, bis zur Anwendungsschicht nach oben durchgereicht und wieder entsprechend entpackt.



Je nachdem auf welcher Schicht sich die Daten befinden, werden sie auch anders bezeichnet.

* Auf der obersten Schicht werden die Daten z. B. allgemein als Anwendungs-Nachricht (Appplication Message) (RFC 1180) bezeichnet. Manchmal aber auch, je nach dem ob die Anwendung TCP oder UDP für den Versand verwendet, Strom (Stream) im Falle von TCP und Nachricht (Message) im Falle von UDP.

* Die allgemeine Bezeichnung für die Schichten 1 – 3 ist "Pakete", dies ist aber nicht ganz korrekt. Die Namen für diese Pakete sind z.B. Segment, Datagramm, Rahmen (Frames), oder Paket, je nachdem zu welcher Schicht die Daten gehören. Wobei diese Bezeichnungen leider je nach Dokument bzw. Veröffentlichung unterschiedlich sein können. Die einzige Bezeichnung die überall übereinstimmt ist Rahmen bzw. Frames für die Pakete der Netzzugriffsschicht (Ethernet). Ich bezeichne die Daten auf der

  * Applikationsschicht mit Datenstrom (TCP) bzw. Nachricht (UDP).
  * Transportschicht mit Paket (UDP) bzw. Segment (TCP).
  * Internetschicht mit Datagramm.
  * Netzzugriffsschicht mit Rahmen.

  Wobei sich Datagramm für die Pakete der Internetschicht und Segment für die TCP-Packete der Transportschicht wohl allgemein durchgesetzt haben (s. hierzu auch RFC1122).

Die Netzzugriffs-/Netzwerkkartenschicht (Network Layer)

Die unterste Schicht ist zuständig für die Steuerung der Netzwerkhardware und greift dafür auf die notwendigen Protokolle wie IEEE 802.3., FDDI, PPP, etc. zu, ist aber auch gleichzeitig verantwortlich für die Bitübertragung und vereinigt damit, bezogen auf das OSI-Modell, zwei Schichten in einer. Diese eine fehlende Schicht wird auch allgemein als ein Mangel empfunden, da eigentlich jede Schicht einen spezifischen Aufgabenbereich haben sollte. Wie wir weiter unten noch sehen, wird TCP/IP deshalb auch manchmal mit 5 Schichten dargestellt. Wenn es dem besseren Verständnis dient, ist es auch kein Problem die beiden

Modelle zu mixen, bzw. für manche Betrachtungen das eine und für andere Betrachtungen dann das andere Modell zu verwenden.

Die Internetschicht (Internet Layer)

Die nächst höhere Schicht ist die Internetschicht mit den 4 Protokollen IP, ICMP, IGMP und ARP. Wobei ARP eine vermittelnde Funktion zur Netzzugriffsschicht hin einnimmt und ICMP und IGMP zur Transportschicht hin.

ARP (Address Resolution Protocol)

ARP löst IP-Adressen in MAC-Adressen (Hardware-Adressen) auf, wobei ich hier jetzt noch den Begriff Ethernet-Adresse einführe. Dies ist aber nichts anderes als die MAC-Adresse. Damit wird dann aber auch gleich der Bezug zum meistverwendeten Netzwerkprotokoll der ersten Schicht hergestellt.

Ethernet kennt keine IP-Adressen,sondern arbeitet mit den MAC-Adressen der Netzwerkkarten. Demzufolge können Daten an einen anderen Computer nur zugestellt werden, wenn dessen MAC-Adresse bekannt ist. Die TCP/IP-Datenpakete enthalten jedoch nur die IP-Adresse des Empfängers. Hier kommt ARP ins Spiel. Der Sender des Paketes versendet deshalb einen **ARP-Request** als Broadcast. Diese Anfrage enthält die Quell-MAC-Adresse, die Quell-IP-Adresse und die Ziel-IP-Adresse. Alle Computer die sich im gleichen Subnetz befinden und online sind erhalten diese Anfrage, aber nur der Host mit der Ziel-IP-Adresse sendet einen **ARP-Reply** an den Quell-Host. Der Quell-Host kann das Paket jetzt versenden.

Damit diese Anfragen nicht jedesmal neu versendet werden müssen, werden die Antworten für einen gewisse Zeit im **ARP-Cache** zwischengespeichert und zwar sowohl beim Quell- als auch beim Ziel-Host.

IP (Internet Protocol)

IP ist für die Adressierung und für das Routing von Paketen zwischen Hosts zuständig. IP garantiert aber nicht die korrekte Zustellung der Pakete. Dafür ist dann TCP in der höheren Schicht zuständig. IP selbst kümmert sich um jedes Paket einzeln. Dadurch kann es auch ohne weiteres vorkommen das die zu einer Session gehörenden Pakete unterschiedlich geroutet werden.

Das Thema IP ist sehr umfangreich. Man muss sich mit Begriffen bzw. Konzepten beschäftigen wie der IP-Adresse, Subnetze, Netzwerkadressen, Broadcastadressen, Netzwerkklassen, IPv4, IPv6, etc..

Die IP-Adresse

Mit Hilfe der IP-Adressen lassen sich die Rechner im Netz eindeutig identifzieren. Im gleichen Netzwerkbereich darf die IP-Adresse also nur einmal existieren. Im folgenden beschreibe ich nur IPv4, also die 4-Byte breite Variante. IPv6 wird später ein extra Kapitel gewidmet. Die IP-Adressen sind logische Adressen. Jede Adresse wird durch 32 Bit bzw. einer 4 Byte breite Zahl dargestellt. Diese kann entweder hexadezimal, binär oder dezimal dargestellt werden. Bei der Darstellung werden die 4 Byte meist durch Punkte getrennt. Es gibt verschiedene Bezeichnungen für diese Bytes, ich bezeichne sie als Oktett.

Hier die dezimale und binäre Darstellung der gleichen IP-Adresse:

dezimal	192.168.1.100
binär	11000000.101010000.00000001.01100100

Wg. der Lesbarkeit wird natürlich meist die dezimale Darstellung verwendet. Für Berechnungen benötigt man jedoch die binäre Darstellung.

Subnetze

Das Internet mit seinen 2^32 IP-Adressen ist in viele kleinere Netzwerkbereiche unterteilt, die sog. Subnetze (sub nets). Ohne dies wäre es wohl auch nicht administrierbar. Um dieses Subnetz-Konzept zu realisieren hat jede IP-Adresse einen **Netzanteil** und einen **Hostanteil**.
Der **Netzanteil** (bzw. die **Netzwerkkennung** oder **Netz-ID)** identifizieren das betreffende Netzwerk, der **Hostanteil** (bzw. die **Hostkennung** oder **Host-ID)** identifizieren den Host in dem betreffenden Netz. Diese Anteile werden über die **Netzmaske** (auch **Subnetzmaske** genannt) festgelegt. Alle Teile der Netzmaske die mit 1 in der binären Darstellung belegt sind, stellen den Netzanteil dar, die Nullen den Hostanteil. Die Einsen in der Netzmaske müssen durchgehend sein. Durch eine UND-Verknüpfung der Netzmaske und der IP-Adresse kann man dann die Netzadresse ermitteln.

Hier erst mal ein kleines Beispiel:

Netzmaske	11111111.11111111.11111111.00000000	255.255.255.0
IP-Adresse	11000000.10101000.00000001.01100100	192.168.1.100
UND		
Netzadresse	11000000.10101000.00000001.00000000	192.168.1.0

Wir haben also die IP-Adresse 192.168.1.100 mit der Netzmaske 255.255.255.0 und erhalten durch Anwendung eines Bitweisen UND die Netzaddresse 192.168.1.0

Damit dies verständlich wird müssen noch einige Begriffe erklärt werden. Ein Subnetz, Teilnetz oder einfach Netzwerk genannt wird durch seine **Netzadresse** identifiziert.

* Dies ist die niedrigste Adresse die in diesem Subnetz vorhanden ist.
* Die Netzadresse darf nicht an einen Host vergeben werden.

Der Umfang des Netzwerkes wird wiederum durch die **Subnetzmaske** bestimmt. Mit ihrer Hilfe kann man die Anzahl der möglichen Hosts in einem Netzwerk ermitteln. Wenn man die Netzmaske invertiert erhält man die Anzahl der zur Verfügung stehenden IP-Adressen im betreffenden Netzwerk:

Netzmaske	11111111.11111111.11111111.00000000	255.255.255.0
invertiert	00000000.00000000.00000000.11111111	0.0.0.255

In diesem Fall, mit einer Subnetzmaske von 255.255.255.0, umfasst das Netz 2^8 = 256 IP-Adressen. Davon dürfen 2 Addressen nicht verwendet werden: die niedrigste, dies ist die **Netzadresse** und die höchste, dies ist die **Braodcast-Adresse** (BC-Adresse). In unserem Fall, auf das obige Beispiel bezogen:

Netzadresse	192.168.1.0
BC-Adresse	192.168.1.255

Es bleiben also 254 IP-Adressen übrig, die an Hosts vergeben werden können: 192.169.1.1 bis 192.168.1.254.

Merke!
Bei der Vergabe von Hostadressen dürfen nicht alle Bits der Hostkennung auf 0 oder 1 gesetzt sein.

Damit haben wir jetzt beschrieben wie solch ein Subnetz definiert wird. Aber was sind die Konsequenzen daraus? Nur Computer, die sich im gleichen Netzwerk befinden, können sich ohne weitere Hilfsmittel gegenseitig "sehen", also Daten miteinander austauschen.

Zusammenfassung:
* Jede IP-Adresse hat einen Netz- und einen Hostanteil.
* Der Netzanteil identifiziert das Netzwerk, der Hostanteil den Host.
* Netz- und Hostanteil werden durch die Netzmaske bestimmt.
* Alle Teile der Netzmaske die mit 1 in der binären Darstellung belegt sind, stellen den Netzanteil dar, die Nullen den Hostanteil.
* Die Einsen in der Netzmaske müssen durchgehend sein.
* Durch eine UND-Verknüpfung der Netzmaske und der IP-Adresse wird die Netzadresse ermittelt.
* Die niedrigste Adresse im Netz ist die Netzadresse, diese darf nicht an einen Host vergeben werden.
* Die höchste Adresse im Netz ist die BC-Adresse. Auch sie darf nicht an einen Host vergeben werden.
* Die Anzahl der Hosts in einem Netz ist (2^Anzahl_der_Host-Bits) – 2.

Routing

Oben habe ich gesagt, dass sich einzelne Subnetze ohne weitere Hilfsmittel nicht gegenseitig wahrnehmen können. Diese Schwelle wird mit Hilfe des Routing überwunden. Beim Routing verwendet man im Prinzip einen Rechner mit 2 Netzwerkkarten. Jede dieser Netzwerkkarten befindet sich in einem der beiden Teilnetze, zwischen denen geroutet werden soll. Auf diesem Rechner laufen dann noch Routinen, die die Daten von dem einen in das andere Subnetz weiterreichen. Dies ist eine stark vereinfachte, im Moment jedoch hinreichende Erklärung.

Stand der Technik sind dedizierte Router, also Rechner die nur diesen Zweck erfüllen. Diese werden im 19" Gehäuse geliefert, sind mit mehreren Schnittstellen ausgerüstet und über herstellereigene Software-Interfaces administrierbar.

Netzwerk-Klassen

Die IP-Adressen sind in verschiedene Klassen unterteilt. Jede dieser Klassen beinhaltet unterschiedliche Netzwerkgrössen. Da im Internet im allgemeinen keine einzelnen IP-Adressen vergeben werden, sondern Netzwerk-Adressen die einen bestimmten Bereich umfassen, können je nach Bedarf Netzwerke unterschiedlicher Grösse vergeben werden.

Klasse	Netzanteil	Hostanteil	von	bis	Netzmaske
A	1 Byte	3 Byte	1.0.0.0	126.255.255.255	255.0.0.0
B	2 Byte	2 Byte	128.0.0.0	191.255.255.255	255.255.0.0
C	3 Byte	1 Byte	192.0.0.0	223.255.255.255	255.255.255.0

Klasse	Bitweise Darstellung (N=Netzbit, H=Hostbit)	Merkmale

A	0NNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH	1. Bit = 0
B	10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH	1. Bit = 1 2. Bit = 0
C	110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH	1. Bit = 1 2. Bit = 1 3. Bit = 0

(Klasse D Multicast)
D 1110HHHH.HHHHHHHH.HHHHHHHH.HHHHHHHH

(Klasse E Reserviert für zukünftige Benutzung)
E 1111HHHH.HHHHHHHH.HHHHHHHH.HHHHHHHH

Loopback 01111111.HHHHHHHH.HHHHHHHH.HHHHHHHH

Klasse A:
* Der Netzanteil umfasst 1 Byte, der Hostanteil 3 Byte.
* Das 1. Bit der Adresse ist 0, damit ist die höchste Netz-ID 127. Da die 127 jedoch für die Loopbackadressen reserviert ist, ist die höchste Netzwerk-ID 126.
* Damit ergeben sich 126 Klasse A Netze mit jeweils bis zu (2^24) – 2 Hosts.

Klasse B:
* Der Netzanteil umfasst 2 Byte, der Hostanteil 2 Byte.
* Die ersten beiden Bits der Adresse sind 10. Dadurch ist die niedrigste Netz-ID 128.0 und die höchste Netz-ID 191.255.
* Damit ergeben sich 2^14 Klasse B Netze mit jeweils bis zu (2^16) – 2 Hosts.

Klasse C:
* Der Netzanteil umfasst 3 Byte, der Hostanteil 1 Byte.
* Die ersten drei Bit der Adresse sind 110. Dadurch ist die niedrigste Netz-ID 192.0.0 und die höchste Netz-ID 223.255.255.
* Damit ergeben sich 2^21 Klasse C Netze mit jeweils bis zu (2^8) – 2 Hosts.

Klasse D:
* Bei den Multicastadressen ist der Bereich des ersten Byte 224 – 239. Mit Hilfe von Multicastadressen werden Datagramme an mehrere Hosts gesendet, die zu einer Multicastgruppe gehören.

Loopback:
* Die Loopbackadressen 127.0.0.0 bis 127.255.255.255 sind nur für die lokale Verwendung auf dem betreffenden Computer zugelassen und im Netz nicht zulässig.

  Das loopback device "lo" ist eine interne Schnittstelle und wird für das Funktionieren der TCP/IP-Schnittstelle benötigt. Viele OS-Funktionen benötigen ein Netzwerk auch wenn kein physikalisches vorhanden ist, z.B. beim Drucken. lo ist aber immer vorhanden und kann deshalb hierfür verwendet werden.

Private/nicht öffentliche Adressbereiche

Lokale Netze, die im Internet nicht gesehen werden, benötigen keine einmalige, unverwechselbare Adresse. Deshalb wurden für diese Netze spezielle Adressbereiche reserviert. Adressen aus diesen Bereichen dürfen öffentlich nicht verwendet werden und werden im Internet auch nicht geroutet.

Dies sind die folgenden Adressbereiche:

Netzwerk-Klasse	von	bis	Netzmaske
Klasse A	10.0.0.0	10.255.255.255	255.0.0.0
Klasse B	172.16.0.0	172.31.255.255	255.255.0.0
Klasse C	192.168.0.0	192.168.255.255	255.255.255.0

CIDR – Classless Internet Domain Routing

Bei der Vergabe von Netzwerkadressen auf Grundlage der Netzwerk-Klassen wird recht verschwenderisch mit dieser Ressource verfahren. Die Netzwerkgrösse ist durch dieses Schema auch ziemlich starr festgelegt. Durch CIDR wurde dies aufgelöst. Die Netzmaske kann jetzt 8 Bit bis 32 Bit betragen. Bedingt dadurch können die Netzwerkbereiche feiner Verteilt werden, da ja jetzt ein Klasse A, B oder C Netz in weitere Subnetze unterteilt werden kann. Parallel dazu hat es sich auch eingebürgert die Netzwerkmaske durch die Anzahl der 1-ser Netz-Bits darzustellen. Diese Technik spielt sowohl bei der Vergabe von öffentlichen Adressen eine Rolle, als auch bei der Aufteilung von privaten, nicht öffentlichen Netzwerken in mehrere Subnetze.

Eine Klasse A Netz hat 8 Netz-Bits, also auf 1 gesetzte Bits und die Netzmaske kann dann entweder so "255.0.0.0" oder so "/8" (sprich: Slash 8) dargestellt werden. Klasse B dann so "255.255.0.0" oder so "/16", etc.. Und alle Zwischenwerte können auch verwendet werden. Hier ein paar Beispiele:

Netzmaske klassisch	Darstellung CIDR
255.0.0.0	/8
255.128.0.0	/9
255.192.0.0	/10
255.255.224.0	/19
255.255.240.0	/20
255.255.255.248	/29
255.255.255.252	/30

Mit Hilfe der Netzmaske kann man jetzt Subnetze bilden, die der Anzahl der zur Verfügung stehenden Hosts nahe kommen. Als Beispiel ein Klasse C-Netz mit der Adresse 192.168.1.0/24. Dieses soll in 2 Netze aufgeteilt werden:

Netz-Nr.	Netzadresse	BC-Adresse
Netz 1	192.168.1.0/25	192.168.1.127
Netz 2	192.168.1.128/25	192.168.1.255

Beide Netze umfassen 128 IP-Adressen. Es können jeweils 126 Adressen an Hosts vergeben werden. Von dieser Aufteilung sind dann auch die Netzwerk- und BC-Adressen betroffen. Aber wie bei den durch Klassen festgelegten Netzen, handelt es sich hier auch um die jeweils niedrigste bzw. höchste Adresse im Netz.

IPv6

Da die zur Verfügung stehenden IP-Adressen langsam knapp werden, wird eine Erweiterung des IP-Protokolls benötigt. Dies ist IPv6 (IP Version 6) auch IPng (IP next generation) genannt. Das oben beschriebene IP-Protokoll trägt die Versionsnummer 4 (IPv4).

Das zuerst offensichtliche Merkmal von IPv6 ist die breitere IP-Adresse. Diese ist jetzt 128 Bit breit. Es stehen jetzt also, rein theoretisch bis zu 2^128 Adressen zur Verfügung. Umgerechnet als Dezimalzahl ist dies 3,4^38!.

* Dies ist jedoch nicht die einzige Änderung. Im Prinzip ist IPv6 ein neues Protokoll. Das Header-Format wurde geändert. Es gibt einen Basis- Header gefolgt von einem oder mehreren Erweiterungs-Headern. Diese Form des Headers soll auch die Erweiterkeit von IPv6 garantieren. Denn das Protokoll kann bei Bedarf um weitere Erweiterungs-Header ergänzt werden.
* Diese Erweiterungs-Haeder können u.a. für die Authentifizierung und Verschlüsselung von Paketen verwendet werden.
* Mit der neuen Quality-of-Service Eigenschaft können Pakete von Diensten, die einen konstanten Datenstrom benötigen, wie z.B. Video-/Audiodaten oder IP-Telefondaten, vorrangig behandelt werden.

Nur um einige der Änderungen zu erwähnen. IPv6 ist auch schon in vielen Anwendungen implimentiert, ist im Internet aber noch kaum präsent. Dies wird sich aber wahrsscheinlich in naher Zukunft ändern.

IPv6-Adressierung

Die IPv6-Adresse wird durch 8 16-Bit Werte in hexadezimaler Darstellung, getrennt durch Doppelpunkte, dargestellt. Also etwa so:

```
FDDC:BA88:6655:3310:FDDC:BAA8:7666:3234
```

Gruppen von Null-Werten können mit dem Ausdruck :: zusammengefasst werden. Dieser Ausdruck darf nur 1x in der Adresse vorkommen!

FF01:0:0:0:0:0:0:45	wird zu	FF01::45
0:0:0:0:0:0:0:1	wird zu	::1

In einer gemischten IPv4 und IPv6 Umgebung kann man auch eine Mixform aus neuer und alter Schreibweise verwenden:

x:x:x:x:x:x:d.d.d.d

Wobei die “x” 6 hexadezimale Gruppen der neuen Form repräsentieren und die “**d**” die 4 Oktette der alten Form, in dezimaler Schreibweise. Also so:

```
0:0:0:0:0:0:192.168.1.1
```

Oder in der abgekürzten Schreibweise so:

```
::192.168.1.0
```

ICMP (Internet Controll Message Protocol)

ICMP dient dem Austausch von Status und Fehlerinformationen zwischen Hosts. Das Protokoll ist Bestandteil von IP und stellt quasi ein Hilfsprotokoll für IP dar. Es arbeitet mit verschiedenen Nachrichtentypen. Einigen von diesen besitzen auch noch ein Code-Feld um die Nachricht genauer zu spezifizieren. Der wohl bekannteste Befehl, der ICMP verwendet, ist Ping. Die ICMP-Nachrichten können sowohl eingehend als auch ausgehend sein. Wenn ein Host einen Ping versendet (Quellhost) erhält er einen Pong als Antwort von dem Host der angepingt wurde (Zielhost). Der Quellhost versendet also einen Ping (ausgehend) und empfängt einen Pong (eingehend). Für den Zielhost ist es genau umgekehrt. Er empfängt einen Ping (eingehend) und versendet einen Pong (ausgehend).

Typ	***Bezeichnung***	***Beschreibung***
8	Echo Request (Ping)	Die Echoanforderung wird von dem Befehl Ping verwendet. Mit Hilfe dieses Befehls wird ermittelt ob ein Host über die angefragte IP-Adresse erreichbar ist.
0	Echo Reply (Pong)	Dies ist die Antwort auf den Ping. Die Antwort erfolgt wenn der Ziel-Host auf dieser IP-Adresse erreichbar ist. Da heute immer öfter der Echo Reply von Firewalls abgeblockt wird, sagt dies jedoch nicht unbedingt etwas über die Erreichbarkeit des Rechners aus. Dieser kann sehr wohl noch, z.B. auf Port 80 (HTTP), erreichbar sein.
3	Destination Unreachable	Diese Nachricht kommt zurück wenn ein Router feststellt, dass der Host bzw. das Netz nicht existiert oder ausser Reichweite liegt. Ein Host sendet diese Antwort wenn er auf dem angeforderten Protokoll oder Port nicht ansprechbar ist.
4	Source Quench	Wenn auf einem der Router die Kapazität des Puffers nicht ausreicht um die Pakete an den nächsten Router, bzw. den Host weiterzureichen, wird diese Nachricht an den Quell-Host gesendet. Dies ist eine Aufforderung die Datenübertragungsrate zu senken.
5	Redirect	Mit Hilfe von Redirect können alternative Routeninformationen an einen Host gesendet werden.
11	Time Exceeded	Wenn die ttl (Time To Live) überschritten ist, wird das Packet vom Router verworfen und diese Nachricht an den Quellhost gesendet.
12	Parameter Problem	Wenn das Paket wegen Problemen mit den Header-Parametern nicht verarbeitet werden kann, wird diese Nachricht an den Quellhost gesendet.

ICMP gilt allgemein als Sicherheitsrisiko. In dem Dokument "Firewall mit iptables" habe ich eine Gefahrenanalyse für ICMP erstellt, jeweils mit einer Empfehlung welcher Nachrichtentyp eingehend bzw. ausgehend verwendet werden kann bzw. blockiert werden sollte.

IGMP (Internet Group Message Protocol)

IGMP ist ein weiteres Hilfsprotokoll für IP. Es dient dem Multicasting. Beim Multicasting werden die Nachrichten nicht wie beim Braodcasting an alle Rechner in einem Netzwerksegment verschickt, sondern nur an Rechner, die zu einer Multicastgruppe gehören. Dies soll den Netzwerkverkehr reduzieren, denn Braodcasting erzeugt viel Verkehr, da die Nachrichten an alle Rechner gerichtet sind.

Bei den Multicastadressen handelt es sich um Klasse D Adressen mit dem Bereich 224.0.0.0 bis 239.255.255.255.

Die Protokolle OSPF (Open Shortest Path First – ein Routingprotokoll) und NTP (Network Time Protocol) arbeiten mit Multicasting, aber auch Anwendungen für Videokonferenzen, für die Softwareverteilung und zum Versenden von News.

Die Transportschicht (Transport Layer)

Über der Internet-Schicht liegt die Transportschicht mit den Protkollen TCP und UDP. TCP ist verbindungsorientiert, UDP verbindungslos.

* Verbindungsorientiert (TCP)
  Zwischen zwei Hosts wird einen feste Verbindung aufgebaut. Die Verbindung bleibt solange bestehen, bis alle Daten übertragen sind. Dies ist für grosse Datenmengen geeignet. Die an der Übertragung beteiligten Programme sind die ganze Zeit aktiv, bis die Übertragung abgeschlossen ist.

* Verbindungslos (UDP)
  Diese Art der Datenübertragung wird für kleine Datenmengen verwendet, wie NTP, DNS-Anfragen, Broadcast etc.. Die beteiligten Programme kommunizieren nicht permanent.
  Beispiel: der Client sendet eine Anfrage an den Server. Wird die Anfrage nicht beantwortet sendet der Client die Anfrage, in der Regel nach einer bestimmten Zeit bzw. in Intervallen, erneut bis er eine Antwort erhält. Wenn keine Anwort eintrifft wird nach einer festgelegten Zeit eine Timeout-Meldung produziert. Der Server sendet die Antwort ohne sich darum zu kümmern ob diese auch den anfragenden Host erreicht. Protokolle wie DNS und NTP enthalten deshalb meist mehrere Server für die Anfragen in Ihrer Konfiguration für den Fall, dass einer der Server ausfällt.

* Das Protokoll aus der Anwendungsschicht bestimmt ob es sich um eine verbindungslose oder verbindungsorientierte Übertragung handelt.
  - FTP, HTTP z.B. sind verbindungsorientiert.
  - DNS, NTP z.B. sind verbindungslos.

TCP (Transmission Control Protocol)

Verbindung – TCP-Flags

Das Protokoll TCP (Transmission Control Protocol) ist verbindungsorientiert. D.h. es muss erst eine feste Verbindung bestehen, bevor die Übertragung von Daten beginnen kann. Für den Verbindsaufbau, die Datenübertragung und beim Verbindungsabbau werden die TCP-Flags verwendet. Hierfür gibt es das TCP-Flags-Feld im Header des TCP-Protokolls. Dieses ist 6 Bit breit und kann damit 6 1-Bit breite Flags aufnehmen.

URG - Urgent Pointer field significant

Gibt an, dass der Urgent Pointer, ein Feld des TCP-Protokoll Headers, verwendet wird. Der Urgent Pointer verweist auf Daten im Datenstrom, die Vorrang haben.

ACK - Acknowledgement field significant

Das ACK-Flag dient der Bestätigung von Flags, wie dem SYN- oder FIN-Flag und wird während der Datenübertragung verwendet.

PSH - Push Function

Entleert den Datenpuffer und gibt die Daten sofort an die Applikation weiter.

RST - Reset the connection

Mit dem RST-Flag kann eine Verbindung zurückgesetzt werden wenn ein Fehler bei der Datenübertragung aufgetreten ist. Es kann auch verwendet werden um eine Verbindung abzuweisen.

SYN - Synchronize sequence numbers

Das SYN-Flag dient der Initialisierung des Verbindungsaufbau's. Dieser findet in der Form eines Dreiwege-Handshake statt (s.u.), bei dem auch das ACK-Bit beteiligt ist. Das vom Client gesendete SYN-Flag ist eine Anfrage an den Server, die Sequenz-Nummern zu synchronisieren.

FIN - No more data from sender

Das FIN-Bit beendet die Datenübertragung. Dies geschieht dann auch wieder mit einem Handshake (s.u.).

Achtung!
Das FIN-Flag darf niemals alleine vorhanden sein, sondern immer nur zusammen mit dem ACK-Flag. Sollte das FIN-Flag alleine vorhanden sein, handelt es sich um ein bösartiges Paket.

Handshakes

Zum Verbindungsaufbau und Verbingsabbau werden sog. Handshakes verwendet.

Der Dreiwege-Handshake für den Verbindungsaufbau

Dieser Handshake wird vom Client eingeleitet, indem dieser ein SYN-Flag an den Server sendet. Der Server bestätigt dies im 2. Schritt mit der Kombination aus einem SYN- und einem ACK-Flag. Im 3. und letzten Schritt antwortet der Client nur mit einem gesetzten ACK-Flag. Damit ist der Verbindungsaufbau abgeschlossen und von jetzt an wird immer das ACK-Flag verwendet. Wobei das ACK-Flag von jetzt ab bei jedem Paket gesetzt sein muss, welches Bestandteil der Verbindung ist!

```
    Verbindsaufbau

    Client                                        Server

1.    -------------------- SYN ---------------------->
2.    <------------------ SYN/ACK --------------------
3.    -------------------- ACK ---------------------->
.
    Bestehende Verbindung
.
N     -------------------- ACK ---------------------->
N+1   <------------------- ACK ----------------------
N+2   -------------------- ACK ---------------------->
N+3   <------------------- ACK ----------------------
```

Der Vierwege-Handshake für den Verbindungsabbau

Da es sich bei TCP um eine Zweiwegeverbindung handelt, müssen auch beide

Verbindungsrichtungen geschlossen werden. Beim Verbindungsabbau sendet der Client eine Kombination aus FIN- und ACK-Flag an den Server. Der Server antwortet mit einem ACK-Flag. Anschliessend sendet der Server die FIN/ACK-Kombination und der Client antwortet mit einem ACK-Flag.

```
    Verbindsabbau

    Client                                          Server

1.    ------------------- FIN/ACK -------------------->
2.    <-------------------- ACK ----------------------
3.    <------------------- FIN/ACK ---------------------
4.    ---------------------- ACK ---------------------->
```

Übertragungssicherheit

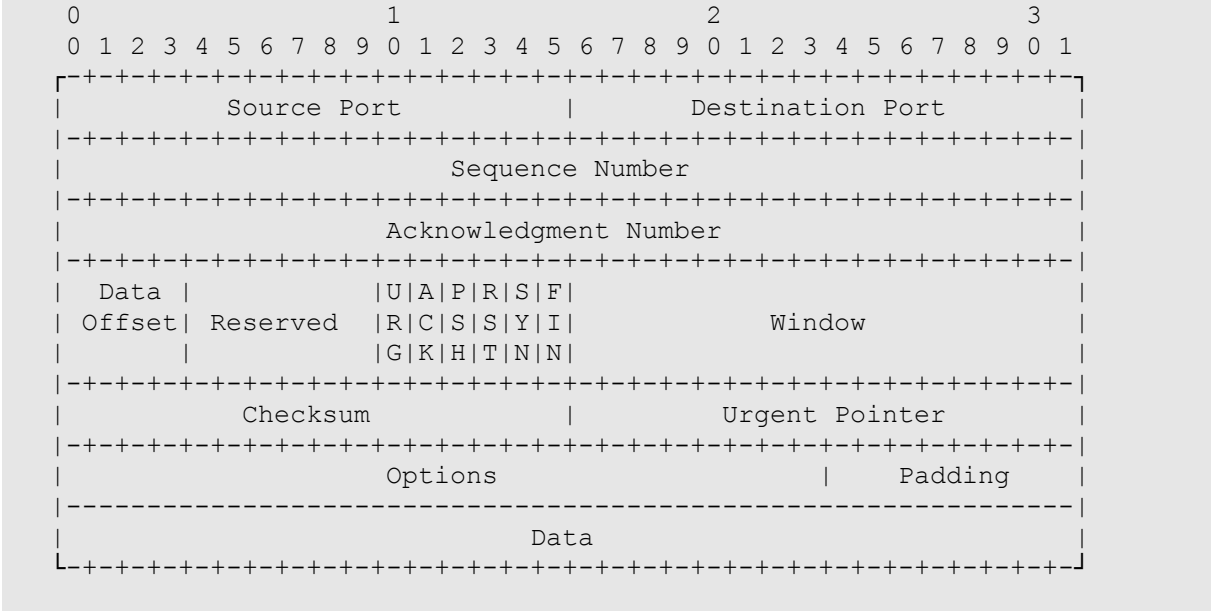
TCP ist für die sichere Übertragung der Daten verantwortlich. Wenn der Empfang eines Paketes nicht vom Empfänger bestätigt (quittiert) wird, wird es noch einmal verschickt. Mit Hilfe von Sequenz- und Bestätigungsnummern werden die Pakete identifiziert. Anhand einer Prüfsumme wird die Integrität des Paketes überprüft. Wenn ein Paket nicht qittiert wird, wird es nach einem Timeout nochmals gesendet.

Kommunikation mit der Anwendungsschicht

Anhand der Portnummern werden die Pakete an die entsprechende Anwendung wie HTTP, FTP, TELNET, etc. in der Anwendungsschicht weitergeleitet.

Der TCP-Header

Auch andere Protokolle haben einen Header. Es ist jedoch nicht meine Absicht alle Header in diesem Dokument darzustellen. Für die anderen Header verweise ich auf die entsprechenden RFC's.



Aus der oben gewählten Darstellung ist die jeweilige Bit-Breite der Felder zu ersehen. Hier die Bedeutung der Felder im einzelnen:

Source Port – Quell Port – 16 Bit

Der von der sendenden Anwendung verwendete Port.

Destination Port – Ziel Port – 16 Bit

Der von der Zielanwendung verwendete Port.

Sequence Number – Sequenznummer – 32 Bit

Die Sequenznummer gibt die Reihenfolge der Pakete wieder.

Acknowledgement Number – Bestätigungsnummer – 32 Bit

Dient zur Bestätigung der empfangenen Pakete. Mit dieser gibt der Host an welches Segment als nächstes erwartet wird. Er erhöht die Nummer des zuletzt empfangenen Segments um 1.

Data Offset – Daten Versatz – 4 Bit

Die Grösse des TCP-Headers in 32 Bit Worten. Diese Angabe ist notwendig, da das Optionen-Feld eine variable Breite hat. Über diesen Wert wird ermittelt, wo das Daten-Feld beginnt. Der Header hat immer ein vielfaches von 32 Bit.

Reserved – Reserviert – 6 Bit

Reserviert für zukünftige Anwendungen. Wird z. Zt. nicht verwendet.

URG, ACK, PSH, RST, SYN, FIN – jeweils 1 Bit

Die TCP-Flags, Erklärung s.o..

Window – Fenster – 16 Bit

Hiermit gibt der Empfänger an, wieviele Bytes er aktzeptiert ohne vorher eine Bestätigung gesendet zu haben (Puffergrösse).

Checksum – Prüfsumme – 16 Bit

Mit Hilfe der Prüfsumme wird die Integrität der Daten in dem Segent überprüft.

Urgent Pointer – Dringlichkeitsanzeige – 16 Bit

Weisst auf dringliche Daten hin, die sofort gelesen werden sollten. Muss immer zusammen mit dem URG-Flag verwendet werden.

Options – Optionen – variable Breite
Stellt div. Zusatzoptionen zur Verfügung, u.a. die Option "Maximum Segment Size", also Maximale-Segment-Grösse. Diese wird von den Hosts während des Verbindungsaufbaus ausgehandelt. Diese Option wird nur bei Paketen mit gesetztem SYN-Flag gesetzt. Wenn diese Angabe fehlt darf das Segment jede Grösse annehmen.

Padding – Füllfeld – variable Bit-Breite

Dieses Feld stellt sicher, das der Header ein vielfaches von 32 Bit aufweisst. Das Feld wird mit Nullen aufgefüllt.

Data

Gehört nicht zum Header, sondern es handelt sich um die eigentliche “Nutzlast” des Segmentes.

Der Handshake (hier noch einmal) mit Sequenz- und Bestätigungsnummer

SQN = Sequenz Number
AKN = Achnowlegdement Number

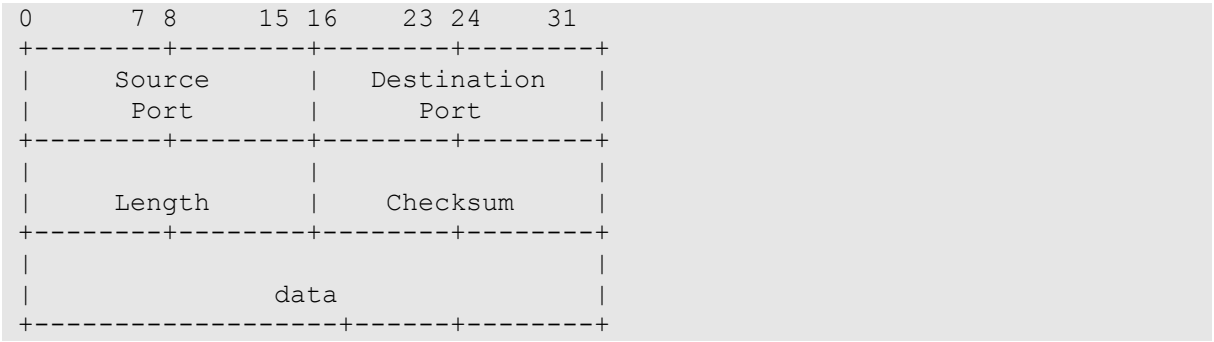
```
    Verbindsaufbau

    Client (X)                                    Server (Y)

1.     ----------------- SYN SQN=X ------------------>
2.     <---------- SYN/ACK AKN=X+1 SQN=Y -------------
3.     ------------  ACK AKN=Y+1 SQN=X+1 ------------>
.
    Bestehende Verbindung
.
N      <-------------ACK AKN=X+2 SQN=Y+1--------------
N+1    ------------  ACK AKN=Y+2 SQN=X+2 ------------>
N+2    <-------------ACK AKN=X+3 SQN=Y+2--------------
```

UDP (User Datagram Protocol)

Das Protokoll UDP (User Datagram Protocol) arbeitet verbindungslos (s.S. 24). DNS und NTP arbeiteten mit dem Protokoll UDP. Anwendungen die UDP verwenden müssen über eigene Korrrekturmassnahmen für verlorene oder in der falschen Reihenfolge eingetroffenen Pakete verfügen, da das Protokoll selbst nicht über diese Fähigkeiten verfügt.

Unten sieht man eine Darstellung des UDP-Headers. Dieser ist bedeutend einfacher aufgebaut als der TCP-Header. Für weitere Informationen zum Header verweise ich auf RFC 768.

Der UDP-Header



Anwendungsschicht (Application Layer)

Die Programme der Anwendungsschicht verwenden die darunterliegenden Protokolle um über das Netz zu kommunizieren. Es handelt sich meist um Client-/Server-Anwendungen. Diese Programme arbeiten mit Ports. Dies ist ein Zusatz zum Protokoll, der die Anwendung identifiziert. Eine HTTP-Verbindung kommuniziert standardmässig auf Port 80. Mail mit SMPT auf Port 25 und POP3 auf Port 110, etc.. Eine Auflistung der Anwendungen und der zugehörigen Ports findet man in /etc/services.

Bei solch einer Client-Server-Anwendung wird die Kommunikation im allgemeinen vom Client eröffnet. Der Client nimmt eine Verbindung zum Server auf, initiiert also diese Verbindung. Dazu verwendet er einen Port > 1024 und tritt mit dem betreffenden Port des Servers in Verbindung. Danach geht die Initiative an den Server über. Dieser verbindet sich jetzt über seinen Port mit dem Port des Clients.

Übertragen auf eine HTTP-Kommunikation sieht dies so aus:

```
HTTP-Client (Browser)                                          HTTP-Server
                              Initiierung
      Port > 1024       ---------------------------------->    Port 80

                              Antwort
      Port > 1024       <----------------------------------    Port 80
```

Für jede Kommunikation bilden sich dann Paare aus der Portnummer und IP-Adresse. Diese Paare werden Sockets genannt und bilden die Endpunkte der Kommunikation.

Hier ein Überblick über einige wichtige Protkolle der Anwendungsschicht

ftp - File Transfer Protocol - Ports 20/tcp, 21/tcp

ftp dient zum Übertragen von Dateien. Bei ftp gibt es 2 Besonderheiten. Zum einen arbeitet ftp mit 2 Ports. Zudem gibt es den aktiven und den passiven Modus (Passiv Mode und Active Mode). Beim aktivem Modus ist Port 21 der Steuerport (Command Port/Control Port) und dient zur Kommunikation zwischen Client und Server. Port 20 ist der Datenport (Data Port). Im passiven Modus wird als Datenport ein Port > 1023 ausgehandelt.

ssh - Secure Shell - Port 22/tcp

Mit Secure Shell kann man sich auf einem entfernten Computer (Remote Host) einloggen. Für die Authentifizierung werden verschlüsselte Passwörter verwendet. Die Datenübertragung erfolgt ebenfalls verschlüsselt.

Secure Copy, scp ist Bestandteil von ssh und dient zum Übertragen von Dateien zwischen den verbundenen Rechnern.

telnet - Port 23/tcp

Mit Hilfe von Telnet kann man sich auf einem entfernten Computer einloggen. Telnet arbeitet mit Klartext-Passwörtern und unverschlüsselt. Aus diesem Grunde wird es heute kaum noch verwendet, sondern modernere Protokolle wie SSH.

smtp - Simple Mail Transfer Protocol - Port 25/tcp

Dieses Protokoll dient zum Versenden von E-Mail. Es gibt eine Fülle von Clients die diese Aufgabe übernehmen. Diese senden die E-Mails an SMTP-Server, auch Mail Transfer Agent (MTA) genannt. Bekannte MTA sind Sendmail, Postfix und qmail.

dns - Domain Name System - Port 53/udp/tcp

dns dient zur Namensauflösung oder mit anderen Worten zum Übersetzen von Domain-Namen in IP-Adressen (forward lookup) bzw. umgekehrt von IP-Adressen in Domain-Namen (reverse lookup). Abhängig von der Paketgrösse verwendet dns die Protokolle udp bzw. tcp.

http - Hypertext Transfer Protokoll - Port 80/tcp

http dient zum Übertragen von, mit der Seitenbeschreibungssprache (x)html formatierten, Webseiten im Internet. Durch Erweiterungen kann es heute jedoch auch andere Aufgaben wahrnehmen wie z.B. Dateiübertragung (File Transfer).

pop3 - Post Office Protocol - Port 110/tcp

Mit Hilfe von pop3 holen Mail-Clients die E-Mails beim Mail-Server ab. Aufgrund seiner Beschränkungen, es erlaubt nur die E-Mails abzuholen oder auf dem Server zu löschen, wird es zunehmend durch imap abgelöst.

nntp - Network News Transfer Protocol - Port 119/tcp

Die im Internet weit verbreiteten Newsgroups verwenden dieses Protokoll.

ntp - Network Time Protocol - Port 123/udp

Mit Hilfe von ntp können Hosts im Internet die genaue Zeit von hochpräzisen Zeitgebern, wie z.B. den Servern der Physikalisch-Technischen Bundesanstalt (PTB) empfangen.

imap - Internet Message Access Protocol - Port 143/tcp

imap dient zum Verwalten von E-Mails im Netz. Bei imap können, im Gegensatz zu pop3, die Mails auf dem Server verbleiben. Dies ermöglicht dann den Zugriff mit unterschiedlichen Rechnern auf ein und dasselbe Postfach.

https - Hypertext Transfer Protocol Secure - Port 443/tcp

https verwendet SSL/TLS zum Verschlüsseln der Daten und bietet damit eine höhere Sicherheit gegenüber http.

Samba - smb - Server Message Block Protocol

Ports:
137/udp - NETBIOS Name Service
138/udp - NETBIOS Datagram Service
139/tcp - NETBIOS Session Service
445/tcp - Microsoft-DS

Samba ist die Linux Inkarnation des Server Message Block Protokolls. Dieses wurde urspünglich von Microsoft als Datei und Druckdienst für Windows entwickelt. Aufgrund des hohen Erfolges von Windows wurde das Protokoll dann auch für Linux entwickelt.

Dadurch können auch Linux-Rechner als Datei- und Druckserver für Windows-Clients im Netzwerk eingesetzt werden. Ursprünglich wurden nur die Ports 137 - 139 verwendet, später kam dann noch der Port 445 hinzu.

Gegenüberstellung TCP/IP und OSI-Referenzmodell

Hier die Gegenüberstellung der 4 Schichten von TCP/IP mit den 7 Schichten von OSI.

	OSI-Referenzmodell	**TCP-IP-Referenzmodell**	
7	Application Layer	Application Layer	4
6	Presentation Layer		
5	Session Layer		
4	Transport Layer	Transport Layer	3
3	Network Layer	Internet Layer	2
2	Data Link Layer	Network Layer	1
1	Physical Layer		

In der Literatur und auch im Internet begegnet man hin und wieder auch dieser Darstellung:

	OSI-Referenzmodell	**TCP-IP-Referenzmodell**	
7	Application Layer	Application Layer	5
6	Presentation Layer		
5	Session Layer		
4	Transport Layer	Transport Layer	4
3	Network Layer	Internet Layer	3
2	Data Link Layer	Data Link Layer	2
1	Physical Layer	Physical Layer	1

Wie bereits oben erwähnt wird es vielfach als Schwäche des TCP/IP-Referenzmodells angesehen, dass 2 unterschiedliche, sehr spezifische Aufgabenbereiche im Network Layer zusammengefasst sind. Daher wird das Modell hin und wieder mit dem OSI-Referenzmodell gemixt. Diese Art der Darstellung wird auch Hybrid-Modell genannt.

Trouble Shooting in IP-Netzwerken (in Stichworten)

Schritt 0: IP-Adresse, Netzmaske und Loopback-Device prüfen mit ifconfig.

Schritt 1: ping 127.0.0.1, wenn ok dann Schichten 7-3 geprüft, falls ping 127.0.0.1 mit Fehler, dann Schichten 3+4 fehlerhaft. Netzwerktreiber neu installieren.

Schritt 2: ping eigene IP-Adresse, wenn ok alle Schichten bis Schicht 1 geprüft, aber das Kabel wird nicht geprüft. Heisst aber, dass alle Treiber und die Karte funktionieren.

Schritt 3: ping fremde IP-Adresse aus eigenem, lokalem Netz.
Wenn Fehler Verkabelung, Hub, Switch, Gegenstelle n. ok.

Schritt 4: ping auf IP-Adresse in fremdem Netzwerk. Wenn ok dann funktioniert auch das Routing.

Schritt 5: ping mit Rechnername im eigenen/fremden Netzwerk, wenn ok dann funktioniert die Namensauflösung.

Vor dem Installieren von Servern wie Webservers, Datei-und Druckserver, etc., müssen die Schritte 1 - 5 im ganzen Netz funktionieren!
D.h.: Netz Hardware, DHCP, Routing und DNS sind damit überprüft.

Ethernet

Das Ethernet-Protokoll gehört zu den Schichten 1 und 2 des OSI-Referenzmodells. Es ist das am meisten verwendete Protokoll für die Datenübertragung über das Transportmedium. Ob andere Protokolle für diesen Zweck evt. besser geeignet sind spielte dabei keine Rolle. Der “Markt” hat diese Entscheidung schlicht und einfach über den Preis getroffen.

Topologien

Ethernet wurde ursprünglich für eine Bus-Topologie mit Koxialkabeln entwickelt. Diese Tech- nik, 10Base5 bzw. 10Base2 erreichte ein maximale Übertragungsrate von10 MBit/s. Die heute gebräuchlichen Techniken 10BaseT, 100BaseT, und 1000BaseT arbeiten mit einer Sterntopologie. Allerdings kann Ethernet seine Herkunft nicht verleugnen. Auch wenn die physikalische Topologie ein Stern ist, die logische Topologie bleibt ein Bus. Dieser wird dann von den Sternpunkten, den Hubs oder den Switches, emuliert.

Ethernet in seiner heutigen Form ist in dem Standard IEEE 802.3 festgelegt.

Die Verkabelung erfolgt mit Twisted Pair Kabeln. Im Moment gibt es diese Ethernet-Varianten:

* 10BaseT mit einer Datenübertragungsrate von 10MBit/s, Kabel min. CAT3
* 100BaseT mit einer Datenübertragungsrate von 100MBit/s, Kabel min. CAT5
* 1000BaseT mit einer Datenübertragungsrate von 1GBit/s, Kabel min. CAT5e

Diese Datenübertragungsraten werden in der Praxis jedoch nicht erreicht. Durch den Protokoll-Overhead und die Kollisionen im Netz (s.u.) wird die Datenübertragungsrate eingeschränkt.

Ethernet und die OSI-Schichten

Auf der **OSI-Schicht 1, der Physikalischen-Schicht**, definiert Ethernet die Datenübertragungsrate und die Form der Kodierung.

Auf der **Schicht 2, der Sicherungs-Schicht** ist Ethernet zuständig für die Adressierung (MAC-Access) und die Flusskontrolle (Flow Control). Für diese 2 Aufgaben wird die Sicherungsschicht von Ethernetprotokoll nochmals in 2 Schichten unterteilt.

Die untere Schicht heisst "Medium Access Control Schicht (MAC)" . Sie regelt den Zugriff der oberen Schichten auf das Übertragungsmedium, sprich die Netzwerkkarte. Zur Identifizierung des Mediums dient die MAC-Adresse, auch Hardwareadresse oder Ethernetadresse genannt. Wobei eine Station auch über mehrere Netzwerkkarten verfügen kann und dann auch mehr als eine Hardwareadresse aufweisst.

* Die MAC-Adresse ist (im Idealfall) einmalig auf der Welt. Leider gibt es auch hin und wieder Fälle wo dies nicht zutrifft.
* Sie besteht aus 6 Byte, die ersten 3 Byte bilden die Hersteller-ID, die letzten 3 Byte die Karten-ID. Die MAC-Adresse ist auf der Karte fest verdrahtet.
* Die MAC-Adresse steht im Datenpaket, damit auch der richtige Empfänger die Daten empfängt. Die 2. Schicht des Empfängers nimmt die Daten nur an, wenn sie an seine MAC-Adresse gerichtet sind.

Die obere Schicht heisst "Logical Link Control (LLC)" . Diese hat dann auch die Aufgabe Datenübertragungsfehler aus der Physikalischen-Schicht zu entdecken und zu korrigieren. Diese Schicht soll eine fehlerfreie Kommunikation gewährleisten. Hier werden die Daten in sog. Frames (Rahmen) verpackt. Diese enthalten neben den Daten eine Prüfsummen sowie

die Ziel- und Quelladresse der beteiligten Netzwerkkarten. Mit Hilfe der Prüfsummen können fehlerhafte Informationen entdeckt und dann neu angefordert werden.

Die Rahmen werden im “Broadcast“-Verfahren versendet, alle Stationen hören die Verbindung ab und warten auf Rahmen die ihre eigene MAC-Adresse enthalten. Wenn solch ein Rahmen empfangen wird, wird dieser verarbeitet und an die höheren Protokollschichten weitergeleitet.

CSMA/CD

CSMA/CD heisst Carrier Sense Multiple Access mit Collision Detection. Dies ist der Mechanismus den Ethernet verwendet um den Zugriff auf das Transportmedium zu kontrollieren.

* In einem Ethernet-Netzwerk kann immer nur eine Station senden. Wenn 2 oder mehr Stationen zur gleichen Zeit senden kommt es zu einer Kollision der Pakete. Diese Pakete sind dann verloren. Kollisionen gibt es in einem Ethernet allerdings des öfteren. Je mehr Stationen sich im Netz befinden, desto öfter kommt es auch zu Kollisionen. Dies wirkt sich dann natürlich auch auf die Datenübertragungsrate aus.

* So funktioniert CSMA/CD

    * Bevor eine Ethernet-Station einen Rahmen (Frame) an das Kabel sendet, hört es an der Verbindung, um herauszufinden ob bereits ein Rahmen auf dem Medium transportiert wird (**Carrier Sense**).
    * Wenn das Medium frei ist, wird die Station ihre Rahmen senden.
    * Es kann jetzt allerdings vorkommen, dass 2 Stationen zur gleichen Zeit feststellen, dass die Leitung frei ist und beide fangen gleichzeitig an Rahmen zu versenden (**Multiple Access**). Dies führt dann allerdings zu einer Kollision.
    * Mit Hilfe der **Collision Detection** können die Stationen feststellen ob eine Kollision stattgefunden hat. In diesem Falle brechen die beiden beteiligten Stationen die Sendung ab.
    * Der ganze Vorgang startet dann wieder von vorne.

Full Duplex

Half Duplex bezeichnet ein Verfahren bei dem immer nur eine Station Pakete senden kann und die andere Station empfängt Pakete. Im Full Duplex Verfahren können beide Stationen gleichzeitig senden und empfangen. Das oben beschriebene CSMA/CD ist ein Half Duplex Verfahren. Ergänzungen am Ethernet-Protokoll machen heute aber auch Full Duplex möglich. Die Pakete der beiden Stationen werden dann nach einem festgelegten Schema zeitversetzt gesendet.

Offene Themen

* Netzwerkbegriffe
  Was ist ein Broadcast, Multicast?

* Netzwerkgeräte
  was ist ein Hub, Router, Bridge, Switch etc...

* Kabel
  Strukturierte Verkabelung

* Subnetting

* Routing unter Linux

* Netzwerkkarte einbinden
  virtuelle Netzwerkkarten

Index