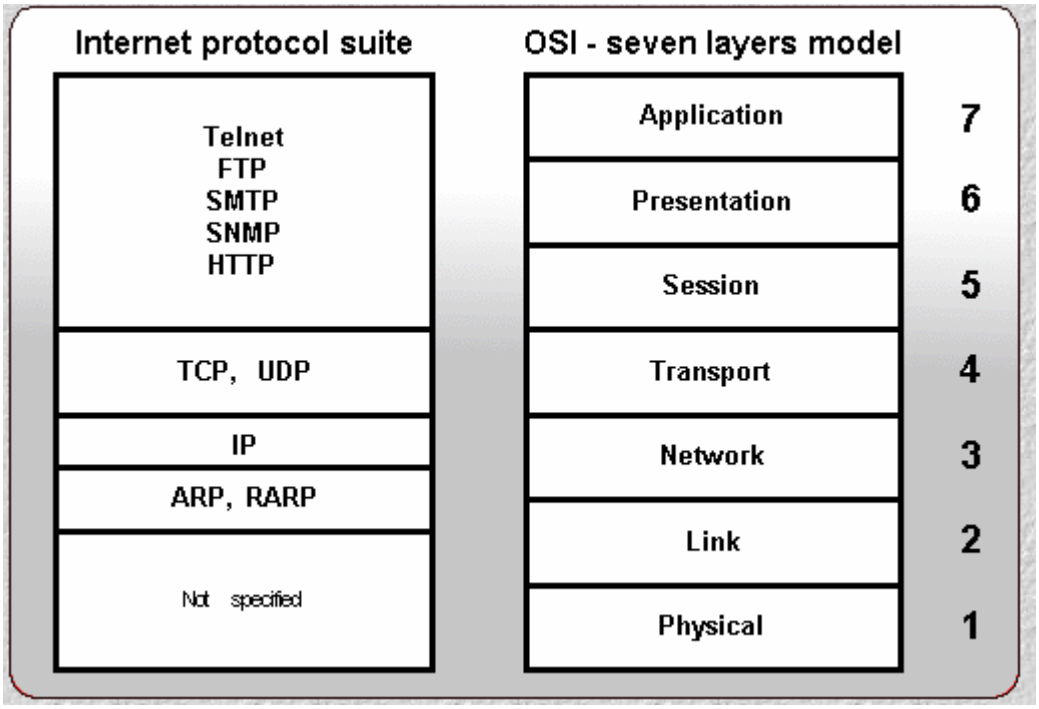
C6 TCP/IP Protokoll

Im Internet werden auf der Netzwerk- und Transportebene die TCP/IP Protokolle benutzt.



Bei TCP/IP ist zunächst auf der Vermittlungsebene (Ebene 3) das IP (Internet protocol) Protokoll angesiedelt. Diese Protokoll ist verbindungslos. Darauf baut dann das TCP (Transmission control protocol) Protokoll auf der Transportschicht (Ebene 4) auf. Dieses Protokoll ist verbindungsorientiert. Für Anwendungen, die die Funktionen einer Verbindung nicht benötigen bzw. für solche für die der Overhead zu groß ist, wird das UDP(User Datagram Protocol) Protokoll, welches mit verbindungslosen Datagrammen arbeitet, benutzt.

Das IP-Protokoll

Die Aufgabe des IP-Protokolls ist es, Datenpakete von einem Sender zu einem Empfänger über mehrere Netze hinweg zu transportieren. Dabei setzt es auf den Link-Layer (z.B. Ethernet) auf. Die Übertragung ist paketorientiert, verbindungslos und nicht garantiert.

Verbindungslos (connectionless)

Dies bedeutet, dass keine direkte Verbindung zwischen dem Sender und dem Empfänger aufgebaut wird, wie z.B. beim Telefon. Die Daten suchen selbständig einen nicht vorher bestimmten Weg durch die Netze. Dabei kann es vorkommen, dass die einzelnen Pakete in falscher Reihenfolge ankommen können. Dieses muss dann am Zielrechner wieder geordnet werden (reassembliert).

Nicht garantiert (best effort)

Das IP- Protokoll verschickt die Datenpakete, kann aber eine Zustellung zum gewünschten Empfänger nicht garantieren, ist also ein best effort service. Da alle Pakete unabhängig von

den anderen losgeschickt werden, kann es vorkommen, dass ein Paket verloren geht oder auch verdoppelt wird. Des weitern prüft das Protokoll eine Summe, die im Header mit gesandt wird, auf Korrektheit. Falls diese nicht stimmt, wird das Paket einfach verworfen.

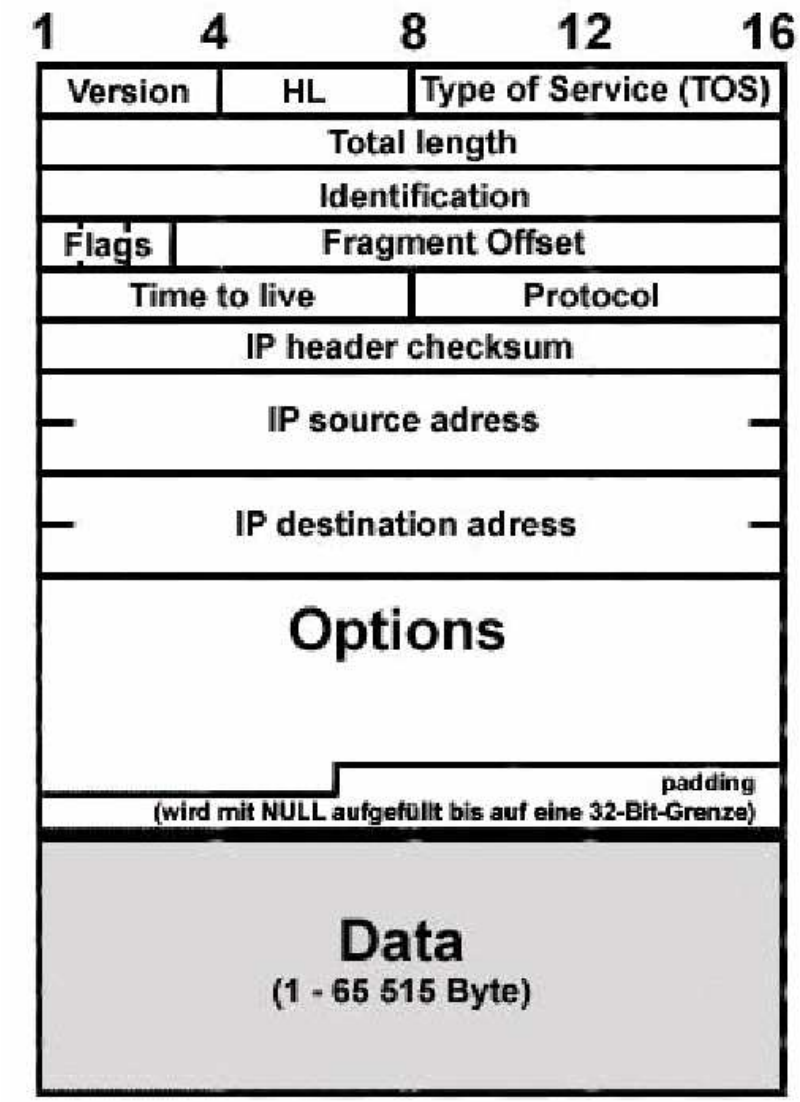
Paketorientiert

Die Daten, die mittels IP verschickt werden, fragmentiert (zerstückelt) das Protokoll in einzelne Datagramms, die unabhängig von einander durch die Netze transportiert werden. Die Größe ist abhängig von der zu transportierenden Datenmenge und der Maximum Transfer Unit (MTU) des jeweiligen Netzes. Dabei können Router die Pakete für Unternetze nochmals fragmentieren. Die maximale Länge von IP-Datenpaketen ist auf 65.535 Bytes beschränkt. Dabei fallen mindestens 20 Bytes auf den Header (maximal 60 Byte).

Header

32- Bit –Blöcke des Headers werden als Wörter bezeichnet. Die ersten 5 Wörter (20Bit) fest sind. Danach können noch Optionen folgen (max. 40 Byte).

Aufbau eines IP- Paketes



Version:

Länge: 4 Bit
Enthält die IP Versionsnummer. Momentan ist die Version IPv4 aktuell.

Header Length (HL)
Länge: 4 Bit
Es gibt die Anzahl der Wörter(32-Bit) einschließlich der "Options" des Headers an. Der Mindestwert ist 5 (0101), falls keine Optionen vorhanden sind und der Maximalwert 15 Wörter (1111) daraus ergibt sich eine maximale Header-Länge von 60 Byte.
Somit sind für die Optionen bis zu 40 Byte vorgesehen. Daraus resultiert, dass einige Optionen heutzutage sinnlos geworden sind (z.B. die Aufzeichnung der Route durch alle IP-Adressen der Router)

Type of Service (TOS):
Länge: 8 Bit
Dieses Feld ist wiederum unterteilt in die ersten 3 Bits (precedence field), in dem die Priorität angegeben werden kann, was heute ignoriert wird.
In den nächsten 4 Bits, die 4 TOS Bits, wird die bevorzugte Funktion auf 1 gesetzt (nur ein Bit darf gesetzt sein).
1.TOS-Bit: Verzögerungsminimierung
2.TOS-Bit: Durchsatzmaximierung
3.TOS-Bit: Zuverlässigkeitsmaximierung
4.TOS-Bit: Kostenminimierung
Das letzte Bit muss 0 sein.
Dieses Feld wird von den Routern heutzutage völlig ignoriert.

Total Length:
Länge: 16 Bit
Die Gesamtlänge des Datagramms in Bytes ist hier angegeben, also Header + Daten. Das Feld ist 16 Bit lang, damit ist die Maximallänge des Pakets auf 65 535 Byte festgelegt. Dieses Feld wird verändert falls das Datagramm aufgrund einer zu kleinen MTU fragmentiert werden muss. Jeder Router oder Host muss in der Lage sein Pakete mit einer Länge von 576 Byte (512 Datenbyte + Kopf) zu empfangen und zu reassemblieren.

Identification:
Länge: 16 Bit
In diesem Feld ist eine Zahl eingetragen, die bei jedem gesendeten Datagramm um eine Eins inkrementiert wird. Das Feld dient der eindeutigen Kennzeichnung von Datenpaketen und der Zuordnung von Fragmenten zu Datagrammen (alle Fragmente eines Datagramms enthalten den gleichen Wert).
Anhand des Identifikator-Feldes ist der Zielknoten in der Lage, Fragmente den entsprechenden Datagrammen zuzuordnen. Die Vergabe der Identifikationsnummern erfolgt durch ein höheres Protokoll und wird als Parameter an das IP- Protokoll übergeben.

Flags:
Länge: 3 Bit
Das erste Bit ist immer 0.
Das zweite Bit gibt an, ob das Datagramm fragmentiert werden darf (0) oder nicht (1). Falls dieses auf 1 (DF=Don`t fragment) steht und das Paket zu groß für das Netz ist, wird es nicht übertragen, sondern weggeschmissen und eine ICMP-Nachricht verschickt.
Eine 1 beim letzten Bit gibt an, dass noch mehr Pakete zu dem Datagramm gehören. Eine 0 bedeutet, dass es das letzte Fragment ist (MF=More fragmens).

Fragment Offset:
Länge: 13 Bit

Wenn das Paket fragmentiert ist gibt der Offset die genaue Lage des Fragmentes innerhalb des Datagramms in Bezug auf den Anfang der Daten wieder. Falls keine Fragmentierung vorliegt oder beim ersten Fragment ist dieses auf Null gesetzt. Da das Feld nur 13 Bit hat, wird der Abstand in Einheiten zu 8 Byte angegeben.

Time-to-Live (TTL):
Länge: 8 Bit
Dies ist die Lebensdauer des Datagrams, die vom Sender gesetzt wird (meist 32 oder 64). Jeder Router verringert diesen Wert um eins. Falls der Wert Null ist, ist die Lebensdauer des Datagrams abgelaufen und es wird weggeschmissen.
Es wird versucht, dem Sender eine ICMP- Nachricht zukommen zu lassen.
Mit diesem Feld wird verhindert, dass Datagramme endlos durchs Netz transportiert werden.

Protocol:
Länge: 8 Bit
Enthält die Identifikation des Transportprotokolls, dem dieses Paket zugestellt werden muss. Für die Standardprotokolle sind vom NIC (Network Information Center) Werte festgelegt
z.B:
17 UDP
6 TCP
1 ICMP
.

Header Checksum:
Länge: 16 Bit
Das Feld beinhaltet die Prüfsumme aller Header-Inhalte, dazu werden diese in 16 Bit-Wörter aufgeteilt. Beim Ankommen eines Datagramms bei einem Router berechnet dieser die Header-Checksum. Falls das Ergebnis nicht 0 ist (checksum error), wird das Packet vernichtet. Beim weitersenden muss erst noch die neue Checksum (mindestens das TTL-Feld wurde geändert) berechnet und eingetragen werden. Falls nur das TTL-Feld geändert wurde, kann diese leicht errechnet werden, indem man die alte Prüfsumme inkrementiert.
Dadurch hat das IP- Protokoll ein hohes Maß an Korrektheit. Die Nutzdaten werden auf der IP-Schicht nicht geprüft, die Prüfung findet in den Protokollen der Transportschicht statt, da nur diese, falls sie verbindungsorientiert sind, eine Wiederholung bewirken können. Die Prüfsumme ist sehr einfach, nämlich das 1er Komplement der Summen der 16-Bit Worte.

Source IP-Address:
Länge: 32 Bit
IP-Adresse des sendenden Rechners.[3]

Destination IP-Address:
Länge: 32 Bit
IP-Adresse des Empfangsrechners.[3]

Options:
Länge: variable bis zu 40 Byte
Kein Pflichtbestandteil des IP-Headers.
In diesem Feld können Optionen angeschaltet werden, welche im RFC791 definiert wurden. Die Optionen müssen von allen Routern auf dem Weg zum Empfänger unterstützt werden bzw. freigeschaltet sein. Diese werden allerdings nur noch selten benutzt.

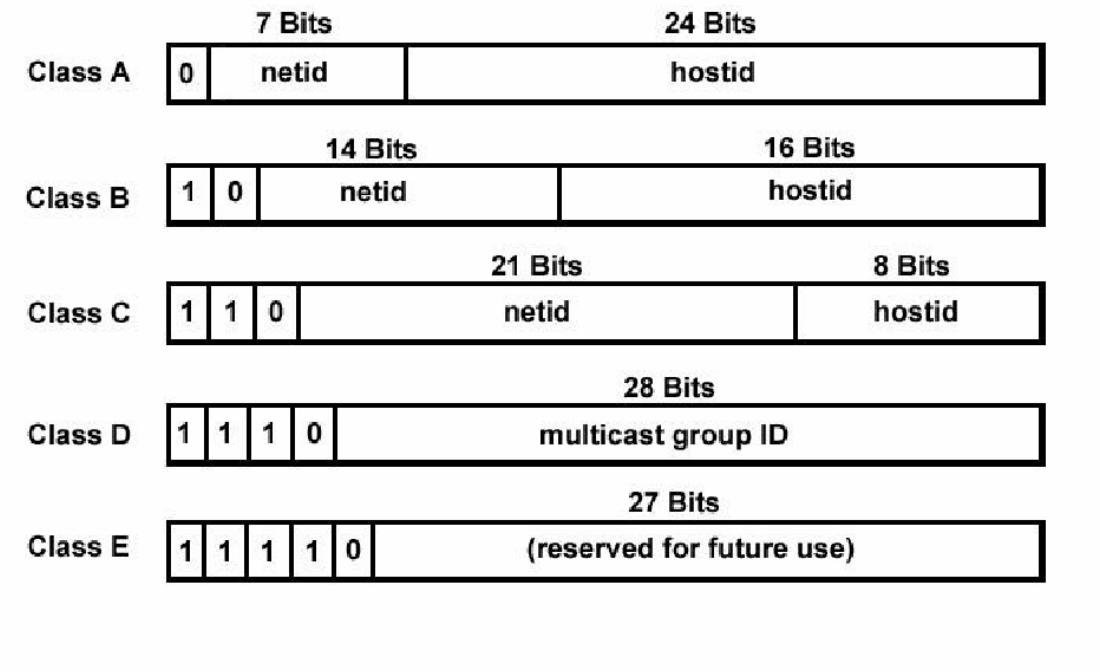
IP-Adressen

Man stelle sich das Internet wie ein grosses Netzwerk vor, welches wie jedes beliebige andere physikalische Netz aufgebaut ist. Für diese Computernetze haben sich die Entwickler der TCP/IP-Protokollfamilie ein Schema überlegt, in welchem angelehnt an physikalische Netze jedem Host im Internet eine Adresse zugeteilt wird, die so genannte „Internet Adresse" oder „IP-Adresse". Diese Adressen bestehen aus ganzen Zahlen und werden systematisch vergeben:
1. um den Verwaltungsaufwand zu minimieren
2. um das Routing effizient zu machen
Es wird jedem einzelnen Host eine eindeutige 32-bit Internetadresse zugewiesen, welche ihn sowie das Netzwerk identifizieren, dem er zugehörig ist.
Im einfachsten Fall besteht jede Adresse aus einem Paar. Dabei steht die "netid" für die Identifikation des Netzwerks und die "hostid" für die Identifikation des Hosts. Die hostid 0 ist nach Vereinbarung nicht an einen individuellen Host zu vergeben, sondern steht immer für das Netzwerk selbst.

Notation Von IP-Adressen

IP- Adressen werden bei der Kommunikation mit und durch Menschen meist als vier dezimale, ganze Zahlen ausgegeben, welche durch einen Punkt getrennt werden. Jede Zahl gibt den Wert von einem Oktett der Adressbinärdarstellung wieder. Somit schreibt man eine beliebige Internetadresse wie folgt:
Binärnotation: 10000000 00001010 00000010 00011110
Dezimalnotation: 128.10.2.30

IP-Klassen



Klasse A:
Das erste Byte hat einen Wert kleiner als 128, d.h. das erste Bit der Adresse ist 0. Das ersten 7 Bits sind die Netzwerknummer, die letzten 3 Bytes identifizieren einen Host im Netz. Es gibt demzufolge also 126 Klasse A Netze, die bis zu 16 Millionen Host in einem Netz verwalten können. (127 ist für die loopback-Adresse reserviert)

Klasse B:
Ein Wert von 128 bis 191 für das erste Byte (das erste Bit ist gleich 1, Bit 2 gleich 0) identifiziert eine Klasse B Adresse. Die ersten 14 Bits identifizieren das Netzwerk, die letzen beiden Bytes einen Host. Das ergibt 16.382 Klasse B Netze mit bis zu 64.000 Hosts in einem Netz.

Klasse C:
Klasse C Netze werden über Werte von 192 bis 223 für die ersten 21 Bits (die ersten beiden Bits sind gleich 1, Bit 3 gleich 0) identifiziert. Es gibt 2 Millionen Klasse C Netze, d.h. die ersten drei Bytes werden für die Netzwerkadresse verwendet. Ein Klasse C Netz kann bis zu 254 Host beinhalten.

Klasse D:
Klasse D Adressen, sogenannte Multicast-Adressen, werden dazu verwendet ein Datengramm an mehrere Hostadressen gleichzeitig zu versenden. Das erste Byte einer Multicast-Adresse hat den Wertebereich von 224 bis 239, d.h. die ersten drei Bit sind gesetzt und Bit 4 ist gleich 0.
Der weitere Bereich der IP-Adressen von 240 bis 254 im ersten Byte ist für zukünftige Nutzungen reserviert. In der Literatur wird dieser Bereich oft auch als Klasse E bezeichnet.

Class Range
A 1.0.0.0 to 126.0.0.0
B 128.1.0.0 to 191.255.0.0
C 192.0.1.0 to 223.255.255.0
D 224.0.0.0 to 239.255.255.255
E 240.0.0.0 to 255.255.255.254

Einige Klasse-B Netze:

```
uni-koblenz.de [141.26.0.0]
Rlp-Net.net     [143.93.0.0]
Uni-Mainz.DE    [134.93.0.0]
uni-sb.de        [134.96.0.0]
dfn.de           [188.1.0.0]
uni-kl.de      [131.246.0.0]
```

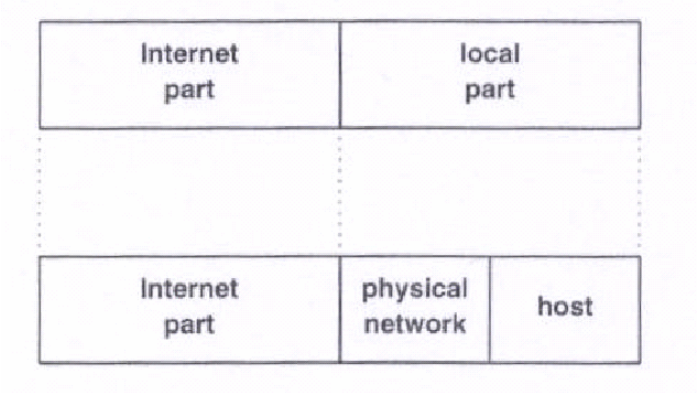
Das Problem der Adressknappheit

Das originale, in Klassen eingeteilte IP-Adresschema hat sich bisher gut bewährt, es hat aber einen großen Nachteil. Die Entwickler haben das explosionsartige Wachstum des Internets nicht mit einkalkuliert. Somit werden irgendwann alle möglichen Adressen vergeben sein (sind teilweise auch schon). Insbesondere betrifft dies die mittelgroßen Netzwerke, die der B-Klasse zugeordnet werden. Es stellt sich also die Frage: Wie kann man die Anzahl der zu vergebenden Netzwerkadressen minimieren, ohne das 32-bit Adresschema zu reformieren, ja sogar aufzugeben?
Um z.B. den Gebrauch von B-Klasse Netzwerken zu verringern, müsste man auf auf C-Klasse Adressen zurückgreifen. Es ist also die Idee, dass viele physikalische Netzwerke sich eine Netzwerkadresse (die netid also) teilen. Eine Möglichkeit besteht in den "standard-IP-subnets".

Subnet Adressing oder subnetting

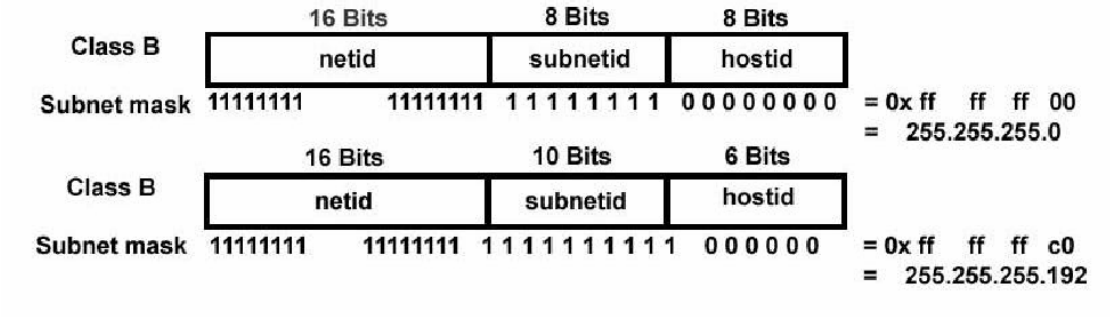
Subnetaddressing ist eine Technik, die dazu benutzt wird mit einer einzigen Netzwerkadresse mehrere physische Netze anzusprechen. Der Host muß dabei das Subaddressing unterstützen. Subnetaddressing benutzt einige Bits der hostid als physische Netzwerkadresse. Dabei werden Netzwerkadressen wie gewohnt angewendet mit dem Unterschied, dass das Host-Suffix (die hostid) in 2 Teile aufgeteilt wird. Der erste Teil wird nun zu der „subnetid", der zweite bleibt

der hostid-Teil, der nun etwas kleiner ist. Dabei werden z.B. die 16 Bits einer Klasse-B Adresse bevorzugt gerecht aufgeteilt, also 8 Bits für die subnetid (bis zu 254 physikalische Subnetze), 8 Bits für die hostid (bis zu 254 Hosts). Die Einteilung ist ansonsten aber beliebig.



Die Subnetzmaske (subnetmask)

Subnetting ist mittlerweile ein obligatorischer Teil bei der Verwendung von TCP/IP in Netzwerken. Ist man einem Netz permanent untergeordnet, muss auch eine Subnetmask angegeben werden. Man ordnet diesen Host damit einem Subnet unter. Die Subnetmask ergibt sich, indem man alle Bits der net- und subnetid auf „1" und alle der hostid zugeordneten Bits auf „0" setzt. Aus der hexadezimalen Darstellung (4 Bits werden jeweils zusammengefasst) ergibt sich dann die Subnetmask.



Broadcasting

Das IP-Adressenschema bringt uns noch einen weiteren Vorteil mit, nämlich die "broadcast-address". Diese wird allen Hosts des Netzwerks zugewiesen. Broadcasting (broadcast zu deutsch: Rundfunk) ist somit die gleichzeitige Adressierung aller Stationen eines Subnetzes. Standardmässig ist jede hostid, die ausschliesslich aus Einsen besteht, reserviert für broadcasting. Broadcasting wird gebraucht, um in grossen Netzen seine eigene IP-Adresse und auch die der anderen Rechner über die Hardware- Adresse (MAC-Adresse) zu ermitteln. Dies erst ermöglicht überhaupt die Kommunikation zweier Rechner über Ethernet. In vielen Netzwerktechnologien, die broadcasting unterstützen (wie z.B. auch Ethernet), kann die Übertragung von Paketen so effizient sein wie eine übliche Pakettransmission, das ist aber trotzdem nicht garantiert.

Routing

Die auf der Transportschicht realisierten Verbindungen müssen nicht notwendig zwischen direkt verbundenen Knoten stattfinden. Daher muss eine Route ermittelt werden über die die

Pakete gesendet werden. Die hierfür notwendigen Routingstrategien kann man einteilen in statische bzw. nichtadaptive und dynamische bzw. adaptive Strategien.
Bei den nichtadaptiven (statischen) Verfahren werden voraus berechneten Tabellen benutzt. Bei den adaptiven (dynamischen) Verfahren werden Veränderungen der Topologie und die aktuelle Netzauslastung berücksichtigt. Hierfür ist ständiger Informationsaustausch zwischen den Knoten notwendig.

Routing -Verfahren Fluten

Hier sendet ein Knoten an alle Nachbarn, mit Ausnahme des Knoten, von wo das Paket gekommen ist. Damit findet das Paket auf jeden Fall, den Empfänger, wenn er im Netz ist. Damit die Pakete nicht im Netz kreisen, enthalten die Pakete einen Zähler, der bei jedem Knoten um eine verringert wird. Ist er 0, wird das Paket vernichtet. Dennoch müssen später eintreffende Duplikate erkannt und verworfen werden.

Routing -Verfahren Hot Potato

Ein Paket wird immer, um es wie eine heiße Kartoffel möglichst schnell loszuwerden, an den Nachbarn gesandt, der die kürzeste Warteschlange hat. Dadurch können Pakete evtl. große Umwege nehmen.

Routing mit einer Routing Tabelle

Normalerweise arbeitet ein Router jedoch mit einer Routing-Tabelle. Dort ist für eine Reihe von Netzen jeweils der Next-Hop, d.h. der Router über den das Paket gesandt wird eingetragen. Ferner ist ein Default-Router eingetragen, über den die Pakete geschickt werden, deren Netzadresse nicht in der Tabelle eingetragen ist. Diese Routimg-Tabelle kann zu einem von hand geändert werden (Static Routing) oder durch Algorithmen, die auf einem Nachrichtenaustausch zwischen den Routern basieren.
In vielen Fällen reicht dieses Static-Routing aus. Der Router für uni-koblenz.de [141.26.0.0] kennt die Router im Landesnetz Rheinland-Pfalz und die Router von unseren Subnetzen. Er weis auch welche Netzwerkadressen welchen Routern zugeordnet sind. Alle übrigen Adressen werden zum Router an der Uni-Mainz geroutet, über den unsere Internet-Konnektivität läuft.

Distanz- Vektor Routing

Das Netz bildet einen Graphen, die Kanten werden mit einem Gewicht belegt, die Distanz zum Ziel ist die Summe der Distanzen auf dem Weg. Eine Routingeintrag ist ein Triple (V,D,N), V ist der Zielknoten, D die Distanz und N (Next Hop) der Knoten, über den der Weg beginnt. Jeder Knoten sendet an seine direkten Nachbarn Routingnachrichten (V,D). Erhält man eine solche Nachricht vom Knoten N:
C =D +Gewicht der Kante zum Knoten N.
if es gibt keine Route zu V **then** neuer Eintrag (V,C,N)
else if es gibt Route mit Next Hop N **then** ersetze Nachricht durch (V,C,N)
else if es gibt Route mit größerer Distanz **then** ersetze Nachricht durch (V,C,N)

Link- State- Routing

Hier werden die Zustände der Verbindung zwischen benachbarten Knoten per Broadcast an alle Knoten gesandt. Jeder Knoten berechnet dann mit dem Dijkstra- Algorithmus (Informatik B) seine Routing- Tabelle.

Würde man diese Informationen zwischen allen Routern austauschen, wäre die datenmenge zu groß. Man teilt daher das Internet in Autonome-Systeme (AS). Innerhalb eines AS wird das IGP (Interior Gateway Protocol) benutzt. Die Tabellen hierfür werden durch Informationsaustausch innerhalb des AS erstellt. Für das Routing zwischen den AS wird heute das BGP4 (Border gateway Protocol Version 4) benutzt. Einen BGP Router betreiben z.B. Internet-Provider. Ein neues AS muss im gesamten Internet annonciert werden, damit es erreichbar ist. Neue Router in einem AS müssen nur innerhalb des AS annonciert werden.

ICMP

Das ICMP (Internet Control Message Protocol) wird zusätzlich zum IP Protokoll benötigt um Nachrichten für die Steuerung der Datenübertragung zwischen den Routern und Hosts auszutauschen. Dies kann u.a. sein:

- Antwort auf eine Echo-Anforderung
- Ziel nicht erreichbar
- Quelle muss mit dem Senden aufhören
- Route über einen anderen Router führen
- Echo Anforderung
- Router versendet Router- Information
- Host fordert Router- Information an
- Time to live erreicht den Wert 0
- Fehler im IP Packet
- Zeitstempel anfordern
- Zeitstempel liefern
- Subnetz- Maske anfordern
- Subnetz- Maske liefern

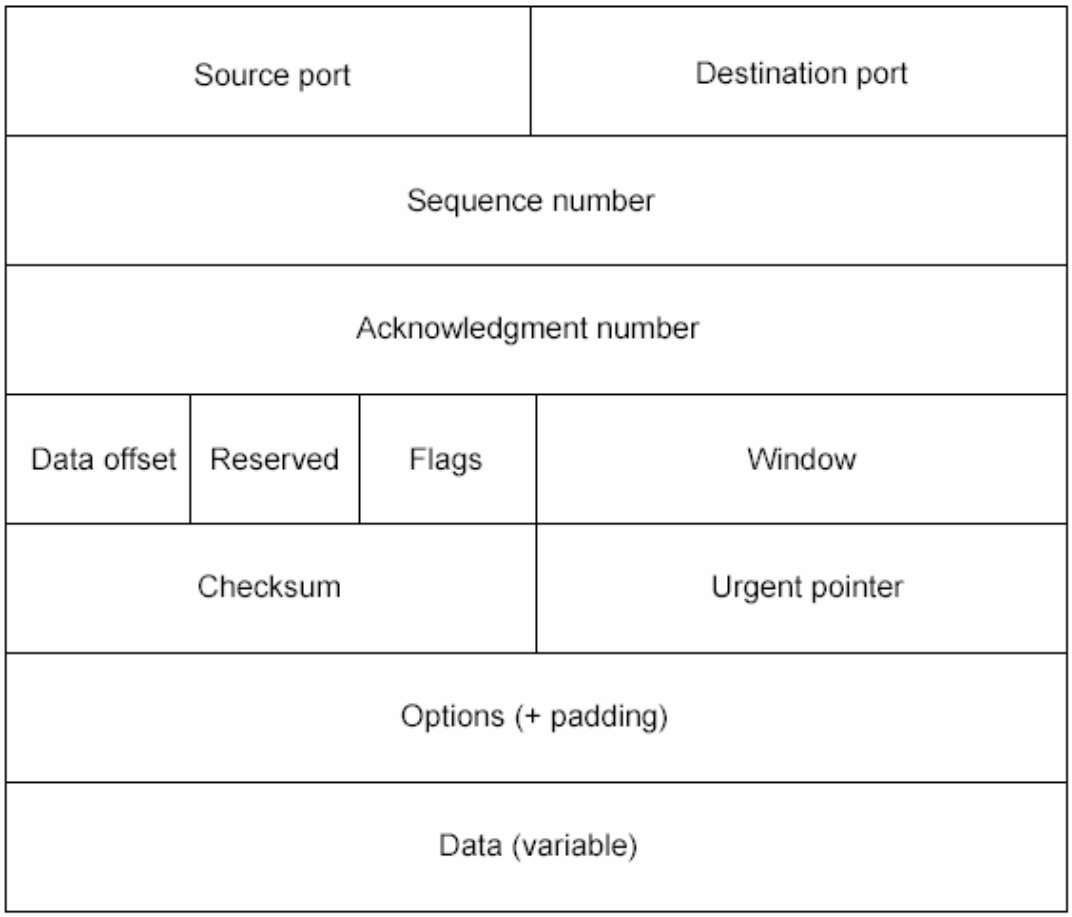
TCP

TCP ist ein verbindungsorientiertes Transportprotokoll für den Einsatz in paketvermittelten Netzen. Das Protokoll baut auf dem IP- Protokoll auf, unterstützt die Funktionen der Transportschicht und stellt vor der Datenübertragung eine gesicherte Verbindung zwischen den Instanzen her. Die Daten der höheren Schichten werden durch TCP nicht verändert, sondern lediglich segmentiert und in einzelnen Datenpaketen versendet, die einen Umfang von bis zu 65 kByte haben können. Das darunter liegende IP- Protokoll fragmentiert die TCP-Datensegmente in kleinere Datenpakete. Jedes Oktett eines Segments wird von TCP mit einer so genannten Sequenznummer versehen, was empfangsseitig die richtige Reihenfolge garantiert.
Die wesentlichen Dienstleistungen, die das TCP- Protokoll in Verbindung mit dem IP-Protokoll für die Anwendungsprozesse bereitstellt, sind die Ende- zu- Ende- Kontrolle, das Verbindungs-Management, die Flusskontrolle, die Zeitkontrolle und das Multiplexen von Verbindungen sowie die Fehlerbehandlung.
Die Ende- zu- Ende-Kontrolle arbeitet mit einer positiven Rückmeldung, bei der alle Datenpakete bestätigt und nicht empfangene erneut gesendet werden. Durch diesen Mechanismus ist eine einwandfreie Datenübermittlung gewährleistet. Das Verbindungs-Management sorgt für einen gesicherten Verbindungsaufbau mittels Handshake- Verfahren. Darüber hinaus sorgt das Verbindungs-Management für die einwandfreie Bereitstellung der Verbindung während der Übertragungsphase und für einen korrekten Verbindungsabbau.
Da alle übertragenen Datenpakete fortlaufend nummeriert und bestätigt werden, verhindert die Flusskontrolle den Verlust von Datenpaketen. Die Zeitüberwachung dient dazu, dass übertragene Datenpakete innerhalb eines bestimmten Zeitraums bestätigt werden. Findet innerhalb dieses Zeitraums keine Bestätigung statt, werden die Datenpakete erneut gesendet.
Um mehrere Prozesse gleichzeitig über TCP zu betreiben, werden für das Multiplexen

mehrere Ports zur Verfügung gestellt. Treten Fehler auf, tritt der Fehlermechanismus in Funktion und fordert die fehlerhaften Datensegmente von den höheren Schichten erneut an.

Eine TCP- Übertragung lässt sich in drei Phasen gliedern: die Initialisierungsphase, die Phase der Nutzdatenübertragung und die Phase des Verbindungsabbaus.
In der Initialisierungsphase erfolgt der aktive oder passive Verbindungssaufbau in der eine Eins- zu- Eins-Verbindung hergestellt wird, die während der gesamten Dauer des Datentransfers aufrechterhalten wird. Diese Phase, die durch einen Zwei- oder Drei- Weg-Handshake eingeleitet wird, dient auch der Synchronisation der Kommunikationspartner.
In der Datenübertragungsphase, die nach dem Verbindungsaufbau beginnt, erfolgt über die aufgebaute virtuelle Verbindung. Diese Phase ist geprägt durch die Übertragung der Datenblöcke und die Empfangsbestätigung der Sequenznummern durch den Kommunikationspartner. Die Phase des Datentransfers wird durch mehrere Timer überwacht, um beispielsweise unbestätigte Datenblöcke nachzusenden, um die Fenstergröße umzustellen, den Verbindungsabbau einzuleiten oder aber einen erneuten Verbindungsaufbau zu initiieren. Die Datentransferphase wird durch eine Flusssteuerung und durch verschiedene Algorithmen optimiert. Diese Algorithmen sind Steuerungsmechanismen für die Datenmenge, den Datenfluss und die Netzauslastung. Die dritte und letzte Phase, der Verbindungsabbau, kann einerseits nach der Übertragung aller Daten erfolgen, andererseits durch einseitigen Abbruch der Verbindung durch ein höheres Protokoll.

Aufbau eines TCP Paketes



- Source Port, Destination Port: identifiziert Anfangs- und Endpunkt der Verbindung. Diese entsprechen den Programmen, die als Sender bzw. Empfänger fungieren. Für die Standard Dienste sind die Ports festgelegt.

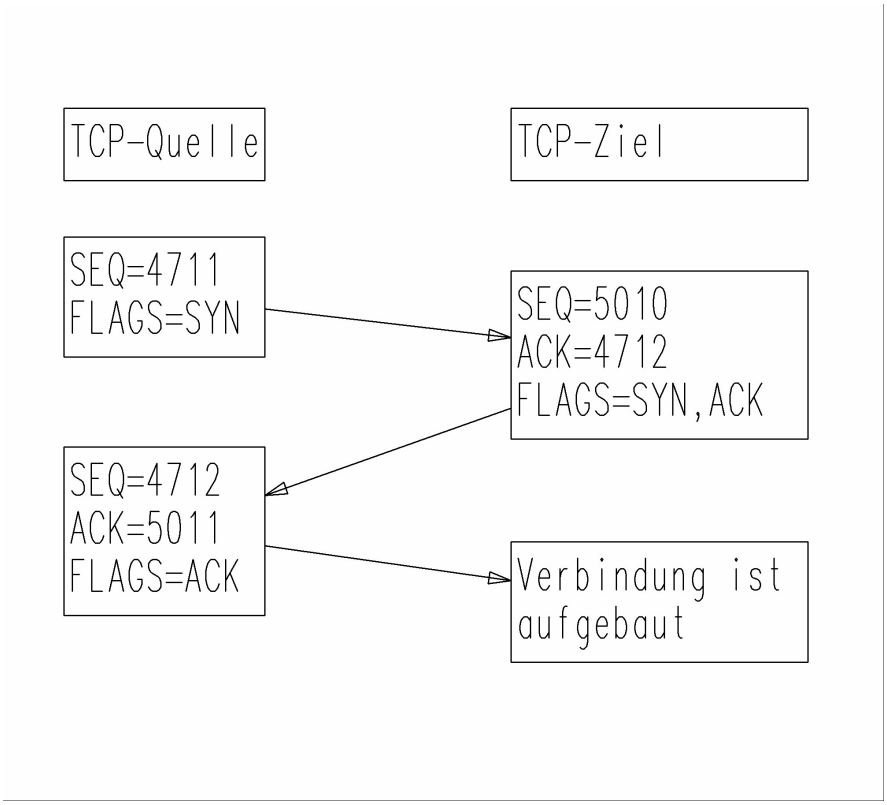
  20 FTP-Daten
  21 FTP-Steuerung
  23 Telnet
  25 SMTP
  53 DNS
  80 WWW
  Der Client benutzt als Zielport, die Portadresse des gewünschten Dienstes, als Quellport benutzt er eine freie, nicht reservierte Portnummer. Damit kann dann z.B. verschiedene gleichzeitig zum gleichen Server aufgebaute TCP- Verbindungen unterscheiden.

- Sequence Number: Sequenznummer des ersten Bytes in diesem Segment.
- Acknowledgement Number: Enthält die Nummer des nächsten Datenbyte, das die Gegenseite erwartet.
- Data offset: Länge des TCP Headers
- Flags:
  - URG: Urgent Pointer ist gültig
  - ACK: Acknowledgementfeld ist gültig
  - SYN: Synchronization der Sequenznummern beim Verbindungsaufbau
  - FIN: keine weiteren Daten vom Sender
  - RST: Zurücksetzen der verbindung im Fehlerfall
  - PSH: Push Data, Daten sorfort ausliefern, nicht zwischenspeichern.

- Window: Größe des zulässigen Sendefensters. Die Station ist bereit ohne weitere Quittung bis Acknowledgement Number+Window Bytes zu empfangen. Durch diese als Sliding-Window bezeichnete Technik, kann trotz der durch Quittungen gesicherten Übertragung, der Sender kontinuierlich Daten senden, falls der Empfänger mit seinen Quittungen das Fenster jeweils entsprechend verlängert.
- Checksum: Prüfsumme
- Urgent Pointer: Zeigt auf das letzte Byte, bis zu dem die Daten dem Anwendungsprogramm des Empfängers möglichst schnell zu übergeben sind.

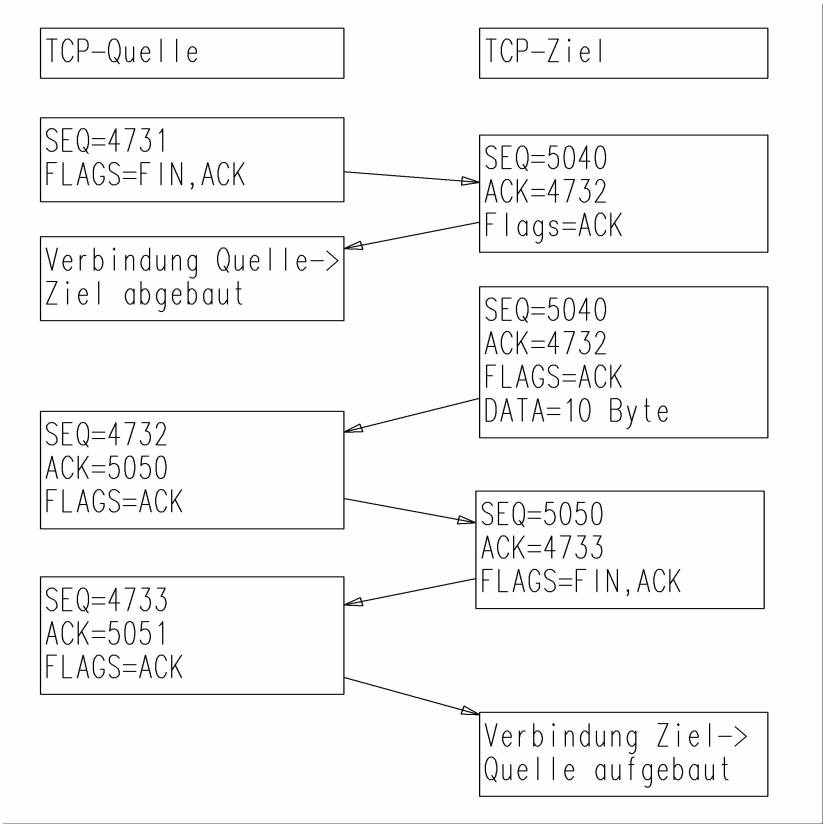
Verbindungsaufbau

Der Verbindungsaufbau wird als 3-Wege Handshake bezeichnet. Der Sender setzt eine Sendefolgenummer (SEQ=4711) und setzt das SYN Flag. Nimmt der Empfänger den Verbindungswunsch an, sendet er seine Sequenzfolgenummer (SYN=5010) mit dem SYN und ACK Flag und der Empfangsfolgenummer (ACK=4711+1). Diese gibt die nächste vom Sendefolgenummer an, die der Empfänger vom Sender erwartet. Ein drittes Paket vom Sender mit SEQ=4712 und ACK=5010+1 zeigt an, dass die Verbindung aufgebaut ist.



Verbindungsabbau

Der Datentransfer wird für beide Richtungen unabhängig voneinander beendet, was den Austausch von 4 TCP- Segmenten erfordert.



Der Sender sendet ein Paket mit dem FIN Flag. Danach kann der Empfänger noch weiter Daten senden TCP erlaubt auch das gleichzeitige Öffnen und Schließen einer Verbindung.

Flusskontrolle

TCP arbeitet mit Hilfe eines Schiebefensterprotokolls (sliding window). Mit der Quittierung bis zu einer Byteposition wird gleichzeitig ein Fenster weiter geöffnet, bis zu dem die Gegenseite Daten senden darf ohne auf eine Quittung warten zu müssen. Ein Problem bei einer Flusskontrolle ist, zu entscheiden, wann die Verbindung wegen fehlender Quittungen als unterbrochen erklärt werden soll. Hier für wird die Paketumlaufzeit (RTT, Round Trip Timer) laufen gemessen und ein gleitender Mittelwert (SRTT, Smoothed RTT) gebildet. Aus diesem wird der Retransmission Timeout (RTO) berechnet, nachdem das Paket nochmals gesendet wird. Erhält man darauf immer noch keine Quittung wird der RTO vergrößert. Hat sich dies mehrfach wiederholt, wird die Verbindung als unterbrochen erklärt. Da TCP nur eine Ende zu Ende Verbindung ist, kann aus dem RTT nicht auf eine Überlast im Netz geschlossen werden. Die Überlaststeuerung kann daher nur über den Fenstermechanismus bewirkt werden. Die dabei genutzten Verfahren sind:

- **Slow- Start- Algorithmus**: Es wird mit einer kleinen Fenstergröße begonnen die allmählich gesteigert wird, wenn die Datenrate mit steigt.
- **Congestion Avoidence Algorithmus**: Man geht davon aus, Pakete durch Überlast (congestion) verloren gehen. Fehlende Quittungen führen zu einer Reduktion der Senderate.
- Nagle Algorithmus: Hier werden die Nutzdaten bis zu einer fälligen Quittung gesammelt und diese dann mit dem Quittungssegment übertragen,

Timer

Hat der Sender eine Nachricht gesendet und die Quittung geht verloren, entsteht ein Deadlock weil der Sender auf die Quittung wartet und der Empfänger auf weitere Daten. Dies ist nur durch einen Timer aufzulösen. Nach einer gewissen Zeit werden die letzten Daten bzw. Quittungen wiederholt.
Um sich an Veränderungen im Netz anzupassen, wird die Datenübertragung bei TCP durch verschiedene Timer überwacht.

Retransmission Timer:
: Steuert die Wiederholung von Segmenten, die nicht bestätigt werden.

Persist Timer:
: Zur periodischen Abfrage der Fenstergröße eines nicht bereiten Empfängers. Hierbei wird auch ohne Verlängerung des Fensters ein kleines Paket gesendet. Dies wird entweder vom Empfänger mit einem neuen Fenster positiv quittiert oder die Acknowlegement Number wird in der Quittung nicht erhöht, weil der Empfänger keinen Puffer mehr frei hat.

Keepalive Timer:
: Überprüfung der Erreichbarkeit nach längeren Kommunikationspausen.

Quit Timer:
: Stellt sicher, dass nach Neustart eines Endsystem für die Dauer der MSL (Maximum Segment Lifetime) keine TCP Verbindung aufgebaut wird. Damit wird eine Interaktionen mit alten Segmenten vor dem Neustart verhindert.

2MSL Timer:
: Wartet beim Verbindungsabbau das doppelte der MSL ab, um einen möglichen Verlust des letzten ACK Segments zu verhindern.

UDP

UDP (User Datagram Protocol) ist ein verbindungsloses unzuverlässiges Protokoll auf der Transport Ebene. UDP wird bei Anwendungen genutzt, wo eine hohe Datenrate erforderlich ist, z.B. Echtzeit Audio oder Video-Übertragungen. Hier macht das Wiederholen eines verlorenen oder fehlerhaften Pakets sowieso i.A. keinen Sinn.

Source Port	Destination Port
Länge	Prüfsumme

UDP –Header

Zusammenwirken mit Protokollen der Sicherungsschicht

Durch verschiedene Protokolle wird das Zusammenwirken von IP und der Sicherngsschicht (Ethernet) geregelt.

ARP

ARP (Address Resolution Protocol) übersetzt in einem physischem Teilnetz IP-Adressen in Hardware (MAC) Adressen. Ein Host A sendet mittels Broadcast die zu übersetzende IP-Adresse an alle Teilnehmer. Wenn Host B darin seine eigene IP- Adresse erkennt, sendet er seine MAC- Adresse als Antwort. Diese Information speichert A in seinem Cache.

RARP

Mit RARP (Reverse Address Resolution Protocol) findet ein Host zu seiner Hardware (MAC) Adresse die IP-Adresse. Host A sendet mittels Broadcast eine MAC Adresse an alle Stationen. Ist im Netz ein RARP- Server aufgesetzt, kennt dieser die IP-Adressen der

Teilnehmer, und sendet sie als Antwort an A. So kann auch ein Rechner ohne Fetplatte nach einem Neustart sich ins Netz einklinken.

DHCP

DHCP (Dynamic Host Configuration Protocol) löst das Problem, der Vergabe einer IP-Adresse, wenn einem Teilnehmer keine feste IP- Adresse zugeordnet wird. Ein DHCP-Server kann einer Station entweder eine der MAC Adresse fest zugeordnete IP- Adresse liefern oder sie stellt einer Station für eine gewisse Zeit aus einem Pool eine IP- Adresse zur Verfügung. Der Client macht per broadcast eine DHCP- Anfrage. Ist ein DHCP- Server im Netz vorhanden, liefert die DHCP Antwort alle für den Netzbetrieb erforderlichen Informationen, wie:

- Zugeordnete IP- Adresse des Client
- Subnetz-Maske
- IP- Adresse des Gateways (Router)
- IP- Adresse des Nameservers
- IP- Adresse des DHCP- Servers
- Servername
- Name einer Initialisierungsdatei

Ferner wird dem Client eine Lease- Dauer mitgeteilt. Für diesen Zeitraum ist die IP- Adresse gültig. Benötigt der Client sie länger muss er mit einem neuen Request diese Zeit verlängern. DHCP wird benutzt bei Funk-LANs und von Internet-Providern für über Modems angeschlossene Benutzer.

SLIP

SLIP (Serial Line IP) stellt ein Protokoll zur Übermittlung von IP- Paketen auf einer seriellen Leitung da. SLIP ist zeichenorientiert und stellt die beiden Steuerzeichen zur Verfügung:

```
END  C0
ESC  DB
```

Das IP- Paket wird in einen Rahmen mit dem Zeichen END am Anfang und am Ende eingefügt. Komm das Zeichen C0 im IP- Paket vor wird es durch DB DC ersetzt. Das Zeichen DB (Esc) wird ersetzt durch DB DD. Damit kann im IP- Paket jede Bytefolge übertragen werden.

SLIP ist zwar sehr einfach es erlaubt jedoch keine Fehlererkennung und es versagt, wenn ein Nutzzeichen in ein END oder ESC Zeichen umgewandelt wird.

PPP

PPP (Point to Point Protocol) ist eine verbessertes Protokoll zur Übermittlung von Datenpaketen über eine physikalische oder virtuelle Punkt- zu- Punkt Verbindung. Eine PPP Dateneinheit beinhaltet folgende Daten:

- Protocol (2Byte) Angabe des Protokolls der Nutzdaten
- Information
- Padding Füllbytes

Diese nackten PPP Dateneinheiten enthalten keine Bits zur Synchronisation und zur Vermeidung von Gleichstromanteilen. Daher werden sie in einen HDLC (High Data Link Control) Frame eingebettet.

Der HDLC- Frame enthält folgende Felder:

- Flag (0111110) Kennzeichen Frame Anfang und Ende.
- Address (11111111) All Station, da Punkt zu Punkt Verbindung
- Control-Field (00000011) keine Nummerierung
- PPP- Dateneinheit
  Hier wird dafür gesorgt, dass die Bitkombination 01111110 nicht in den Daten vorkommt. Nach der Bitkombination 11111 (d.h. 5 aufeinander folgende 1), wird eine 0 eingefügt. Diese wird vom Empfänger wieder entfernt.
- Frame Check Sequence (FCS) Prüffolge, die auf einem CRC Code basiert.

Eine PPP Verbindung muss auf- und abgebaut werden. Die Zustände der Verbindung sind:

- Dead
  Anfangs und Endzustand
- Establish
  Zustand beim Aufbau
- Authenticate
  In diesem Zustand erfolgt die Authentisierung.
- Network
  Hier werden die Datenpakete übertragen
- Terminate
  Zustand beim Abbau der Verbindung

Der Übergang in einen neuen Zustand erfolgt jeweils durch spezielle Pakete des Network Control Protocols (NCP) , des Link Control Protocols (LCP), des Password Authentication Protocol (PAP) bzw. des Challenge Handshake Authentication Protocol (CHAP).
Das Challenge Handshake Authentication Protocol (CHAP) wird benutzt, damit nicht die Passwörter über die Leitung geschickt werden müssen. Beide Stationen benutzen die gleiche Einweg- Hashfunktion H. Der Server erzeugt eine Zufallsfolge X, die er dem Clienten schickt. Der Client erzeugt aus X und dem Passwort P eine Zahl A=H(X, P), die er dem Server schickt. Der Server erzeugt ebenfalls eine Zahl B=H(X, P). Es ist nur dann A=B, wenn der Client das richtige Passwort benutzt hat. Mit diesem Challenge Handshake können sich jeweils 2 Stationen ihres Gegenübers versichern, ohne dass ein Passwort über die Leitung geschickt werden muss.