TCP/IP Protokolle

Inhaltsverzeichnis

TCP/IP Protokolle

(Eine Ergänzung zu Tanenbaum Computernetzwerke)

Wozu Protokolle:

Computeranwendungen kommunizieren mittels sogenannter *Protokolle*. Diese Protokolle sind Regeln, welche den Nachrichtenaustausch zwischen den einzelnen Anwendungen koordinieren.

Aufgrund der Komplexität der Kommunikation zwischen EDV- Anlagen ist es aber nicht sinnvoll, alle notwendigen Aufgaben in einem einzelnen Protokoll abzuwickeln. Es kommen daher meistens *mehrere* Datenübertragungsprotokolle *gleichzeitig* zum Einsatz. Sie arbeiten in Form von mehreren sich überlagernden Protokollschichten mit unterschiedlichen Funktionen zusammen, wobei sie gemeinsam eine Dienstleistung für den Benutzer erbringen.

Das OSI Referenzmodell:

Damit hinsichtlich der Funktionalität der einzelnen Schichten eine einheitliche Betrachtungsweise existiert, hat die ISO (International Standardisation Organisation) das Modell einer Protokollschichtung entworfen, das ***7-Schichten Referenzmodell***.

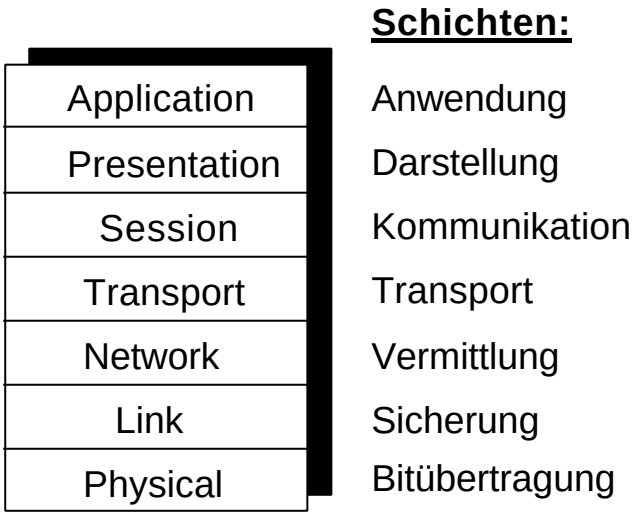


Bild: OSI 7-Schichten Referenzmodell

Es bildet die Grundlage für den Entwurf und die Standardisierung der von der ISO genormten eigenen Protokollschichten. In der Regel hat jede Protokollsammlung (wie auch TCP/IP) ihre eigene Architektur. Sie können aber mit der ISO-Terminologie verglichen werden, da diese allgemein bekannt und anerkannt ist.

Die Schichtung beruht auf dem Prinzip, dass eine Schicht die Dienstleistung der nächsten tieferen Schicht in Anspruch nimmt, ohne zu wissen, wie diese ihre Dienstleistung erbringt. Es wird also der jeweils höheren Schicht eine genau definierte Dienstleistung angeboten. Auf diese Weise wird eine Arbeitsteilung innerhalb der Schichten erreicht.

Ziele der TCP/IP Architektur:

1974 wurden in einem Artikel von V. Cerf und R. Kahn zum ersten mal die Grundzüge der TCP/IP Protokolle und der Netzknoten-Architektur niedergelegt.

- Unabhängigkeit von der verwendeten Netzwerk-Technologie sowie der Architektur der Hostrechner
- universelle Verbindungsmöglichkeiten im gesamten Netzwerk
- Ende-zu-Ende Quittungen
- standardisierte Anwendungsprotokolle

Als Hauptmerkmale der TCP/IP Architektur, die die obengenannten Ziele unmittelbar umsetzen, können folgende Punkte genannt werden:

- verbindungsloses Protokoll auf der Vermittlungsebene
- Netzknoten als Paketvermittlungrechner
- Transportprotokolle mit Sicherungsfunktionen
- einheitlicher Satz von Anwendungsprogrammen

<u>Vergleich der OSI und der TCP/IP Architektur:</u>

Weil die Erfahrungen des TCP/IP Projektes in die OSI-Standardisierung eingeflossen sind, können die beiden Modelle gut verglichen werden.

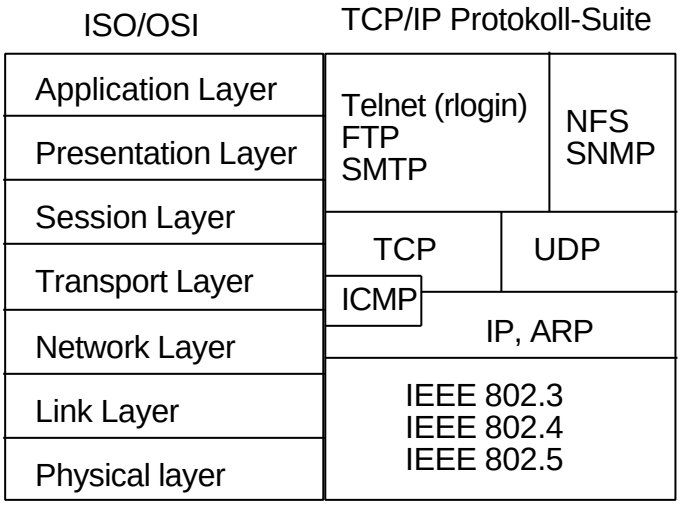


Bild: OSI Modell verglichen mit TCP/IP Modell

Die beiden Modelle unterscheiden sich oberhalb der Transportebene. Das ISO Modell besitzt hier noch zwei zusätzliche Schichten, die Kommunikationsschicht und die Darstellungsschicht. Die TCP/IP Architekten betrachteten dagegen die Funktionen dieser beiden Schichten als Teil der Anwendungsschicht.

Aus der Forderung nach Unabhängigkeit vom Uebertragungsmedium resultiert, dass auf der Netzwerkebene von TCP/IP die späteren OSI Schichten Bitübertragung und Sicherung vereinigt wurden.

Der Forderung nach universellen Verbindungsmöglichkeiten trägt das IP (Internet Protocol) Rechnung. Auf der Netzwerkebene von TCP/IP existiert nur **ein einziges** Protokoll, das alle am Netzwerk beteiligten Hosts und Knoten verstehen können.

Das ICMP (Internet Control Message Protocol), Bestandteil jeder IP-Implementierung, transportiert Fehler- und Diagnose- Informationen für IP.

Auf der Transportebene werden neben den am weitesten verbreiteten Protokollen TCP (Transport Control Protocol) und UDP (User Datagram Protocol) im Internet noch weitere Protokolle betrieben, etwa zu Forschungszwecken.

Die TCP/IP Schichten

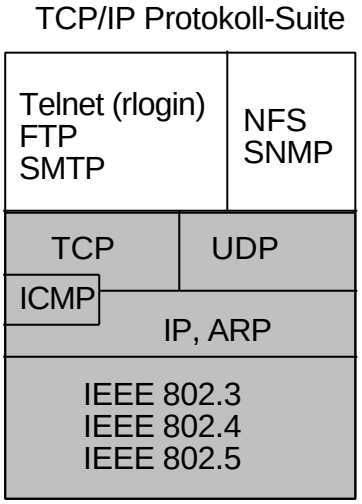


Bild: die Schichten der TCP/IP Architektur

Das Internet Protocol:

Das IP-Protokoll ist die Grundlage der TCP/IP Architektur. **Alle** Rechner im Internet verstehen IP. Die Adressierung und das Fragmentieren von Paketen gehört zu den Hauptaufgaben des IP. Es enthält keine Flusskontrolle.

Merkmale des IP sind:

- verbindungsloses (CLNS) Protokoll
- Fragmentiert (zerteilt) die Pakete bei Bedarf
- Adressierung durch 32-Bit Internet Adressen
- maximal 65535 Bytes Paketgrösse
- 8-Bit Transportprotokolladressen
- ermittelt lediglich eine Kopfprüfsumme, keine Datenprüfsumme
- nicht ständig benötigte Protokollfelder sind optional
- endliche Lebensdauer eines Paketes
- "Best Effort" Zustellung

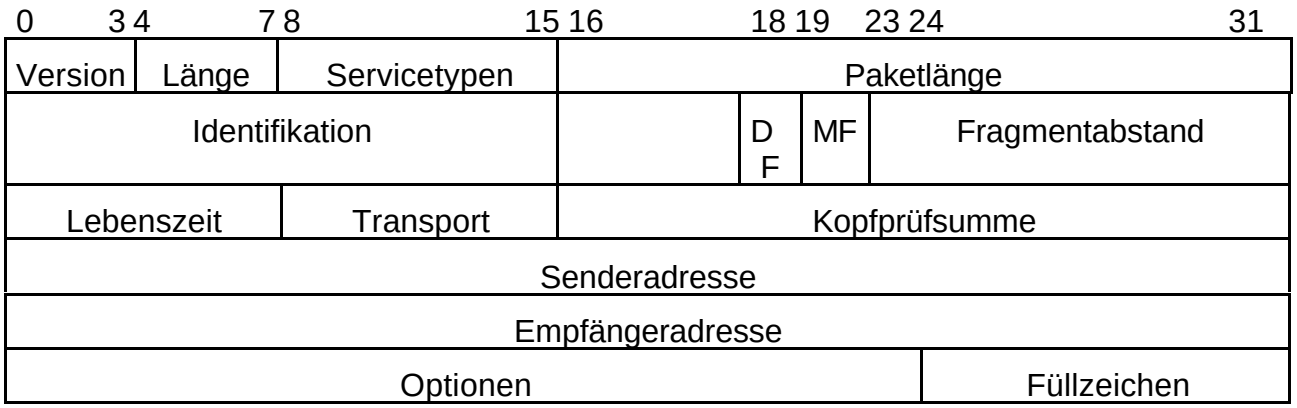


Bild: IP Protokollkopf

Versionsnummer: (4 Bits)

Enthält in 4 Bits die Versionsnummer des IP-Protokolles. Derzeitige Versionsnummer ist 4.

Länge: (4 Bits)

Länge des Paketkopfes in 32 Bit Worten. Weil sich durch Anfügen von Optionsfelder die Länge ändern kann, muss sie angegeben werden

Servicetypen: (8 Bits)

Sie geben die Kriterien an, wie der IP-Protokollautomat die Nachricht zu behandeln hat. In der Praxis sind sie meistens 0.

Bits	**Wenn 0**	**Wenn 1**
0-2	Priorität	
3	Normale Wartezeit	Niedrige Wartezeit
4	Normaler Durchsatz	
5	Normale Zuverlässigkeit	
6-7	Reserviert	

Tabelle: Aufbau des Sevicetypenfeldes

Praktisch keine Unix-IP-Implementierung wertet diese Felder aus, weil es keine Programmierschnittstelle zum Ausfüllen dieses Feldes gibt.

Paketlänge: (16 Bits)

Gibt die Gesamtlänge des IP Paketes an, inklusive des Protokollkopfes. Die IP-Spezifikation legt fest, dass jeder Host in der Lage sein muss, Pakete bis zu 576 Bytes Länge zu empfangen und zu reassemblieren (zusammenstückeln, Stichwort: Fragmentierung, siehe weiter unten). In der Regel können Rechner aber wesentlich grössere Pakete, mindestens aber bis zur Maximalgrösse des angeschlossenen Netzwerkes, z.B. eines Ethernet Paketes verarbeiten.

Identifikation: (16 Bits)

Eindeutige Identifikation, welche z.B. durch einen Zähler am absendenden Host vergeben wird. Diese Identifikation wird bei der Reassemblierung von Fragmenten verwendet, um alle Teile einer Fragmentkette identifizieren zu können.

Flags: (2 Bits)

Diese 2 Bits steuern die Behandlung des Paketes im Falle einer Fragmentierung. Ist das Bit DF: "don't fragment" gesetzt, darf das IP-Paket unter keinen Umständen fragmentiert werden, auch wenn es dann nicht mehr

weitertransportiert werden kann und weggeworfen werden muss. Das Bit MF: "more fragments" zeigt an, ob dem IP-Paket weitere Teilpakete nachfolgen.

Fragmentabstand: (13 Bits)

Ist das MF Bit gesetzt, gibt der Fragmentabstand die Lage der in diesem Paket gespeicherten Teilnachricht relativ zum Beginn der Gesamtnachricht an. Mit dieser Angabe kann der empfangende Host das Originalpaket wieder zusammensetzten. Der Abstand wird in Einheiten von 8 Bytes gezählt.

Lebenszeit: TTL - Time To Live (8 Bits)

Hier wird angegeben, wielange ein Paket im Netz verweilen darf, bevor es weggeworfen werden muss. RFC791[2] spezifiziert die Einheit in Sekunden. Jeder Netzknoten verringert diesen Wert um 1. Die Lebenszeit (TTL) ist gleichbedeutend mit der Anzahl der Netzknoten, die von einem Paket maximal durchlaufen werden können. Enthält das Feld den Wert 0, muss es vom verarbeitenden Rechner weggeworfen werden. Somit wird verhindert, dass ein Paket endlos im Netz zirkuliert. Der Absender des Paketes erhält in diesem Fall eine ICMP-Nachricht über den Vorgang. Unix Rechner setzen diesen Wert in der Regel auf einen Wert zwischen 15 (4.2BSD) und 30 (4.3BSD)

Transport: (8 Bits)

Gibt die Identifikation des Transportprotokolls an, dem dieses Paket zugestellt werden muss.

Protokoll	**Nummer**
ICMP	1
TCP	6
UDP	17

Tabelle: Transportprotokollnummern

Diese Nummern werden in regelmässig erscheinenden RFC's festgelegt. Derzeit existieren ca. 50 offizielle Protokolle.

Kopfprüfsumme: (16 Bits)

Dieses Feld enthält die Prüfsumme der Felder im Protokollkopf. Ein Netzknoten oder ein Host kann mit der entsprechenden Gegenprüfung verhindern, dass er mit verfälschten Daten arbeitet. Die Nutzdaten des IP-Paketes werden aus Effizienzgründen nicht geprüft. (Sie werden beim Empfänger innerhalb des Transportprotokolles geprüft.) Da das IP-Paket in jedem Netzknoten verändert wird, ist eine effiziente Bildung dieser Prüfsumme sehr wichtig.

Die *Internet-Prüfsumme* ist in allen TCP/IP Protokollen gleich:

Das 1er Komplement der 16-Bit-Summe aller 16-Bit-Worte der zu überprüfenden Daten

Die 16 Bits der Kopfprüfsumme werden dabei als 0's angenommen. Die Fehlerer-kennungsfähigkeiten sind allerdings begrenzt, weil nur einfache Addition verwendet wird. Da in die Datenprüfsumme von TCP und UDP neben den Daten weitere Angaben wie Länge, Absender usw. einbezogen werden, sind nichterkannte Datenverluste praktisch auszuschliessen.

Senderadresse und Empfängeradresse: (je 32 Bits)

In diesen Feldern sind die 32 Bit langen Internet Adressen eingetragen. Der Aufbau dieser Adressen wird weiter unten näher erklärt.

Optionen und Füllzeichen:

Für Spezialaufgaben wie z.B. Netzwerkmanagement, Sicherheit usw. kann der IP-Kopf um Optionen erweitert werden. Die Füllzeichen werden eventuell benötigt, um die Zahl der 16-Bit-Worte des IP-Kopfes auf ein Vielfaches von 4 zu bringen.

Adressierung in einem IP Netzwerk

Bei der Adressierung eines Kommunikationspartners müssen beim Durchlaufen der ersten 4 Schichten auch 4 verschiedene Adressen und Identifikationen angegeben werden:

- eine Subnetzwerk-Adresse (z.B. Ethernet-Adresse)
- eine Internet-Adresse
- eine Transportprotokoll-Identifikation
- eine Portnummer

Die Internet und die Transportprotokollidentifikation finden sich als Felder im IP-Protokollkopf wieder. Dabei ist die Internet-Adresse von grösserer Bedeutung, weil jeder Knoten im Internet eine oder mehrere solcher eindeutiger Adressen trägt.

Adressierung auf der IP Ebene

Damit ein Kommunikationspartner im Internet adressiert werden kann, wird ein 32 Bit Wert (4 octets = 4 Bytes) als Adresse verwendet. Diese Adresse enthält die Netzwerk-Nummer (network number) und die Rechner-Nummer (host number). Die Netzwerknummer ist für alle Hosts im selben Netzwerk gleich. Sie darf nur einmal verwendet werden. Im Gegensatz dazu muss jeder Host im selben Netzwerk eine eigene, eindeutige Nummer haben.

Weil eine Internet Adresse verschieden dargestellt werden kann, soll im Folgenden diese Notation verwendet werden:

Die Internet Adresse wird in 4 Felder à 8 Bits aufgeteilt. Jedes Feld beschreibt einen 8 Bit Wert zwischen 0 und 255 (dezimal). Die einzelnen Felder werden dabei mit einem Punkt getrennt. Beispiel: 124.96.18.0

Aufteilung der Netze in Klassen

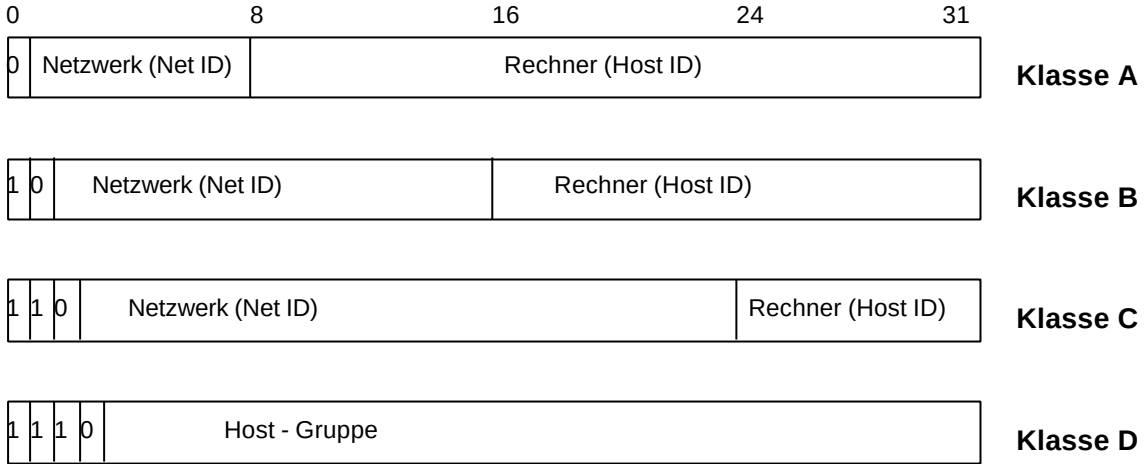


Bild: Netz Klassen

Am Anfang des ARPANETs gab es nur Klasse A Netzwerke, da man annahm, dass es nur wenige, grosse Netzwerke geben würde. Als aber in vielen Organisationen lokale Netze eingeführt wurden, stimmte diese Annahme nicht mehr. Man entschloss sich deshalb, weitere Klassentype einzuführen. Die Klasse B ist für mittelgrosse Netzwerke, Klasse C für kleine Netzwerke mit wenigen Hosts. Die Klasse D dient seit einigen Jahren dem *"Multicasting"*, d.h. IP-Pakete werden gleichzeitig an eine Gruppe von Hosts verteilt.

Klasse E sei hier nur der Vollständigkeitshalber erwähnt. Hier sind die ersten 4 Bits auf 1 gesetzt. Diese Klasse ist ausschliesslich für die Forschung reserviert.

Die Netzklassen:

Klasse	max Netze (weltweit)	max. Hosts	Internet Adressen	Bemerkung
A	126	>16'000'000	1 - 126 (Adresse 127 für Loopback reserviert)	Wenige Netze mit sehr vielen Hosts möglich
B	>16'000	65536	128.0 - 191.254	Viele Netze mit vielen Hosts
C	> 2'000'000	254	192.0 - 223.255.254	Sehr viele kleine Netze mit nur wenigen Hosts möglich

Tabelle 4: Uebersicht der Netzklassen

Welche Klasse für ein bestimmtes Netzwerk verwendet werden soll, richtet sich nach der Zahl der vorhandenen Hosts und der Netzwerke.

Bei der Vergabe von Internet Adressen ist ausserdem zu berücksichtigen, das Netzwerk und Host ID's mit allen Bits auf 0 oder 1 reserviert sind und deshalb nicht vergeben werden dürfen.

Soll das Netzwerk an fremde Netzwerke angeschlossen werden, so muss eine registrierte Netzwerknummer verwendet werden. Weltweit werden die Netzwerknummern vom Network Information Center (NIC) verteilt. Das NIC ist unter der folgenden Adresse erreichbar:

DNN Network Information Center

SRI International

Room EJ286

333 Ravenswood Avenue

Menlo Park, CA 94025

Subnetzwerke, Netzwerk-Maske

Um das Routing innerhalb von grossen Netzwerken (Klasse A und B) zu ermöglichen, kann ein Netzwerk in sogenannte Subnetzwerke aufgeteilt werden. Dabei wird der Host-Teil der Internet Adresse in eine Subnetzwerknummer und die eigentliche Stationsnummer unterteilt. Mit Hilfe der Subnetzwerknummer kann nun innerhalb des Netzwerkes in interne Unternetze verzweigt werden. Von aussen ist aber das gesamte Netzwerk als eine Einheit sichtbar.

Die Netzwerk Maske gibt dem System an, welche Bits der Internet Adresse als Netzwerk / Subnetzwerk Adresse und welche Bits als Host Adresse verwendet werden. Die Netzwerk Maske ist wie die Internet Adresse 32 Bit lang; jedes Bit der Maske entspricht einem Bit der Internet Adresse.

Für die Bestimmung der Netzwerk Maske geht man wie folgt vor:

- für jedes gesetzte Bit (binär 1) der Netzwerk Maske gilt, dass das entsprechende Bit der Internet Adresse als Teil des Netzwerkes / Subnetzwerkes interpretiert wird.
- jedes gelöschte Bit (binär 0) wird als Teil der Host Adresse interpretiert.

Einfachheitshalber werden als Netzwerk Maske meistens die Werte 255 (binär 1111'1111) oder 0 verwendet. Mit diesen Werten lässt sich der Netzwerk / Subnetzwerkteil besser vom Hostteil unterscheiden.

Je nach der Klasse des Netzwerkes muss das erste (Klasse A), die ersten beiden (Klasse B) oder die ersten drei Bytes der Netzwerk Maske angegeben werden, da sie die Netzwerk Adresse angeben. Die verbleibenden Bits können je nach Subnetzwerk gesetzt oder gelöscht werden.

Beispiele von Netzwerken/Subnetzwerken und Netzwerkadressen:

Klasse A Netzwerk:

	1. Byte	2. Byte	3. Byte	4. Byte
Klasse A Netzwerk	Netzwerk-nummer (1-126)	Hostnummer		
Netzwerk Maske	255	255	0	0
Subnetzwerk	Netzwerk-Nummer	Subnetzwerk-Nummer	Hostnummer	

Für ein Klasse A Netzwerk setzt man die Maske üblicherweise auf 255.255.0.0 oder 255.255.255.0. Ist bei einem Klasse A Netzwerk die Maske auf 255.255.255.0 gesetzt bedeutet das, dass das erste Byte die Netzwerk Adresse, das zweite *und* dritte Byte die Subnetzwerk Adresse und das vierte Byte die Host Adresse ist.

Klasse B Netzwerk:

	1. Byte	2. Byte	3. Byte	4. Byte
Klasse B Netzwerk	Netzwerknummer (129 - 191)		Hostnummer	
Netzwerk Maske	255	255	255	0
Subnetzwerk	Netzwerknummer		Subnetzwerk-Nummer	Hostnummer

Ist bei einem Klasse B Netzwerk die Maske auf 255.255.255.0 gesetzt, so geben das erste und das zweite Byte die Netzwerk Adresse, das dritte die Subnetzwerk Adresse und das vierte die Host Adresse an.

Normalerweise werden bei Netzen der Klasse C keine Subnetzwerke gebildet, da der Host Adressbereich mit 8 Bits zu klein ist.

Falls keine Subnetzwerke benötigt werden, sehen die Netzwerkmasken folgendermassen aus (default network mask):

Klasse A	255.0.0.0
Klasse B	255.255.0.0
Klasse C	255.255.255.0

Tabelle: Default Network Mask

Die Broadcast Adresse oder Broadcast Maske

Ein Vorteil des Internets ist, das mit der Internet Adresse nicht nur Netzwerke sondern auch einzelen Host direkt adressiert werden können. Normalerweise wird die Hostnummer 0 nicht an einen individuellen Rechner vergeben sondern bezeichnet das Netzwerk. Um nun eine Meldung an alle Hosts eines Netzwerkes zu versenden, wird nun eine Broadcast Maske oder Broadcast Adresse verwendet. Dabei wird die Broadcast Maske als Adresse interpretiert (to broadcast: weit verbreiten, senden). Alle Hosts in einem Netzwerk besitzen dieselbe Broadcast Adresse.

Die Broadcast Adresse setzt sich aus der verwendeten Netzwerk Adresse und der Host Adresse, bei welcher alle Bits gesetzt sind (binär 1), zusammen. (Ausnahme: bei den Betriebssystemen UNIX BSD 4.2 und ULTRIX vor Version 1.2 sind die Host Bits alle gelöscht).

Die Broadcast Adresse kann auch aus der Internet Adresse und der Netzwerk Maske berechnet werden:

broadcastaddress = (NOT networkmask) OR (internetaddress)

Beispiel: ist die Host Internet Adresse 147.88.96.1 und die Netzwerk Maske 255.255.0.0 dann folgt daraus die Broadcast Adresse 247.88.255.255.

Beispiele von Internet Adressen, Netzwerk- und Broadcast Masken:

Internet Host Adresse	Host Nummer	Netzwerk Klasse	Netzwerk Nummer	Netzwerk Maske	Broadcast Adresse
3.0.0.10	10	A	3	255.0.0.0	3.255.255.255
11.1.0.12	12.	A	11.1	255.255.0.0	11.1.255.255
129.39.0.15	15	B	129.39	255.255.0.0	129.39.255.255
128.45.2.8	8	B	128.45.2	255.255.255.0	128.45.2.255
192.0.1.8	8	C	192.0.1	255.255.255.0	192.0.1.255

Tabelle: Internet Adressen und Masken

Das Routing

In einem Internet Netzwerk ist es möglich, ein Datenpaket über viele Wege zu einem bestimmten Ziel zu versenden. In diesem Datenpaket sind ausser den Nutzdaten auch die Startadresse und die Zieladresse angegeben. Mit *Routing* bezeichnet man das Verfahren, welches beim Suchen und Auswählen des Weges zur Zieladresse angewendet wird.

Es gibt zwei Arten von Routing: *direktes* und *indirektes* Routing

Direktes Routing:

Sind der sendende Host und der Zielhost im selben Netzwerk, können die Datenpakete direkt ausgeliefert werden.

Indirektes Routing:

Falls der Zielhost nicht im selben Netzwerk wie der sendende Host ist, muss das Datenpaket geroutet werden. Zu diesem Zweck wird das Datenpaket zuerst zum lokalen Gateway geschickt. Dieser entscheidet, wohin das Datenpaket weitergeschickt werden soll. Das Datenpaket "hüpft" nun von einem Gatewayzum anderen, bis das Zielnetzwerk erreicht wurde. Dort sendet der Gateway das Datenpaket an den Zielhost.

Alle Hosts und Gateways in einem Netzwerk speichern Routinginformationen in Tabellen ab. Jeder Eintrag in einer dieser Tabellen enthält normalerweise:

- die Ziel-Internet-Adresse
- die Internet-Adresse des Gateways, welcher physikalisch ans gleiche Netzwerk angeschlossen ist.
- ein Flag, welches angibt, ob direkt oder indirekt geroutet wird.
- ein weiteres Flag, welches über die Art des Routings, Host- oder Network-Routing, informiert.

Einige Routingtabellen enthalten noch Informationen z.B. über die Performance (Leistungsfähigkeit) von Netzwerken usw.

Host- und Network- Routing:

Das Host- oder Network- Routing Flag bestimmt, wie ein Host oder ein Gateway die Zieladresse im IP-Paket mit der Zieladresse in der Routingtabelle vergleicht.

IP Paket:

Zieladresse
91.2.4.5

Routing Tabelle:

Zieladresse	Gateway Adresse	Host- oder Netzwerk-Routing?	direkt oder indirektes routing?
91.2.4.5	89.23.1.4	Host	indirekt

Tabelle: Host Routing

Der Host oder das Gateway vergleicht die ganze Zieladresse des IP Paketes mit den Einträgen in der Routing Tabelle. Beide Adressen müssen übereinstimmen um das IP Paket mit der definierten Routingart zu verwenden. (Tabelle 7)

IP Paket:

Zieladresse
91.2.4.5

Routing Tabelle:

Zieladresse	Gateway Adresse	Host- oder Netzwerk-Routing?	direkt oder indirektes routing?
91.0.0.0	89.23.1.4	Netzwerk	indirekt

Tabelle: Network Routing

Beim Netzwerk-Routing vergleicht der Host oder das Gateway nur den Netzwerk- oder den Subnetzwerkteil der Zieladresse des IP Paketes mit seiner Routingtabelle. Der Hostteil der IP Adresse ist irrelevant.(Tabelle 8)

Wie die IP Pakete gerouted werden:

Wenn ein Host oder ein Gateway ein IP Paket senden möchte, werden die folgenden Schritte durchgeführt:

1. Zuerst wird die Routingtabelle nach einem gleichen Eintrag mit *Host Routing* gesucht. Wird kein Eintrag gefunden, wird mit Schritt 2 weitergefahren. Falls ein Eintrag gefunden wird, kann mit Schritt 4 fortgefahren werden.
2. Die Routingtabelle wird nach einem Eintrag mit *Netzwerk Routing* abgesucht. Ist kein Eintrag vorhanden, so wird mit Schritt 3, sonst mit Schritt 4 fortgefahren.
3. Es wird die Routingtabelle nach einem voreingestellten Router abgesucht. Wird ein Router gefunden, so kann mit Schritt 4 weitergemacht werden, ansonsten wird das IP Paket gelöscht. Ist der Host ein Gateway, so wird mittels des ICMP Protokolls dem Absender des Paketes mittgeteilt, dass kein Weg zum Ziel bekannt ist.
4. Nun wird das direkt/indirekt Flag getestet. Falls das direkte Routing möglich ist, wird das IP Paket direkt gesendet. Kann nur indirekt geroutet werden, so wird das IP Paket zu dem in der Routingtabelle definiertem Gateway geschickt.

Verwaltung der Routing Tabellen:

Es gibt mehrere Verfahren, um in Gateways Roting Tabellen aufzubauen:

- Feste (statische) Routingtabelle, welche bei der Konfiguration des Netzwerkes erstellt wird. Diese Methode ist aber unflexibel bei Veränderungen im Netzwerk.
- Laden der Tabelle während des Betriebes durch eine zentrale Routing-Leitstelle.
- Dynamisches Anpassen der Tabellen durch auswerten der vorbeifliessenden Nachrichten und/oder durch den Informationsaustausch mit den Nachbar Gateways. Diese Methode wird am häufigsten verwendet, weil sie eine rasche und selbständige Anpassung der Routingtabelle an veränderte äussere Umstände ermöglicht.

Statisches Routen:

Die Routingtabelle wird beim Aufstarten des geladen, z.B. bei einem Unix Host die Tabelle in */etc/gateways* . In diesem File können beliebige Zuordnungen von Netzwerken und Router eingetragen werden. Bei einfacheren Systemen (z.B. MS-DOS) kann nur der Default-Router angegeben. D.h. alle Pakete, welche nicht direkt versendet werden können, werden an den Default-Router geschickt. Dieser versendet nun das IP Paket an den zuständigen Host oder Router.

Dynamisches Routen:

Der Host hört die Leitung nach Paketen mit Routing Informationen ab. Falls der Host ein Internet-Router ist, so versendet er periodisch eine Kopie seiner Routing Tabelle zu allen Host, die er direkt erreichen kann (Broadcast).

Das am Meisten verwendete Protokoll für den Austausch von Routing Informationen ist das RIP Protokoll.

Praktische Beispiel eines Routing Vorganges:

Anhand des in Bild 6 wollen wir den Weg eines IP Paketes von Host *Bart* zu Host *Trick* verfolgen:

1. Host Bart 128.40.0.20 möchte ein Paket zum Host Trick 130.30.0.31 verschicken.
2. Host Bart stellt nun mit Hilfe der Netzwerkmaske fest, dass sich der Zielhost nicht im gleichen Netzwerk befindet:

   Meine.Adresse AND Netzwerkmaske ≠ Ziel.Adresse AND Netzwerkmaske
3. Jetzt sucht sich Host Bart aus der Routingtabelle die Adresse des Routers Simpson (128.40.0.10) für das Netzwerk 130.30.0.0.
4. Mit Hilfe des ARP Protokolls (Address Resolution Protocol) wird nun die Ethernet Adresse des Routers Simpson bestimmt.
5. Host Bart (128.40.0.20) versendet nun das Paket mit mit der Ethernet-Adresse des Routers Simpson und der IP Adresse des Zielhosts Trick (130.30.0.31).
6. Das Gateway Simpson (128.40.0.10) empfängt nun das Paket, weil ja seine Ethernet-Adresse angegeben wurde, stellt aber anhand der IP Adresse fest, dass das Paket nicht für ihn bestimmt ist.
7. Mit Hilfe der Netzwerkmaske stellt das Gateway Simpson fest, dass der Zielhost sich im anderen Netzwerk befindet.
8. Gateway Simpson bestimmt mit dem ARP Protokoll die Ethernet-Adresse des Zielhosts Trick.
9. Jetzt versendet das Gateway das IP Paket mit der vorher bestimmten Ethernet-Adresse und der IP Adresse des Zielhosts Trick (130.30.0.31).
10. Host Trick empfängt das IP Paket mit seiner Ethernet-Adresse und stellt nun anhand der IP Adresse fest, dass das Paket für ihn bestimmt ist.

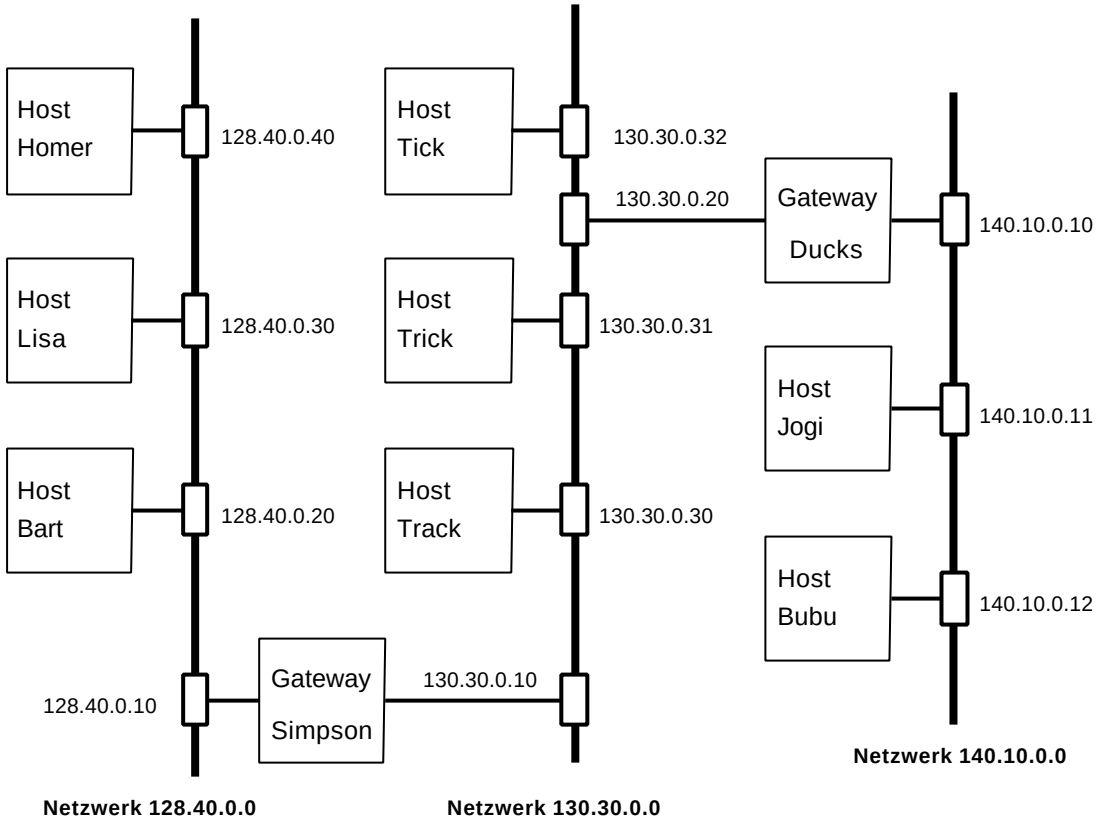


Bild: einfaches Netzwerk

Falls das IP Paket via mehrere Router versendet werden muss, so wiederholen sich die Schritte 2 - 6 je Router. Es kommt dabei nicht drauf an, ob es sich um ein Netzwerk oder ein Subnetzwerk handelt. Befinden sich zwei Netzwerke auf dem physikalisch gleichen Strang wird genau gleich vorgegangen. Der Router benötigt zwei Interfaces, welche je eine eigene IP Adresse ihres Netzes (Subnetzes) erhalten (Bild 7).

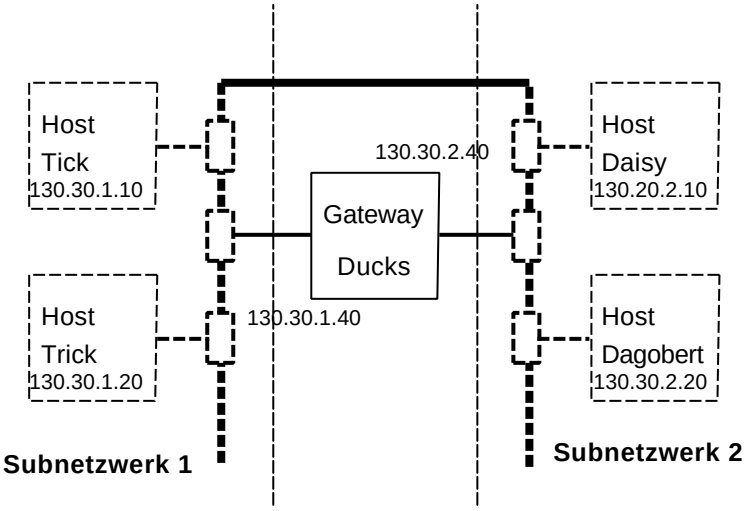


Bild: Zwei Netzwerke auf demselben physikalischen Netzwerk

TCP

Das TCP Protokoll liegt oberhalb des IP Protokolls und befindet sich auf der Transportschicht. Seine Hauptaufgabe ist der sichere Transport der Daten durch das Netzwerk. Die Transportadresse ist im IP Protokollkopfe 6.

Merkmale von TCP:

- erstellt eine full-duplex-fähige bidirektionale virtuelle Verbindung
- die Datenübertragung erfolgt aus der Sicht des Benutzers als Datenstrom und nicht blockweise
- Sicherung der Datenübertragung durch Sequenznummern, Prüfsumme, Quittierung mit Zeitüberwachung und Segmentwiederholung
- Sliding Window Funktionsprinzip
- Urgent Data und Push Funktion
- Transportbenutzeradresse durch 16 Bit Portnummer

TCP ist Byteorientiert; fast alle im Protokollkopf verwendeten Felder rechnen in Bytes und nicht in Blöcken. Das erlaubt eine flexible Dimensionierung von Blöcken. Daten werden zwar in Form von Blöcken (Segmenten) übertragen, die Grösse der Segmente im Netzwerk wird aber durch Parameter wie Netzwerkauslastung, Fenstergrösse oder die Ressources eines der beiden Partner bestimmt.

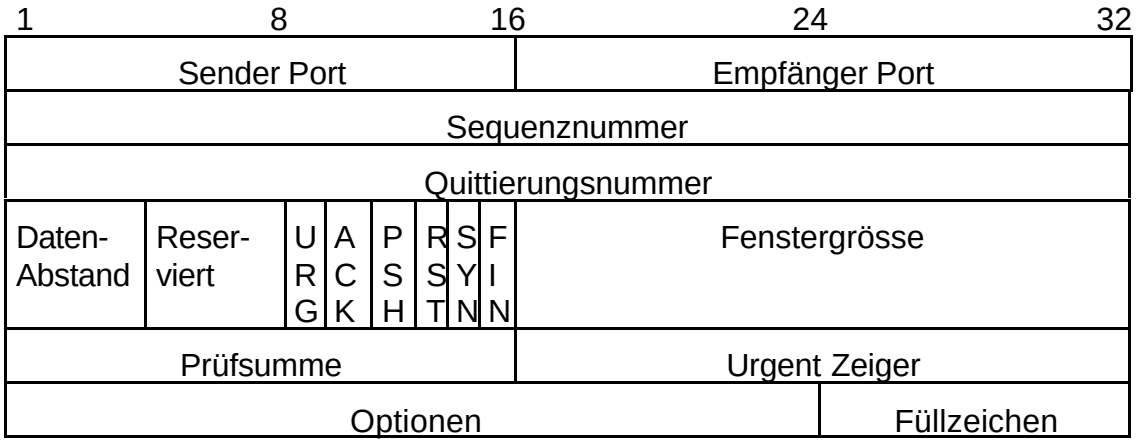


Bild:Der TCP Protokollkopf

Die Felder im Protokollkopf haben folgende Bedeutung:

- Sende und Empfangs-Port Nummer: Zwei je 16 Bit breite Felder, die die Endpunkte der virtuellen Verbindung Bezeichnen.
- Sequenz und Quittierungsnummer: Jeweils ein 32 Bit Wort, das die Stellung der Daten des während dieser Verbindungausgetauschten Datenstroms angibt. Die Sequenznummer gilt in der Senderichtung, die Quittierungsnummer für die Quittierung. Jeder Partner einer TCP Verbindung generiert bein Aufbau der Verbindung eine Sequenznummer. Diese Nummer muss während der Lebenszeit den Paketes im Netz (IP Lebensdauer) einmalig sein. Sie werden am Anfang des Verbindungsaufbaus gegenseitig ausgetauscht und quittiert. Beim Datentransfer wird die Sequenznummer vom Absender um die Anzahl der bereits gesendeten Bytes erhöht. In der Quittierungsnummer wird vom Empfänger angegeben bis zu welchem Byte die Nachricht ordnungsgemäss empfangen wurde. Der grosse Zahlenraum von 2^32 bietet ausreichend Schutz vor Verdoppelung und Ueberlappung
- Datenabstand: Angabe der Länge des TCP Protokollkopfes in 32 Bit Worten zur Ermittlung, wann die Daten beginnen.
- Flags: Mit den Bits in diesem Feld können Aktionen im TCP Protokoll ausgelöst werden.

  - URG: Zeiger im Urgent Feld ist gültig.
  - ACK: Quittierungsnummer ist gültig
  - PSH: Daten in diesem Segment sollten sofort der Anwendung übergeben werden. Eine Quittierung für diese Segment bedeutet, dass alle Daten bis zu dieser Quittierungsnummer beim Kommunikationspartner angekommen sind.
  - RST: Rücksetzen der Verbindung oder Antwort auf ein ungültiges Segment.
  - SYN: Wunsch eines Verbindungsaufbaus, Segment muss quittiert werden.
  - FIN: Einseitiger Verbindungsabbau und Ende des Datenstroms aus dieser Richtung, Segment muss quittiert werden.
- Fenstergrösse: Gibt die Anzahl Bytes an, die der Empfänger in seinen Datenpuffern augenblicklich für diese Verbindung aufnehmen kann (Receive Window). Mit dieser kann der TCP-Empfänger den Datenfluss steuern. Die Fenstergrösse 0 den TCP-Sender stoppen. Durch langsames anheben wird der Datenfluss wieder in Gang gebracht. Verwendete Fenstergrössen sind 2048 Bytes (4.2 BSD) und 4096 Bytes (4.3 BSD). Die optimale Ermittlung der Fenstergrösse während des Datentransfers gehört zu den kompliziertesten Algorithmen einer TCP Implementierung.

Das TCP Protokoll arbeitet nach dem Prinzip des *Sliding Window* jede Seite einer Verbindung darf die Anzahl an Bytes senden, die im Fenster angegeben ist, ohne auf eine Quittierung warten zu müssen. Während des Sendens können gleichzeitig Quittierungen, welche wiederum neue Fenstergrössen einstellen, für die von er anderen Seite empfangenen Daten eintreffen. Auf diese Weise wird vermieden, dass der Sender nach jedem Segment auf das Eintreffen einer zugehörige Quittierung warten muss.

Durch gleichzeitiges Senden und Quittieren wird ein Parallelismus erreicht, der selbst bei Netzwerken mit hoher Laufzeit, wie z.B. bei Satellitenverbindungen, einen optimalen Datendurchsatz erzielt.

- Prüfsumme: Summiert Protokollkopf und Daten. Es wird dasselbe Verfahren wie bei der IP Prüfsamme angewendet.
- Urgent Zeiger: Ergibt zusammen mit der Sequenznummer einen Zeiger (Adresse) auf ein Datenbyte. Das damit definierte Datenbyte ist das Ende eines Nachrichtenabschnittes, die nachfolgenden Daten werden als wichtig (Urgent Data) gekennzeichnet.
- Optionen: Für TCP gibt es drei Optionen: *End Of Option List*, *No Operation* und *Maximum Segment Size* , die letzte Option wird beim Verbindungsaufbau verwendet, um die Bereitschaft für den Empfang von grösseren Segmenten als 536 Bytes anzuzeigen.

Portnummern

Wie erwähnt exisitieren zur Adressierung auf der Transportebene sogenannte Portnummern. Das Feld für die Portnummer ist 16 Bit gross, es lassen sich also theoretisch maximal 65535 verschiedene Verbindungen aufbauen. Der Gültigkeitsbereich einer Portnummer ist auf einen Host beschränkt. Die Netzwerknummer, Host ID und Portnummer spezifizieren einen Kommunikationsendpunkt der auch *Socket* gennant wird.

Analogie von Internet Socket und Telefonanlage:

Internet Socket	Telefon
Netzwerk Nummer	Ortsvorwahl
Host ID	Telefonnummer
Portnummer	Nebenstelle

Aehnlich wie beim Telefonieren muss vor dem Verbindungsaufbau die Adresse (Portnummer) des Anzurufenden bekannt sein. Beide Seiten müssen diese Nummer vorher eine Portnummer vereinbart haben. Der Anzurufende, auch *Server* genannt, wartet dann passiv auf den Verbindungsaufbau. Die Nummer des Anrufenden, *Client* genannt, ist irrelevant, solange der Server keine spezielle Portnummer für den Client vorschreibt.
Einige TCP/IP Applikationen wie z.B. Telnet oder FTP haben eine feste, allgemein bekannte Nummer. Auf einem UNIX Rechner ist die Liste der angebotenen Dienste in der Datei /etc/services enthalten.

In der Datei /etc/inetd.conf steht, welche Programme bei den verlangten Diensten gestartet werden.

Ausschnitt aus der Datei /etc/inetd.conf:

```
#
# Internet server configuration database
#
ftp   stream      tcpnowait_root_/usr/etc/ftpd__ftpd
telnet_stream_tcp_nowait_root_/usr/etc/telnetd_telnetd
shell_stream_tcp_nowait_root_/usr/etc/rshd__rshd
login_stream_tcp_nowait_root_/usr/etc/rlogind_rlogind
exec_stream_tcp_nowait_root_/usr/etc/rexecd__rexecd
# Run as user "uucp" if you don't want uucpd's wtmp entries.
#uucp_stream_tcp_nowait_root_/usr/etc/uucpd__uucpd
finger_stream_tcp_nowait_nobody_/usr/etc/fingerd_fingerd
#tftp_dgram_udp_wait_nobody_/usr/etc/tftpd__tftpd
tftp_dgram_udp_wait_root_/usr/etc/tftpd__tftpd -s /private/tftpboot
comsat_dgram_udp_wait_root_/usr/etc/comsat_comsat
talk_dgram_udp_wait_root_/usr/etc/talkd_talkd
ntalk_dgram_udp_wait_root_/usr/etc/ntalkd_ntalkd
echo_stream_tcp_nowait_root_internal
discard_stream_tcp_nowait_root_internal
chargen_stream_tcp_nowait_root_internal
daytime_stream_tcp_nowait_root_internal
time_stream_tcp_nowait_root_internal
echo_dgram_udp_wait_root_internal
discard_dgram_udp_wait_root_internal
chargen_dgram_udp_wait_root_internal
daytime_dgram_udp_wait_root_internal
time_dgram_udp_wait_root_internal
NSWS_stream_tcp_wait_root_/usr/lib/NextStep/NSWSd_NSWSd
#
#ypupdated/1        stream     rpc/tcp   wait   root   /usr/etc/rpc.ypupdated
rpc.ypupdated
rquotad/1      dgram   rpc/udp wait root /usr/etc/rpc.rquotad   rpc.rquotad
rstat_svc/1-3  dgram   rpc/udp wait root /usr/etc/rpc.rstatd    rpc.rstatd
rusersd/1-2    dgram   rpc/udp wait root /usr/etc/rpc.rusersd   rpc.rusersd
sprayd/1       dgram   rpc/udp wait root /usr/etc/rpc.sprayd    rpc.sprayd
walld/1        dgram   rpc/udp wait root /usr/etc/rpc.rwalld    rpc.rwalld
renderd/1            dgram      rpc/udp  wait  root  /usr/prman/rpc.renderd
rpc.renderd
```

User Datagram Protocol (UDP)

Das UDP Protokoll ist ein verbindungsloses Transportprotokoll. Seine Merkmale sind:

- Verbindungslos
- Adressierung durch Portnummern
- Prüfsumme der Daten
- äusserst einfach
- "Best Effort" Zustellung

Die Felder im UDP Protokollkopf sind:

0 15 16 31

Sender - Portnummer	Empfänger - Portnummer
Länge	Prüfsumme

Bedeutung der Felder:

- Sender- und Empfänger-Portnummer: Wie beim TCP Protokoll sind auch hier die Portnummern die Referenz zu den Protokollbenutzern.
- Länge: Enthält die Länge des gesamten Datagramms (Pakets), inklusive des Protokollkopfes.
- Prüfsumme: Enthält die Internet Prüfsumme der Daten und des Protokollkopfes. Ist in diesem Feld eine 0 eingetragen, so hat der Absender keine Prüfsumme eingetragen und der UDP-Empfänger macht ebenfalls keine Prüfung.

UDP liefert zu den Leistungen von IP hinaus lediglich eine Portnummer und eine Prüfsumme der Daten. Im Gegensatz zu TCP gibt es hier keine Transportquittierungen oder andere Sicherheistmassnahmen. Dadurch wird UDP besonders effizient und daher geeignet für Hochgeschwindigkeitsanwendungen wie z.B. NFS (verteilte Dateisysteme) oder dergleichen.

Internet Control Message Protocol (ICMP)

Weil in jedem Netzwerk und Knoten ab und zu Fehler auftreten, welche an die Verursacher oder an die Betroffenen weitergemeldet werden müssen. Diese Aufgabe wird vom ICMP Protokoll wahrgenommen. ICMP ist Bestandteil jeder IP Implementierung und hat in seiner Eigenschaft als Transportprotokoll nur die Aufgabe, Fehler- und Diagnoseinformationen für IP zu transportieren. Seine Transportprotokoll-Adresse im IP Protokollkopf ist 1.

Da ICMP unterschiedliche Informationen zu transportieren hat, ist nur der Grundaufbau des ICMP Protokollkopfes immer gleich, Die Bedeutung der einzelnen Felder wechselt jedoch.

0 7	8 15	16 31
Typ	Code	Prüfsumme
Verschiedenes		
IP Kopf und 8 weitere Bytes oder Testdaten		

Die oben abgebildete Struktur, welche allen ICMP Nachrichten zugrundeliegt, enthält folgende Felder:

- Typenfeld: spezifiziert die Art der ICMP Nachricht. Eine Auswahl von Typen ist:

Typenfeld	Funktion
0	Echo replay
3	Destination Unreachable
4	Source Quench
5	Redirect (Routenwechsel)
8	Echo Request
11	Time Exceeded for a Datagramm
12	Parameter Porblem on a Datagram
13	Timestamp request
14	Timestamp Replay

- Code: weiterer Code, welcher Unterfunktionen innerhalb eines Types darstellt.
- Prüfsummenfeld: Internet Prüfsumme der gesamten ICMP Nachricht.
- Verschiedenes: 32 Bit langes Feld, das Informationen für verschiedene Zwecke wie z.B. Sequenznummer, Internet-Adresse usw. aufnehmen kann.
- IP Protokollkopf: Enthält das auslösende IP Datagramm und die ersten 8 Byte der darin transportierten Nachricht. Wurde die ICMP Nachricht durch eine Nachricht von TCP oder UDP ausgelöst, kann mit Hilfe der ersten 8 Bytes des TCP- oder UDP Protokollkopfes das Applikationsprogramms ermittelt und eine Fehlermeldung übergeben werden, z.B. dass die beim Verbindungsaufbauwunsch spezifizierte Server-Portnummer am Zielrechner nicht aktiv ist.

ICMP Nachrichten werden intern sowohl von IP als auch von TCP und UDP angestossen und verarbeitet. Man darf niemals als Antwort auf eine andere ICMP Nachricht oder eine Broadcast erzeugt werden, da sonst eine Endlosschlaufe entstehen könnte.

ICMP Pakettypen

Folgende Pakettypen sind auf allen UNIX Systemen implementiert.

Destination Unreachable: Meldet bei der Zustellung einer Nachricht folgende Störungen: 1. Netzwerk, Host, Protokoll oder Port unerreichbar. 2. Fragmentierung benötigt, aber das DF Bit war gesetzt. 3. Source Route Option war nicht erfolgreich.

Source Quench: Wenn ein Gateway keine Kapazität zur Pufferung einer Nachricht hat, kann es diese ICMP Nachricht an den verursachenden Host senden. Der sendende Host muss dann die Aussenderate von weiteren Nachrichten verringern.

Redirect: Wird ausgesendet, wenn ein Gateway erkennt, dass der Absender eines IP Paketes dieses direkt an den nächsten Gateway senden könnte. Die ICMP Nachricht enthält die Internet Adresse des nächsten direkten Gateways, sie wird in die Routingtabelle des Absenders eingetragen.

Echo Replay und Echo Request: Ein Echo Request wird an einen Rechner gesendet und hat ein Echo Replay zur Folge. Somit lassen sich praktische Funktionen wie die Ueberprüfung der Betriebsbereitschaft oder Laufzeitmessungen implementieren.

Time Exceeded: Diese Nachricht wird an einen Absender gesendet, dessen IP Paket nach Ablauf seiner Lebenszeit wegeworfen werden muss.

Parameter Problem: Der Absender eines IP Paketes wird verständigt, dass das Paket aufgrund von fehlerhaften Angaben im IP Protokollkopf weggeworfen werden musste.

Konfigurationsdatein bei UNIX-Systemen

Die nachfolgende Beschreibung der Konfigurationsdateien eines Unix-Betriebssystems ist nicht vollständig. Die Dateien können von System zu System verschieden sein. Die Beschreibung der Befehl bezieht sich auf das Betriebssystem SUN OS 4.1.3. Eine vollständige Beschreibung der Dateien befindete sich in den man-Pages (UNIX-Befehl man 5 <filename>, wobei <filename> durch den entsprechenden Namen der Datei zu ersetzen ist, z.B.: man 5 passwd). Alle Konfigurationsdateien befinden sich im Directory /etc.

passwd

Die Datei passwd enthält Angaben (Username, Passwort (verschlüsselt), User-ID von 0 bis 32767, Group-ID von 0 bis 32767, richtiger Name, Home-Directory, Login-Shell) über Benutzer des Systems.

Beispiel /etc/passwd:

```
ecgisler:bkua17Gh/KcPY:11012:1000:Iwan.
Gisler:/home/users1/E/ecgisler:/bin/csh
ecrohrer:ayijET85hDb3I:11020:1000:Edgar
Rohrer:/home/users1/E/ecrohrer:/bin/csh
ebspeck:DPNvPYyWRXXMU:11021:1000:Christoph
Speck:/home/users1/E/ebspeck:/bin/csh
wakaech:dvfOd76H1BPfc:30114:1000:Franz
Kaech:/home/users1/Kurse/wakaech:/bin/csh
```

group

goup enthält Informationen (Gruppennamen, Gruppenpasswort, Group-ID, Userliste) über alle Grouppen, die dem System bekannt sind.

/etc/group

```
B:*:10004:zathoma,zcoderma
5A:*:14051:habrande
E:*:10002:zabuergl,zadyntar,zaengel,zagabrie,zagenhar,zahaemme,zahausma,zaheini,
zahuser,zamatter,zaott,zatrutma,zbschmid,zbzimmer
3Ec:*:11033:eabaerts,eabassi,eaimober,eakaesli,eamahler,eamuelle,eapoli,easchuma
,eastaub,easteime,eawidmer,eazanutt,eazettel,ebspeck,ecrohrer,eczgragg
5Ec:*:11053:eadurrer,eaemch,eahauser,eaheini,eaimhof,eakurman,ealuesch,eanietli,
earogger,easchade,easchnyd,eastirni,eastocke,ebburri,ebzimmer,ecgisler,masteinm
```

hosts

Die Datei hosts enthält Informationen (IP-Adresses, Name) von Hosts (Zuordnung Name->IP-Adress).

Beispiel /etc/hosts:

```
# Sun Host Database
#
127.0.0.1       localhost
#
147.88.16.4     ztxs01main mailhost loghost     # le2
147.88.144.4    ztxs01          ztxs01a         # le0
147.88.160.4    ztxs01b                         # le1
#
147.88.16.5     ztxs02main                      # le2
147.88.144.5    ztxs02a                         # le1
147.88.160.5    ztxs02          ztxs02b         # le0
#
147.88.16.13    ztxw01
147.88.16.14    ztxw02
147.88.145.3    ztxw03
147.88.145.4    ztxw04
147.88.145.5    ztxw05
147.88.145.6    ztxw06
147.88.145.7    ztxw07
```

ethers

In der Datei /etc/ethers befinden sich Information über die Ethernetadressen von Hosts (Zuordnung Ethernetadressen->IP-Adressen).

Beispiel /etc/ethers:

```
8:0:20:13:2c:63 czxw01
8:0:20:12:1c:10 czxw02
8:0:20:12:28:14 czxw03
8:0:20:12:1b:b8 ztxw05
8:0:20:12:42:af ztxw06
8:0:20:13:2c:c9 ztxw07
8:0:20:12:1c:f0 ztxw08
8:0:20:12:41:0b ztxw09
8:0:20:12:23:9f ztxw10
8:0:20:13:2c:8e ztxw11
8:0:20:12:1e:ad ztxw12
8:0:20:13:2c:c5 ztxw13
8:0:20:12:40:f3 ztxw14
```

networks

In dieser Datei sind alle, dem System bekannten, Netzwwerke aufgelistet (Zuordnung Netwerkname->IP-Adresse).

Beispiel /etc/networks:

```
# Subnet at ZTL
NET-ZTL                 147.88
NET-ZTL-BACKBONE        147.88.16
NET-ZTL-ADM             147.88.32
NET-ZTL-IZE             147.88.48
NET-ZTL-IZM             147.88.64
NET-ZTL-HLK-A           147.88.80
NET-ZTL-HLK-B           147.88.84
NET-ZTL-NT              147.88.96
NET-CBZS                147.88.112
NET-CBZS-GL             147.88.116
NET-CBZS-BW             147.88.120
NET-ZTL-PRL             147.88.128
NET-ZTL-IZME-A          147.88.144
NET-ZTL-IZME-B          147.88.160
NET-ZTL-IZ              147.88.176
NET-ATIS                147.88.192
NET-ZTL-CIM             147.88.208
NET-7AIR-BSLSTR         147.88.240
NET-7AIR-HITZ           147.88.236
NET-7AIR-GBRSTR         147.88.244
```

rpc

Diese Datei enthält Namen, die anstatt von Nummern in RPC-Programmen verwendet werden können.

Beispiel /etc/rpc

```
portmapper      100000  portmap sunrpc
rstatd          100001  rstat rup perfmeter
rusersd         100002  rusers
nfs             100003  nfsprog
ypserv          100004  ypprog
mountd          100005  mount showmount
ypbind          100007
walld           100008  rwall shutdown
yppasswdd       100009  yppasswd
etherstatd      100010  etherstat
rquotad         100011  rquotaprog quota rquota
sprayd          100012  spray
3270_mapper     100013
rje_mapper      100014
selection_svc   100015  selnsvc
database_svc    100016
rexd            100017  rex
alis            100018
sched           100019
```

services

Diese Datei enthält Information über verschiedene Services, die das System zur Verfügnung stellt.

Beispiel /etc/services

```
tcpmux          1/tcp                           # rfc-1078
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
name            42/udp          nameserver
whois           43/tcp          nicname         # usually to sri-nic
domain          53/udp
domain          53/tcp
hostnames       101/tcp         hostname        # usually to sri-nic
sunrpc          111/udp
sunrpc          111/tcp
x400            103/tcp                         # ISO Mail
x400-snd        104/tcp
```

netgroup

In dieser Datei werden Netzwerkgruppen definiert.

Beispiel /etc/netgroup:

```
c204    (ztxs01,,) (ztxs01main,,) (ztxs01a,,) (ztxs01b,,) \
        (ztxs02,,) (ztxs02main,,) (ztxs02a,,) (ztxs02b,,)
e210    (ztxw05,,) (ztxw06,,) (ztxw07,,) (ztxw08,,) (ztxw09,,) (ztxw10,,) \
        (ztxw11,,) (ztxw12,,) (ztxw13,,) (ztxw14,,) (ztxw15,,) (ztxw16,,) \
        (ztxw17,,) (ztxw18,,) (ztxw19,,) (ztxw20,,)
e209    (ztxw21,,) \
        (ztxw22,,) (ztxw23,,) (ztxw24,,) (ztxw25,,) (ztxw26,,) (ztxw27,,) \
        (ztxw28,,) (ztxw29,,) (ztxw30,,) (ztxw31,,) (ztxw32,,) (ztxw33,,)
dozcad  (ztxw34,,) (ztxw35,,)
b2      (ztxw36,,) (ztxw37,,)
iz      (ztxw38,,)
abtE    (ztxw39,,) (ztxw40,,) (ztxw41,,) (ztxw42,,) (ztxw43,,) (ztxw47,,) (ztxw4
8,,) (ztxw49,,)
cim     (czxw01,,) (czxw02,,) (czxw03,,)
ultrix  (ztxw01,,) (ztxw02,,) (ztxw03,,) (ztxw04,,)
sun     c204 e210 e209 dozcad b2 iz abtE cim
cluster sun ultrix (ztxw99,,) (dsj-d402,,)
vax     (ztle01,,) (ztle02,,) \
        (ztlw01,,) (ztlw02,,) (ztlw03,,) (ztlw04,,) (ztlw05,,) (ztlw06,,) \
        (ztlw08,,) (ztlw09,,) (ztlw10,,) \
        (ztlw12,,) (ztlw13,,) (ztlw14,,) (ztlw15,,) (ztlw16,,) (ztlw17,,) \
        (ztlw18,,) (ztlw19,,) (ztlw20,,) (ztlw21,,) (ztlw22,,) (ztlw23,,)
```

aliases

aliases definiert Mail-Aliase für sendmail.

Beispiel /etc/aliases:

```
Postmaster: root, zahammer, zakappel
# Alias for mailer daemon; returned messages from our MAILER-DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster
# Aliases to handle mail to programs or files, eg news or vacation
# decode: "|/usr/bin/uudecode"
nobody: /dev/null

# Sample aliases:

# Alias for distribution list, members specified here:
staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie
```

netmasks

Mit Hilfe dieser Datei , kann für ein Netzwerk die Netzwerkmaske bestimmt werden.

Beispiel /etc/netmasks:

```
# only non-default subnet masks need to be defined here
#
# Network       netmask
147.88          255.255.252.0
```

resolv.conf

Die Datei enthält Angaben über die Nameserver und den Domain-Name.

Beispiel /etc/resolv.conf:

```
domain          ztl.ch
nameserver      127.0.0.1
nameserver      147.88.144.5
```

inetd.conf

Die Datei inetd.conf enthält eine Liste von Servern, die von inetd benutzt wird, wenn der Prozess eine Anfrage über einen Socket erhält.

Beispiel /etc/inetd.conf:

```
# @(#)inetd.conf 1.24 92/04/14 SMI
# Configuration file for inetd(8).  See inetd.conf(5).
# Internet services syntax:
#  <service_name> <socket_type> <proto> <flags> <user> <server_pathname> <args>
# Ftp and telnet are standard Internet services.
#
ftp     stream  tcp     nowait  root    /usr/etc/in.ftpd        in.ftpd
telnet  stream  tcp     nowait  root    /usr/etc/in.telnetd     in.telnetd
#
# smtp to pp-mta
smtp    stream  tcp     nowait  pp      /usr/local/lib/pp/cmds/chans/smtpsrvr sm
tpsrvr -t 60 smtp
#
# Shell, login, exec, comsat and talk are BSD protocols.
shell   stream  tcp     nowait  root    /usr/etc/in.rshd        in.rshd
login   stream  tcp     nowait  root    /usr/etc/in.rlogind     in.rlogind
exec    stream  tcp     nowait  root    /usr/etc/in.rexecd      in.rexecd
comsat  dgram   udp     wait    root    /usr/etc/in.comsat      in.comsat
talk    dgram   udp     wait    root    /usr/etc/in.talkd       in.talkd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers."  Many sites choose to disable
# some or all of these services to improve security.
#
finger  stream  tcp     nowait  nobody  /usr/etc/in.fingerd     in.fingerd
#
# Time service is used for clock syncronization.
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
#
# Echo, discard, daytime, and chargen are used primarily for testing.
echo    stream  tcp     nowait  root    internal
echo    dgram   udp     wait    root    internal
discard stream  tcp     nowait  root    internal
discard dgram   udp     wait    root    internal
daytime stream  tcp     nowait  root    internal
daytime dgram   udp     wait    root    internal
chargen stream  tcp     nowait  root    internal
chargen dgram   udp     wait    root    internal
```

*hostname.**

Die Datei enthält den Namen des Hosts, der unter dieser Schnittstelle verfügbar ist. In der Datei /etc/hosts kann mit Hilfe der Datei /etc/hosts die entsprechende IP-Adresse bestimmt werden.

Beispiel /etc/hostname.*:

```
ztxs01% ls hostname.*
hostname.le0  hostname.le1  hostname.le2
ztxs01% more hostname.*
::::::::::::::::
file hostname.le0
::::::::::::::::
ztxs01
file hostname.le1
::::::::::::::::
ztxs01b
file hostname.le2
::::::::::::::::
ztxs01main
ztxs01%
```

Beschreibung der wichtigsten UNIX-Befehle

In diesem Kapitel werden die wichtigsten Befehle, die Sie im Internet Versuch anwenden müssen, beschrieben. Dieses Kapitel stellt eine Zusammenfassung der original Dokumentation dar. Für den Versuch steht ein Kopie der Befehlsanwendungen aus TCP/IP Netzwerk Administration von Craig Hunt, O'Reilly Verlag zur Verfügung.

ifconfig

Das Dienstprogramm ifconfig kann für viele verschiedene Zwecke verwendet werden. Bei der Störungsbeseitigung verwenden Sie ifconfig mit dem Argument `interface`, um zu erfahren, wie eine Ethernet-Schnittstelle (oder Port) konfiguriert wurde und ob das System über diese Schnittstelle mit dem Netzwerk kommunizieren kann. Auf NeXT-Computern lautet die erste Ethernet-Schnittstelle immer `en0`. und die Loopback-Schnittstelle `lo0` (auf der Sun-Workstation heißt die erste Ethernet-Schnittstelle le0 und die Lookback-Schnittstelle lo0).
Syntax:

```
ifconfig interface [ip-addr] [netmask mask] [broadcast
broadcast-addr]
```

Optionen:

`interface`	en0 oder lo0 für Ethernet-Schnittstelle 1 und Loopback-Schnittstelle
`ip-addr`	IP-Adresse
`netmask mask`	Setzt die Netzmaske auf `mask` (wird die Netzmaske nicht spezifiziert, wird die Default Netzmaske aufgrund der IP-Adresse gesetzt).
`broadcast broadcast-addr`	Setzt die Broacast-Adresse auf die Adresse `broadcast-addr` (wird die broadcast-Adresse nicht spezifiziert, wird die Default-Broacastadresse aufgrund der IP-Adresse und der Netzmaske gesetzt).

arp

ARP ist die Abkürzung für "Address resolution protocol". Mit Hilfe diese Programms können Sie die ARP-Tabelle ihres Systems anzeigen lassen und verändern.
Syntax:

```
arp hostname
arp -a
arp -d hostname
arp -s hostname ether_addr [temp]
```

Optionen:
Ohne Flags zeigt das Programm den aktuellen ARP-Eintrag für den Host `hostname` an (hostname kann ein Name oder eine IP-Adresse sein).

`-a` Die gesamte ARP-Tabelle wird ausgegeben.
`-d` Der Eintrag für den Host `hostname` wird gelöscht (benötigt Superuser-Rechte).
`-s` Die ARP-Tabelle wird um den Eintrag `hostname ether_addr` erweitert. Wird zusätzlich die Option `temp` angegeben, wird der Eintrag nicht permanent in die Tabelle aufgenommen.

ping

Das Programm ping tauscht zwischen zwei Hosts systemnahe Meldungen (ICMP ECHO_REQUEST) aus, um die Verbindung zu bestätigen. Sie wenden ping auf andere Hosts im Netzwerk an, um zu prüfen, ob sie Meldungen erhalten und Informationen austauschen können. Der Befehl ping ermöglicht einen einfachen und direkten Test.
Syntax:

```
ping [-r] [-v] host [packetsize]
```

Optionen:

`-r` Die Routing-Tabelle wird nicht beachtet. Das Paket wird direkt an den Host gesendet.
`-v` Verbose-Ausgabe, alle ICMP-Pakete werden angezeigt.

Bemerkung:
Um das Netzwerk zu testen wird ping zuerst auf den lokalen Host angewandt, dann auf einen Host im selben Netzwerk und schließlich auf einen Host in einem anderen Netzwerk, das über einen Router oder über eine Gateway verbunden ist.

netstat

Das Dienstprogramm netstat zeigt Informationen über den Netzwerkstatus an. Zahlreiche Argumente ermöglichen es Ihnen, spezifische Komponenten des Netzwerkes zu prüfen - wie etwa die Hosttabelle, Leitwegtabellen (Routing-Tabelle) usw.
Syntax:

```
netstat [-i] [-r] [-n]
```

Optionen:
Wird kein Argument angegeben, wird eine Tabelle aller aktiven Sockets ausgegeben.

`-a` Es wird eine Tabelle aller Sockets angezeigt.

`-i` Netstat zeigt den Status der Schnittstellen an.

`-r` Die Routing-Tabelle wird ausgegeben. Die Ausgabe sieht etwa folgendermaßen aus:

```
Routing tables
Destination      Gateway            Flags     Refs     Use
Interface
localhost        localhost          UH          4     1520  lo0
140.211.192.0    140.211.192.27     U           0        0  en0
```

Die Zieldaten (destination) beziehen sich auf eine Netzwerkadresse oder auf einen Hostnamen. Die Gateway-Daten beziehen sich auf Internet-Adresse oder Hostnamen, die verwendet wurden, um zu diesem Ziel zu gelangen. Im obigen Beispiel werden Pakete, die für das Netzwerk 140.211.192.0 bestimmt sind, an 140.211.192.27 gesendet.

`-n` Die Hostnamen als IP-Adressen ausgegeben.

nslookup

Mit Hilfe des Programms nslookup können Sie interaktiv einen DARPA Internet Domain-Nameserver abfragen.
Syntax:

```
nslookup [host-to-find|-[nameserveraddress | nameservername]]
```

Beschreibung:
Wird kein Nameserver angegeben, werden die Namserver, die in der Datei `/etc/resolv.conf` spezifiziert sind, abgefragt. Bei der interaktiven Abfrage kann der Name des Host nach dem Start des Programms eingegeben werden.
Domain-Namen werden in folgende Gruppen eingeteilt (die Liste ist nicht vollständig):

com	U.S. kommerzielle Domains, z.B.: microsoft.com
edu	U.S. Schulen, z.B.: wisc.edu für University of Wisconsin
gov	U.S. Sites von der U.S. Regierung, z.B.: nih.gov für National Institutes of Health
mil	U.S. Militär, z.B.: ddn.mil für Defense Data Network
net	U.S. Sites, die für die Administration eines Netzwerkes dienen, z.B.: conncert.net
org	U.S. Organistions, normalerweise von Privaten, z.B.: isoc.org für Internet Society
ch	Sites in der Schweiz, z.B.: ztl.ch
de	Sites in Deutschland, z.B.: archie.th-darmstadt.de für den Archie-Server an der technischen Hochschule in Darmstadt.

traceroute

Das Programm traceroute kann den Weg eines Pakets im Netzwerk verfolgen und die einzelnen Stationen (Router) ausgeben.
Syntax:

```
traceroute [-m max_ttl] [-n] [-r] [-v] [-w time] host
```

Optionen:
Werden keine weiteren Optionen außer der Host angegeben, zeigt traceroute alle Router an, bei denen das Paket geroutet wurde

`-m`	Setzt das 'time to life'-Feld (Anzahl hops, Router) im IP-Protokollkopf. Default 30
`-n`	Es werden anstelle von Namen, IP-Adressen ausgegeben.
`-r`	Die Routing-Tabelle wird ignoriert und das Paket direkt auf das Netzwerk gegeben.
`-v`	Verbose-Ausgabe. Es werden alle ICMP-Pakete auf den Bildschirm ausgegeben.
`-w time`	Setzt die Wartezeit auf eine Antwort auf sec (default 3s). Wird während dieser Zeit keine Antwort erhalten wird in der Liste ein "*" ausgegeben.

Bemerkung:
Traceroute verwendet das Feld 'time to life' des IP-Protokols. Diese Feld wird bei jedem Router dekrementiert. Ist der Wert 0, wird das Paket weggeworfen und ein ICMP TIME_EXCEEDED Paket an den Absender gesendet. Diese Paket wird von traceroute ausgewertet.

telnet

Telnet wird verwendet um Terminalverbindungen zu einem Host herzustellen.
Syntax:

```
telnet [host [port]]
```

Optionen

`host`	Spezifiziert den Zielhost.
`port`	Spezifiziert die Portadressse auf dem Zielhost.

Bemerkung:
Wird beim Start von Telnet kein Host angegeben, kommt man in dem Kommandomodus von Telnet. Hier stehen eine Vielzahl von Kommandos zur Verfügung (siehe `help`). Die einzelnen Kommandos können von System zu System verschieden sein (Syntax und Implementierung). Mit einem Fluchtzeichen (normalerweise `^]`, CTRL +]) kann man nachdem eine Verbindung hergestellt wurde, wieder in den Kommandomodus wechseln.

ftp

Ftp ist ein Front-End für das ARPANET Standard File-Transfer-Protokoll. Ftp ermöglicht das Kopieren von Dateien zwischen zwei Hosts.

Syntax:

```
ftp [-v] [-i] [host]
```

Optionen:

`-v` Verbose-Ausgabe, ftp gibt alle Meldungen des remote-ftp-servers auf den Bildschirm aus.

`-i` Während eines Multi-File-Transfer wird die Rückfrage unterdrückt.

Bemerkung:

Wird beim Start von ftp ein Host angegeben, wird eine Verbindung zu diesem Host aufgebaut. Ftp kennt, nicht wie telnet, einen On-Line und einen Kommandomodus, sondern nur einen Kommandomodus. Für den Benutzer sind folgende Befehle von Interesse:

```
?       Hilfe
help    Wie ?
open                                                                    host
Stellt eine Verbindung zum Host Host her.
close   Schließt eine Verbindung.
binary  Stellt auf binary-Übertragungsmodus um.
ascii   Stellt auf ascii-Übertragungsmodus um.
hash    Während der Übertragung wird für jeden Block ein #-Zeichen auf den Bildschirm
geschrieben (ein Block entspricht 1024 Bytes).
get                             remotefilename                      [localfilename]
Kopiert die Datei remotefilename vom remote-Host auf den local-Host.
put                             localfilename                     [remotefilename]
Kopiert die Datei localfilename vom local-Host auf den remote-Host.
mget                                                                    filemaske
Kopiert Dateien, die der filemask entsprechen, vom remote-Host auf den local-Host.
mput                                                                    filemaske
Kopiert Dateien, die der filemask entsprechen, vom local-Host auf den remote-Host.
pwd     Wie unter UNIX.
dir     Entspricht 'ls -l' in UNIX.
cd      Wie unter UNIX.
ls      Wie unter UNIX.
user    Setzt Username und Passwort.
status  Zeigt den Status der Verbindung, des Programms an.
cr      Schaltet zwischen der Umsetzung von CR auf CR oder CR/LF im ASCII-Transfermodus um.
quit    Wie bye.
bye     Beendet das Programm.
```

Serial Line IP (SLIP)

SLIP erlaubt die Verbindung von zwei Rechnern über eine serielle Leitung, z.B. V.24.

Jedes SLIP Paket beginnt mit einem Byte mit dem Wert: #EB(hex), welches im Protokoll als ESC bezeichnet wird. Am Ende der Daten steht ein #C0(hex) (END). Falls diese Werte in den Dateien vorkommen, so werden sie als 2 Byte Sequenz ESC #EC(hex) bzw. als ESC #ED(hex) gesendet und vom Empfänger wieder in END und ESC zurückverwandelt. Eine maximale Paketlänge ist nicht vorgeschrieben, pro Konvention liegt die zu_erwartende Obergrenze bei 1006 Bytes.

SLIP enthält keine Adressen oder sonstige Protokollfelder. Es dient nur zur Einteilung der Daten in Pakete für den Transport über eine Punkt - zu - Punkt - Verbindung. Der SLIP Treiber wird in den meisten UNIX Implementierungen mitgeliefert. Mit Hilfe einer seriellen Leitung lässt sich jeder UNIX Rechner einfach in ein Gateway verwandeln, z.B. für die Kopplung zweier Ethernet Segmente in verschiedenen Gebäuden via eine Telefonleitung. Allerdings muss gesagt werden, dass bei einer maximalen Uebertragungsgeschwindigkeit von 19200 Baud ein grosser Leitungsverlust gegenüber den normalen 10MBit/s Ethernet besteht. Die Belastung des UNIX Rechners mit der Bedienung der seriellen Leitung noch nicht mitgerechnet.

Heute wird anstelle von Slip immer häufiger **PPP (Point to Point Protokoll)** eingesetzt. Auch für die Verbindung von Multiprotokollroutern wird z.T. PPP eingesetzt.

Ethernet 802.3

Ethernet ist eine LAN-Technologie (Local Area Network), welche am Anfang der 70er Jahre von der Firma XEROX entwickelt wurde. 1978 wurde die Version 2 dieses Verfahrens von den Firmen XEROX, DEC und Intel standartisiert. Sie ist die heute am weitesten verbreitete Technologie für lokale Netzwerke. Anfang der 80er Jahre wurde eine nahezu identische Version zum Internationalen Standart mit der Bezeichnung IEEE 802.3 erklärt (EthernetII).

Charakteristische Eigenschaften:

Als physikalisches Medium des Ethernets dient ein Koaxialkabel mit einer Uebertragungs-geschwindigkeit von 10MBits pro Sekunde. Die Verkabelung erfolgt bus- oder sternförmig.

Zunehmend beliebter (aus Kostengründen) wird aber auch die Verkabelung mit verdrillten Zweidrahtleitungen, welche abgeschirmt (STP: shielded twisted pair) oder aber nicht abgeschirmt (UTP: unshielded twisted pair) sein können.

Die folgende Tabelle listet die gebräuchlichsten Kabeltypen des Ethernets auf:

Kabeltyp	Thick Wire (Yellow Cable; 50 Ohm) Dickes Koaxialkabel	Thin Wire (RG58; 50 Ohm) Cheapernet dünnes Koaxialkabel	Twisted Pair
Bezeichnung:	10Base5	10Base2	10BaseT
Segmentlänge	500m	185m	100m
Abgriffe pro Segment	100	30	Sternverkabelung mit Repeater (max 4 Repeater in Serie!)

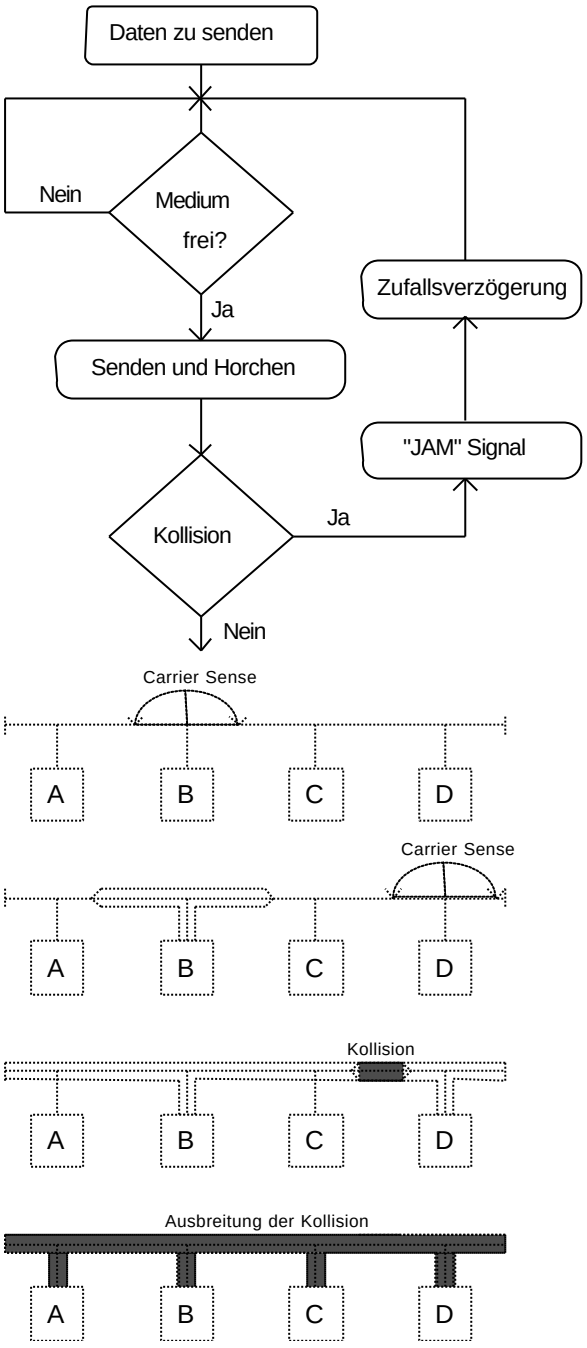
Andere Kabeltypen wie das CATV 75Ohm Kabel werden nur selten und in Spezialfällen eingesetz. Das Glasfaserkabel wird heute vermehrt als sogenannter "Backbone" (Rückgrat) eingesetzt, um Teilsegmente eines LAN's miteinander zu verbinden.

Weitere Einschränkungen bilden die maximal 1024 Nodes pro Netz und die 64 Netze pro Netzwerk sowie die maximale Länge von 2.8 km (wegen Signallaufzeit).

Das Protokoll:

Das von Ethernet verwendete Protokoll wird als CSMA/CD (Carrier Sense Multiole Access with Collision Detection) bezeichnet, was etwa mit: pegelgesteuerter Mehrfach-Zugriff mit Kollisionsentdeckung übersetzt werden kann. Einfach ausgedrückt: bevor eine Station ein Paket sendet, überprüft sie, ob bereits eine andere Station sendet (Carrier Sense). Falls ja, wartet sie solange bis das Kabel frei ist, ansonsten sendet sie sofort. Sollten zwei oder mehrere Stationen durch Zufall gleichzeitig zu senden beginnen (Collision), erkennen sie diesen Umstand, weil sie die ausgesendeten Daten ständig mit mit den Daten auf dem Kabel vergleichen. Sind sie verfälscht, wird der Sendevorgang für eine von einem Zufallsgenerator ermittelten Zeit unterbrochen. Die Wartezeit wird bei Wiederholungen von Kollisionen exponentionell erhöht.

Bild: CSMA/CD



Bemerkung: Das "JAM" Signal bedeutet, dass bei einer Kollision eine zufällige Zeichenfolge weitergesendet wird (jam-sequence).

Die Adressierung wir mit einer 48 Bit langen realisiert, welche in jedem Ethernet-Kontroller fest in die Hardware eingetragen ist. Diese Adresse ist weltweit einmalig, da die IEEE den Herstellern von Ethernetkontroller diese Nummern zuteilt.

Der Aufbau des Ethernet Pakete ist im folgenden Bild dargestellt. Zur Veranschaulichung ist in dieses Paket die Lage der IP und TCP Protokollköpfe eingezeichnet. So lassen sich die einzelnen Protokollschichten leichter erkennen.

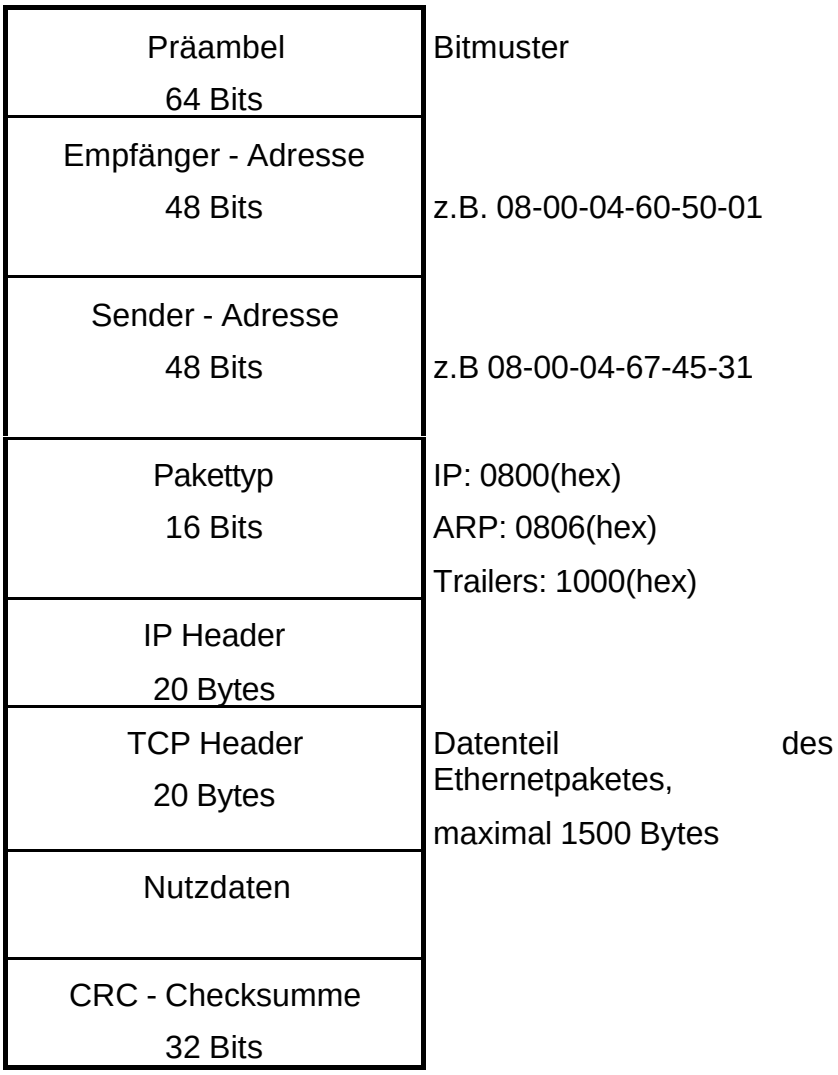
Am Anfang des Paketes steht ein spezielles Bitmuster, die Präambel. Sie dient zum Synchronisieren der empfangenden Stationen.

Danach folgen die Empfänger - und die Sender - Adresse mit je 48 Bits.

Im Feld: Pakettyp wird beim Ethernet-Standart die Adresse des nächsthöheren Protokolls eingetragen, wie hier z.B. IP. Damit lassen sich verschiedene Protokolltypen oberhalb von Ethernet gleichzeitig unterstützen.

Nach dem bis zu 1500 Bytes grossen Datenteil folgt eine 32 Bit grosse CRC - Prüfsumme.

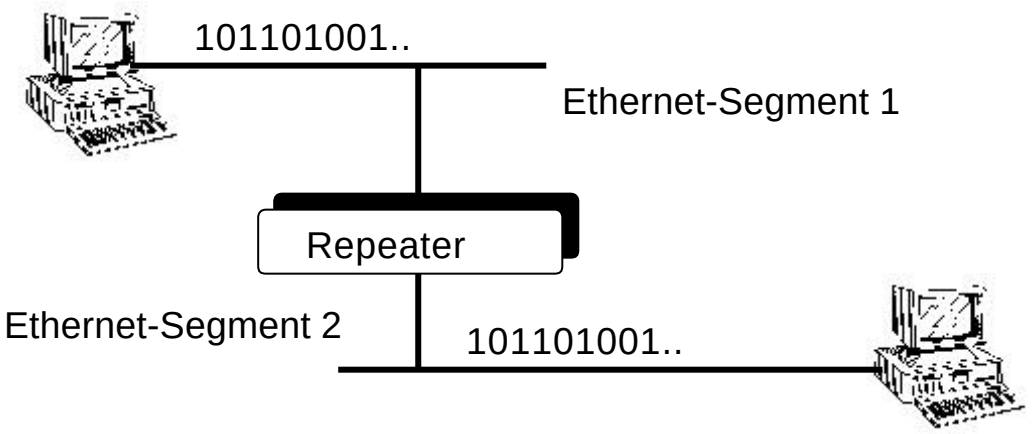
Aus übertragungstechnischen Gründen ist ein EthernetII Paket mindestens 64 Bytes lang sein. Falls weniger Daten zu senden sind, wird das Paket vom Ethernettreiber des Betriebssystems künstlich auf diese Länge gebracht.



Im Betrieb liest jede Station die vorbeilaufenden Pakete immer mit und vergleicht die Empfängeradresse mit der eigenen. Falls sie übereinstimmen, ist die Station der Empfänger und sie übernimmt das Paket vollständig. Aufgrund der Eigenschaft, dass alle Stationen im Netz jedes Paket lesen, kann man gleichzeitig eine Nachricht an all (Broadcast) oder an Gruppen von Stationen (Multicast) senden. Das Broadcasting wird z.B. vom ARP / RARP Protokoll verwendet.

Repeater

Der Repeater verstärkt nur die physikalischen Signale in einem Netzwerk. Er liest die empfangenden Daten und speichert sie. Dann rekonstruiert er das Signal und sendet die Daten erneut aus. Mit ihm kann aber auch ein Netzwerk sternförmig aufgebaut werden (UTP/STP Mehrfachrepeater) oder ein Uebergang von Koaxialkabel auf STP/UTP realisiert werden.

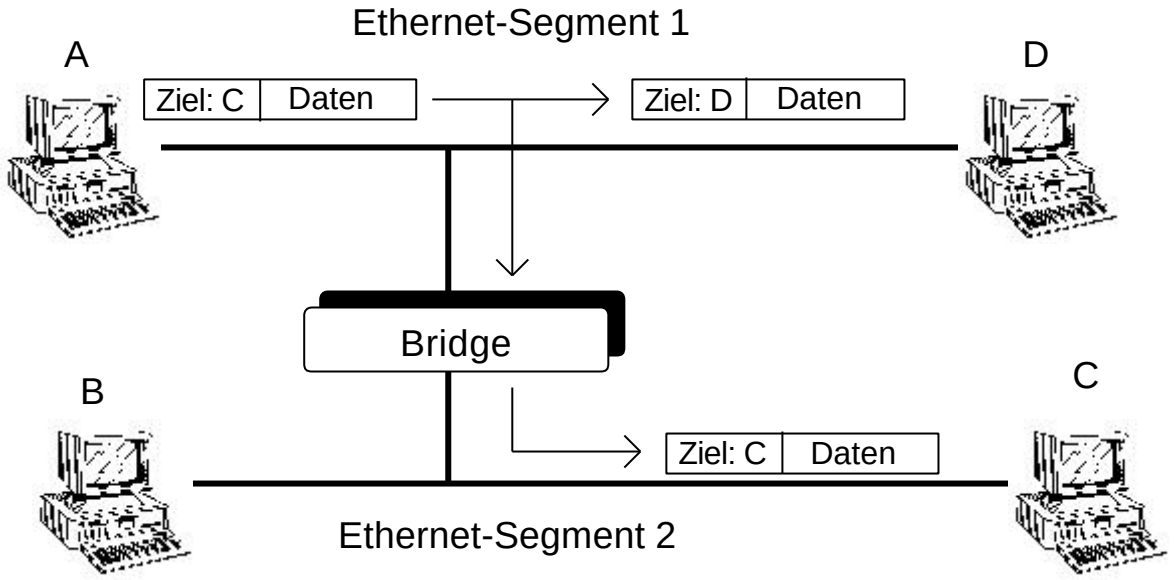


Bridge

Die Bridge regelt den Datenverkehr im Netzwerk. Sie sorgt dafür, dass z.B. der lokale Datenverkehr nicht in andere Netzwerksegmente "exportiert" wird. Die Bridge horcht an den beiden Anschlüssen die angeschlossenen Netzwerke ab und merkt sich die verwendeten Adressen in den Ethernet-Paketen in internen Tabellen. Sobald sie weiss, auf welcher Seite sich die einzelnen Hosts befinden, lässt sie den Verkehr nur noch bei Bedarf durch. Die Tabellen werden dynamisch auf- (und ab-) gebaut. Der absendende Host merkt nichts davon.

Die Bridges arbeiten fast ausschliesslich auf dem MAC-Layer (MAC = Medium Access Control), wo sie die Pakete auf der Sicherungsschicht (Link Layer) wie z.B. Ethernet verarbeiten.

Eine weitere Aufgabe kann der Uebergang zu einer andere Uebertragungsrate sein.



Router

Sie halten wie die Bridges den lokalen Verkehr zurück und lassen nur Pakete durch, die zu einem anderen Netzwerksegment gelangen müssen. Sie arbeiten auf der Netzwerkschicht (Network Layer), also z.B. mit dem IP Protokoll. Sie sind in der Lage, Routing Entscheide anhand von Informationen im Protokoll zu treffen. Beim IP-Protokoll sind diese Informationen im Protokollkopf zu finden. Als Router kann ein gewöhnlicher UNIX-Rechner mit mehreren Netzwerkanschlüssen eingesetzt werden.

Gateway

Das Gateway kann wie die Routers und die Bridges eingesetzt werden. Zusätzlich aber kann noch eine Protokollumwandlung, also z.B. von DecNet nach TCP/IP vorgenommen werden. Sie sind, z.T. erheblich, teurer als die Routers und Bridges, weil in der Regel dafür ein spezieller Rechner eingesetzt wird.
In der TCP/IP-Literatur wird der Begriff Gateway im allgemeinen für einen Router im Sinne der oben genannten Definitionen verwendet.

Routing im Internet

Aus TCP/IP Netzwerk Administration von Craig Hunt, O'Reilly Verlag

Routing-Protokolle

Routing-Protokolle suchen zwischen zwei Stationen die beste Route. Die Bewertung der besten Route wird mit Metriken vorgenommen. Als Metriken können z.B. Anzahl zu durchquerende Knotenrechner, die zurückgelgte Distanz, der optimale ausgeleich der Last etc. verwendet werden.

Wir unterscheiden **Interne Routing-Protokolle**, welche innerhalb eines monolythischen abgegrenzten Systemes routen und **Externe Routing-Protkolle**, welche die monolythisch abgegrenzten Systeme miteinander verbindet. Diese monolytisch gegen aussen abgegrenzten Systeme werden als **Auonomous System (Autonome Systeme, AS)** bezeichnet.

Wir unterscheiden zwei gundsätzlich verschiedene Routing-Algorithmen, die **distance-vector Algorithmen** und die **link-state Algorithmen. Link-State-Algorithmen** berücksichtigen bei der Wegwahl den aktuellen Zustand des Links, das heisst Auslastung, Qualität etc.

Beispiele von Internen Routing-Protokollen

Routing Information Protokoll (RIP). RIP wird vor allem in kleineren und mittleren Netzen verwendet. RIP wählt nach der Metrik hop-count die küzeste Route aus. Die maximale Anzahl Hops ist auf 15 definiert.
Funktionsweise: Der **routed** (=Prozess der RIP abarbeitet) sendet alle 30 sec seine volle Routing-Tabelle an alle direkten Nachbarn. Jeder RIP Rechner ergänzt seine Routing-Tabelle mit noch unbekannten Routen, sofern diese weniger als 15 Hops entfernt sind. Ausserdem werden alte Routen durch kürzere neue ersetzt, sofern es eine solche gibt. Einträge die über 180 sec keinen Update haben, werden aus der Tabelle entfernt, da diese als nicht mehr aktuell angesehen werden. Routed befindet sich in usr/etc/in.routed (SUN) oder etc/routed. Default Routen oder erreichbare Gateways, welche keine Tabelle senden können in der Datei etc/gateways abgespeichert werden. Beim Start von routed werden die Routen aus etc/gateways übernommen.

Hello Protokoll. Hello verwendet als Metrik die Transportzeit des Datagrammes. Bei der Berechnung der Transportzeit wird angenommen, dass alle Systeme eine exakte Zeitbasis haben. Die Differenz zwischen Absendezeit und Empfangszeit eines Hello Datagrammes ist die Transportzeit. Hello wird in NFSNET im 56 und 1554 kBit/sec Bereich verwendet.

Shortest Path First (SPF). SPF wird im Bereich Intemediate Systems (IS-IS) eingesetzt und wurde von OSI definiert. SPF ist ein Link-State-Protokoll und ist deshalb auch für sehr grosse Netze im Backbonebereich geeignet.

Open Shortest Path First (OSPF). OSPF ist ein Link-State-Protokoll, welches multipathfähig ist, das heisst OSPF kann die Auslastung von mehreren

parallelen Routen ausgleichen. OSPF kann für IP nicht verwendet werden, da IP immer die erste (=beste und einizige) Route nimmt.

Beispiele von Externen Routing-Protokollen

Externe Routing-Protkolle vermitteln die Erreichbarkeitsinformation (Reachability Infromation) zwischen Autonomen Systemen. Typischerweise ist unser Anschluss ans Switch die vermittlung der weltweiten Erreichbarkeit.

Exterior Gateway Protokoll (EGP). EGP verteilt die Information über das ihm zugeordnete Netz des Autonomen Systems an die mit ihm verbundenen Autonomen Systeme. Z.B. am ZTL die Adresse ztl.ch.
Entstehung: Als EGP entwicklet wurde war das Internet (wie heute das Milnet immernoch!) hierarchisch gegliedert. Im zentralen Bereich waren Core-Gateways (Backbone Ebene), welche die Autonomen Systeme zentral verwalteten. Ein Autonomes System schickt sein Netz an den Core-Gateway. Als Antwort erhält dieses (AS) vom Core-Gateway tausende Adressen von erreichbaren AS. Heute wird von diesem hierarchischen System wegegangen, da der Verwaltungsaufwand zu gross war. Neu fasst man die AS als verteilte Systeme mit intelligenten Protokollen auf.

Boarder Gateway Protokoll (BGP). BGP wird heute im T3-Backbone (45 mbps) des NFSNET und als Zugriff auf regionale Zentren eingesetzt. BGP kann neben der Routing Information auch Path-Attribute für weitere Informationen mitsenden. Deshalb wird es möglich mehr Kriterien und eine andere als die erst-beste Route zu verwenden. Damit können dann die Autonomen Systeme gleichberechtigt und ohne die Core-Gateways kommunizieren.

Bemerkung: Externe Protokolle werden selten eingesetzt. Normalerweise sind Autonome Systeme Teile eines Autonomen Systems eines Dienstanbieters.

Praktisches einsetzen von Protokollen

Im Normalfall wird in lokalen netzen RIP eingesetzt. Falls es notwendig ist ein Exterior-Gateway-Protokoll einzusetzen wird die Wahl das Partnersystem, welches bereits ein Exterior-Gateway-Protokoll hat, vorgeben ob EGP oder BGP zu wählen sei. OSPF hat in der UNIX-Welt keine grosse Bedeutung.