Band

3

FACHHOCHSCHULE HAGENBERG

Skriptenserie zu Betriebssysteme und Netzwerktechnologie

Vernetzung

Zweite Auflage

FACHHOCHSCHULE HAGENBERG

Unterlagen zur Lehrveranstaltng

Stefan Probst

Jormannsdorf 99 • A-7431 Bad-Tatzmannsdorf
Telefon +43 676 / 3045029
E-Mail: stefan.probst@ssw-consulting.net

Einleitung

Das vorliegende Dokument enthält die Begleitunterlagen zu der Lehrveranstaltung „Netzwerktechnologie“. Das Werk einschließlich aller Teile ist urheberrechtlich geschützt. Die komplette oder nur teilweise Verbreitung und Verwendung dieses Dokumentes bedarf der ausdrücklichen Zustimmung des Urhebers.

Danksagung

Besonderer Dank gilt Herrn Klaus Wolfmaier für die Mitarbeit und Mitgestaltung dieses Dokuments.

Verwendung des Dokuments

SYMBOL-LEGENDE	
ⓘ	Informationen
!	Wichtige Hinweise
✎	Beispiele
?	Verständnisfragen

In diesem Skriptum werden Sie immer wieder auf verschiedene Symbole und Schreibweisen stoßen.

Spezielle Ausdrücke und Begriffe werden durch einfache Anführungszeichen (‚‘) gekennzeichnet. Kommandos werden innerhalb doppelter Anführungszeichen („“) erwähnt. Variable Teile innerhalb eines Kommandos sind *kursiv* gedruckt, genauso wie betonte Wörter. Am Seitenrand werden Sie immer wieder Symbole finden, die auf spezielle Informationen hindeuten.

Inhaltsverzeichnis

Notizen

Kapitel 1

Vernetzung

Dieses Kapitel befasst sich mit den Grundbegriffen der Vernetzung von Computersystemen. Dabei werden verschiedene Netzwerkkonzepte vorgestellt – unter anderem auch das Ethernet-Protokoll, aber auch das ISO-OSI Referenzmodell.

Die Notwendigkeit einer Vernetzung von Rechnern nahm während der gesamten Geschichte der Computer immer mehr zu. Mit dem Fortschritt in der Computertechnik änderten sich auch die Anforderungen an die Kommunikationsfähigkeiten. Gerade die rasante Entwicklung des größten Computernetzwerks überhaupt – dem Internet – verdeutlicht dies. In diesem Kapitel werden Grundbegriffe der Vernetzung und verschiedene Netzwerkkonzepte vorgestellt.

Lehrinhalte und -ziele

Am Ende dieses Kapitels sollten Sie den Zweck von Netzwerken verstanden haben und Netzwerke klassifizieren können. Darüber hinaus sollten Sie das ISO-OSI-Referenzmodell verstanden haben und den Zweck der einzelnen Schichten kennen. Sie sollten aber auch die verschiedenen, eingesetzten Topologien der Vernetzung und die damit verbundenen Vor- und Nachteile kennen.

Ziele von Netzwerken

Warum Vernetzung

- Gemeinsame Nutzung vorhandener Ressourcen
- Verteilen von Arbeitslasten im Netz
- Erreichen einer höheren Zuverlässigkeit (vernetzte Ersatzkomponenten)
- Austausch von Informationen - Informationszugang
- Kommunikationsmöglichkeiten

Der wichtigste Aspekt und Grund für die Vernetzung ist die gemeinsame Nutzung vorhandener Ressourcen. Dabei können im Netz vorhandene Programme, Daten und Geräte von den einzelnen Benutzern gemeinsam genutzt werden, und das unabhängig vom Standort des Benutzers.

Ein weiterer Grund für die Vernetzung von Rechnern ist die Aufteilung der Arbeitslast im Netz. Da sich innerhalb eines Netzes verschieden ausgelastete Rechenkapazität befindet, liegt es nahe, komplexe Berechnungen parallel auf verschiedenen Rechnern durchzuführen. Auch das gemeinsame, gleichzeitige Arbeiten an einem Problem wird durch gemeinsame Ressourcennutzung unterstützt.

Vernetzte Komponenten können außerdem zum Erreichen einer hohen Zuverlässigkeit beitragen. Dazu sind Ersatzkomponenten vorhanden, die bei einem Ausfall einzelner Geräte einspringen und deren Aufgaben übernehmen. Das beste Beispiel hierfür sind Reservedrucker, die einfach bei Ausfall oder Überlastung des Standarddruckers ausgewählt werden können.

Im Zeitalter des Internets haben Rechnernetze jedoch völlig neue Dimensionen erreicht und widmen sich anderen Zielen. So steht im Internet in erster Linie der Informationsaustausch und der Informationszugriff im Vordergrund. Solche globalen Netzwerke eröffnen aber auch ganz neue Wege der weltweiten Kommunikation.

Klassifizierung von Netzen

Klassifizierung von Netzen

- Local Area Network (LAN)
  - Räumlich enges Gebiet (wenige 100m)
  - Hohe Datenraten (100 mBit/s) bei niedriger Fehlerrate
- Metropolitan Area Network (MAN)
  - Größere Region (bis 100 km)
  - Hohe Bandbreite durch Bündelung
- Wide Area Network (WAN)
  - Netze innerhalb von Ländern/Kontinenten
  - Datenrate bis 100 kBit/s
- Global Area Network (GAN)
  - Weltweit (auch Satelliten)
  - Niedrige Datenrate, hohe Fehlerrate

Netzwerke werden prinzipiell nach deren Ausdehnung klassifiziert. In der Regel werden an die unterschiedlichen Klassen auch verschiedene Ansprüche gestellt.

Local Area Network (LAN)

LANs beschränken sich auf ein räumlich enges Gebiet, wie z.B. Bürogebäude. Dabei dehnt sich das Netzwerk auf das unmittelbare Umfeld des Anwenders aus und beschränkt sich in der Regel auf nur wenige 100 Meter.

LANs zeichnen sich durch hohe Datenraten bei extrem niedriger Fehlerrate aus. Dies ist möglich, da durch die beschränkte Ausdehnung nur wenig Störfaktoren auf das Netzwerk wirken können.

Die in LANs eingesetzten Netzwerkkabel erlauben eine Übertragungsgeschwindigkeit im Bereich von 1 MBit/sec bis hin zu 1000 MBit/sec. Dabei sind aber nur kurze Übertragungsstrecken realisierbar und es existiert meist nur ein einziger Weg zwischen den Endrechnern.

LANs zeichnen sich auch dadurch aus, dass die gemeinsame Nutzung von Ressourcen im Vordergrund steht und nur selten das Beschaffen von neuartigen, vielleicht wissenschaftlichen Informationen.

Metropolitan Area Network (MAN)

Diese Art von Netzwerken deckt bereits eine größere Region ab. Ausdehnungen bis zu 100 km fallen unter diese Kategorie von Netzwerken. In MANs sind auch hohe Bandbreiten bei den Übertragungswegen gefordert, jedoch ist die tatsächliche

Übertragungsgeschwindigkeit für den einzelnen Anwender viel geringer da viele Aufträge in ein gemeinsames Übertragungsmedium gebündelt werden.

Wide Area Network (WAN)

WANs sind kontinentale Weitverkehrsnetze. WANs dehnen sich über einzelne Länder und Kontinente aus. Dabei existieren für die Übertragung aber mehrere verschiedene Wege, die auch den Betrieb des Netzwerkes garantieren wenn ein Knoten ausfällt. Durch die große Spannweite gibt es jedoch eine hohe Fehlerrate und eine niedrige Übertragungsgeschwindigkeit (ca. 100 kBit/sec).

Global Area Network (GAN)

Diese Art von Netzwerk dehnt sich über den gesamten Globus. Zur Verbindung der einzelnen Kontinenten werden nicht selten Satelliten zur Übertragung verwendet. Auch hier stehen verschiedene Wege und Routen zur Übertragung bereit. Fallen einzelne Knoten aus, können Alternativrouten für die Übertragung gewählt werden. Durch die enorme Spannweite und die verschiedenen Übertragungswege gibt es jedoch eine sehr große Fehlerrate und eine sehr niedrige Übertragungsgeschwindigkeit.

ⓘ Das beste Beispiel für ein GAN ist das Internet. Hier werden Rechner auf der ganzen Welt zu einem großen Netzwerk verbunden. Auch hier kann man die niedrige Übertragungsgeschwindigkeit beobachten, aber auch die Tatsache dass das Internet trotz eventueller Ausfälle einiger Knotenpunkte immer noch funktioniert.

ISO-OSI Referenzmodell

Aufbau von Netzen

- Problem
  - Technologie herstellerabhängig → muss bei allen Clients vorhanden sein
  - Änderung von Hardware Auswirkung auf Software
- Lösung
  - Hierarchie von Schichten
  - Jede Schicht hat definierte Funktion und normierte Schnittstelle
  - Jede Schicht nutzt Funktionalität aller darunter liegenden Schichten
  - Schichte auf gleicher Ebene kommunizieren über ein Protokoll

Anfängliche Probleme in der Vernetzung

In den Anfängen der Vernetzung zielten verschiedene Hersteller darauf ab, dass die Vernetzung lediglich mit Hard- und Software ihrer Firma möglich war. Das Ergebnis waren proprietäre, geschlossene Systeme die nicht mit Produkten anderer Hersteller kommunizieren konnten. Wollten kleine Firmen Netzwerkprodukte herstellen, so mussten sie ihre Produkte an die Marktführer anpassen.

Außerdem war die Technologie herstellerabhängig was der Weiterentwicklung von Netzwerkprodukten nicht dienlich war. Zudem musste dieselbe Technologie bei allen Clients vorhanden sein. Es war nicht möglich Clients unterschiedlicher Technologiestufen (selbst vom gleichen Hersteller) oder unterschiedlicher Hersteller miteinander zu koppeln. Da Software stark an die Hardware gebunden war, hatte die Änderung von Hardware oder das Erreichen einer neuen Technologiestufe auch gravierende Auswirkungen auf die Software.

Lösung

Um die oben erwähnten Probleme bewältigen zu können und um eine hersteller-unabhängige Kommunikation zu ermöglichen, wurde das Problem durch eine Schichtenbildung bewältigt. Diese Schichtenbildung weist folgende Eigenschaften auf:

- Es existiert eine Hierarchie von Schichten.
- Jedes an der Kommunikation beteiligte Endgerät realisiert alle Schichten.

- Abgesehen von der untersten Schicht, benutzt jede Schicht die von der darrunterliegenden Schicht angebotene Funktionalität zur Aufgabenbewältigung.

- Da jedes Endgerät die gleichen Schichten benutzt, kommuniziert jede Schicht mit ihrer Partnerschicht. Dadurch kann die Funktionalität der Schicht von den anderen Schichten abstrahiert werden. Lediglich die unterste Schicht besitzt eine physikalische Verbindung zum Partner. Alle anderen Schichten kommunizieren über virtuelle Verbindungen. Regeln, die dabei einzuhalten sind, werden *Protokoll* genannt.

- Jede Schicht hat ihr eigenes Protokoll. Die Gesamtheit der Schichten ergibt die Protokollmenge des Systems (*Protokollstack*).

ISO-OSI Referenzmodell I

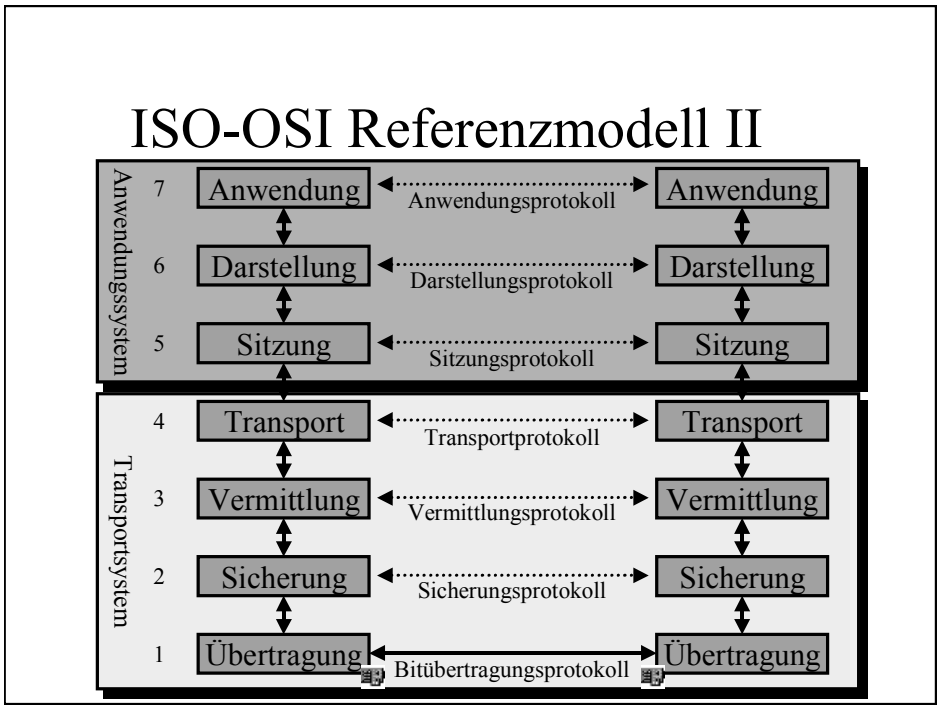
- Von International Standardization Organization (ISO) in den 70ern entwickelt (Open System Interconnection (OSI) Komitee gegründet)
- Modell besteht aus 7 Schichten
  - Transportsystem: Schicht 1 – 4
  - Anwendungssystem: Schicht 5 – 7
- Definiert ist nicht technische Realisierung sondern das Verhalten der einzelnen Schichten (Schnittstellen)

Das ISO-OSI Referenzmodell

Die oben genannte Schichtenbildung war das Ergebnis des 1977 gegründeten OSI-Komitees (OSI = Open System Interconnection) der International Standardization Organization (ISO). Ziel war es, eine Architektur für Kommunikationssysteme zu entwickeln, die unabhängig vom jeweiligen Hersteller war. Das Ergebnis dieser Normungsbemühungen war das sogenannte OSI-Architekturmodell innerhalb dessen detaillierte Normen existieren. Dieses Architekturmodell ist heute als ISO-OSI Referenzmodell bekannt.

Das Modell selbst besteht aus sieben einzelnen Schichten mit fest definierter Funktionalität. Die ersten vier Schichten befassen sich dabei mit dem Transport von Informationen über das Netz und werden deshalb auch unter den Begriff Transportsystem zusammengefasst. Die restlichen Schichten (also 5 – 7) befassen sich mit der Anwendung und werden deshalb auch als Anwendungssystem bezeichnet.

Im ISO-OSI Referenzmodell ist nicht die Art der technischen Realisierung festgelegt, sondern lediglich das Verhalten der einzelnen Schichten nach aßen definiert. Es wird beschrieben, welche Dienste jede Schicht implementieren muss, da diese ja von der nächsthöheren Schicht erwartet werden. Das Referenzmodell schreibt auch nicht die Art und Weise der Implementierung vor. Lediglich die Schnittstellen zwischen den einzelnen Schichten sind fest definiert. Dadurch kann ein Hersteller eine einzelne Schicht auswechseln (z.B. Netzwerkkartentreiber) ohne die restlichen Schichten ändern zu müssen oder zu beeinflussen.



Das hier gezeigte ISO-OSI Referenzmodell wird in dieser Form kaum realisiert. In vielen Implementierungen (z.B. TCP/IP) werden einzelne Schichten zu einer einzigen Schicht zusammengefasst. Trotzdem eignet es sich wegen seiner klassischen Systematik dazu, die Aufgaben der Kommunikationsebenen darzustellen. In praktisch jedem vorhandenen Protokoll-Stack findet sich der Grundgedanke dieses Modells wieder. Im folgenden werden nun die einzelnen Schichten und deren Aufgaben näher behandelt.

Das Transportsystem

Transportsystem I

- Physical Layer (1)
  - Bitübertragung (Hardware)
  - Low- und High-Bit, Kabel, Pins, etc.
- Data Link Layer (2)
  - Sicherungsschicht → zuverlässiger Austausch von Binärpaketen
  - Definiert Telegrammstruktur, Zugriffsverfahren, Synchronisation, Adressierung der Teilnehmer
  - Unterteilung in Media Access Control (MAC) und Logical Link Control (LLC)

Schicht 1: Bitübertragung (Physical Layer)

In dieser Schicht werden ungesicherte Verbindungen für die Übertragung von Bits zwischen verschiedenen Systemen realisiert. Durch die Festlegung wie einzelne Bits in elektrische, optische oder sonstige Signale umzuformen sind, wird die eigentliche, physikalische Verbindung realisiert. Dabei ist die Realisierung dieser Schicht natürlich stark an das verwendete Übertragungsmedium gebunden. Besondere Beachtung wird folgenden Charakteristika geschenkt:

- Mechanik: Spezifikation von Kabel und Stecker, Pin-Belegung
- Elektrik: Zuordnung der logischen Werte 0 und 1 zu physikalischen Größen wie Strom oder Spannung, aber auch Berücksichtigung elektrischer Eigenschaften wie Dämpfung und Bandbreite.
- Funktion: Bedeutung der einzelnen Leitungen und Zeitabläufe, Aussagen über Datenflussrichtungen und Anzahl der Datenleitungen.

Schicht 2: Sicherung (Data Link Layer)

Die Sicherungsschicht sorgt für einen zuverlässigen Austausch von Binärpaketen zwischen den Endgeräten. Diese Schicht bewältigt die folgenden Aufgaben:

- Sicherung der Übertragung: Fehler in der Datenübertragung müssen erkannt und korrigiert werden. Dazu wird Redundanz in die Übertragung eingebracht (Checksummen) anhand derer der Empfänger fehlerhafte Pakete erkennen kann und eventuell korrigieren kann. Ist eine Korrektur

nicht möglich, so kann der Empfänger eine wiederholte Übertragung anfordern.

- Telegrammstruktur: Gruppierung der Bitströme in einzelne Datenpakete
- Zugriffsverfahren: Regelung und Festlegung eines geordneten Zugriffs auf das Medium der Schicht 1
- Synchronisation zwischen Sender und Empfänger
- Adressierung der Teilnehmer: Teilnehmer werden durch die Vergabe von Adressen eindeutig identifiziert.

Oft wird die Sicherungsschicht in die Teilschichten *Medium Access Control* (MAC) und *Logical Link Control* (LLC) unterteilt. Die MAC-Teilschicht steuert den Zugang zur Datenleitung- und bestimmt das Zugriffsverfahren, die LLC-Teilschicht übernimmt die verbleibenden Aufgaben der Schicht 2.

Transportsystem II

- Network Layer (3)
  - Transportiert Pakete über Teilstrecken des Netzes zu Endsystem
  - Wegsteuerung (Routing), Flusskontrolle, Segmentieren
  - Schicht besitzt "Landkarte" des Netzwerks
  - z.B. IP
- Transport Layer (4)
  - Kennt nur mehr Endpunkte des Weges
  - Ist Schnittstelle für Anwendungssystem
  - Zerlegt Nachrichten in Pakete und baut Sie wieder zusammen
  - z.B. TCP, UDP

Schicht 3: Vermittlung (Network Layer)

Diese Schicht ist für den Pakettransport über die Teilstrecken des Netzes zuständig. Dabei obliegt dieser Schicht die Verwaltung von Paketen in den End- und Zwischenstationen. Die Vermittlungsschicht hat folgende Aufgaben zu bewerkstelligen:

- Wegsteuerung (Routing): Abhängig von der Größe und der Struktur des Netzwerkes, können Nachrichten auf verschiedene Wege zum Endsystem gelangen. Die Wegsteuerung soll unter Berücksichtigung von Qualitätskriterien (z.B. maximaler Datendurchsatz und minimale Transportzeit) geeignete Wege finden.

  Um diese Aufgabe bewerkstelligen zu können, besitzt die Vermittlungsschicht eine Landkarte des Netzes oder eines gewissen Teils des Netzwerkes (wenn das Netz sehr groß ist).

  ⓘ Die Wegsteuerung berücksichtigt bei der Wahl des Weges auch, für welchen Dienst Daten übertragen werden sollen. So ist z.B. bei Telnet ein möglichst schneller Weg gefordert während FTP eher großen Datendurchsatz (große Datenblöcke) wünscht.

- Flusskontrolle: Die Flusskontrolle verhindert die Überlastung des Netzes oder einzelner Leitungen. Außerdem verhindert die Flusskontrolle, dass einem (langsameren) Zielrechner mehr Pakete zugestellt werden, als dieser abarbeiten kann. Erst wenn der Rechner wieder bereit ist Pakete zu empfangen, werden vom Sender weitere Pakete geliefert.

- Segmentieren: Pakete der Schicht 3 sind im Normalfall länger als die von der Schicht 2 angebotene maximale Paketlänge. Dadurch werden die Pakete vor der Übergabe an die Schicht 2 zerteilt und am Zielrechner wieder zusammengefügt.

Schicht 4: Transport (Transport Layer)

Diese Schicht fasst die unteren Schichten zum Transportsystem zusammen und ist die Schnittstelle zum Anwendersystem. Die Transportschicht kennt – im Gegensatz zur Schicht 3 – nur mehr die Endteilnehmer.

Die Transportschicht wählt, abhängig von den jeweiligen Anforderungen des Anwendersystems, geeignete Netze für die Übertragung der Daten.

Das Anwendungssystem

Anwendungssystem I

- Session Layer (5)
  - Verwalten Sessions zwischen entfernten Rechnern
  - Authentifizierung, Dialogkontrolle (wer sendet, wer empfängt), Wiederaufsetzen von unterbrochenen Sessions
- Presentation Layer (6)
  - Stellt gemeinsame Sprache für verteilte Applikationen zur Verfügung (Darstellung von Datentypen: Character Encodings, Floating Point Darstellung)

Schicht 5: Sitzung (Session Layer)

Die Sitzungsschicht enthält Funktionen zur Dialogsteuerung und Synchronisation des Datentransfers. Die Dialogsteuerung bestimmt, in welcher Reihenfolge die beteiligten Stationen miteinander kommunizieren. Maßnahmen zur Synchronisation unterstützen eine transaktionsorientierte Kommunikation. Damit wird gewährleistet, dass eine Menge von logisch zueinander gehörenden Zugriffen oder Operationen als gesamtes ausgeführt werden oder gar keiner der Zugriffe behandelt wird (=Transaktion). Die Synchronisation bietet Wiederaufsetzpunkte, falls Transaktionen nicht ausgeführt werden konnten oder sonstige Fehler aufgetreten sind.

Schicht 6: Darstellung (Presentation Layer)

In einem Netzwerk kommunizieren verschiedene Rechnerarchitekturen auf denen verschiedene Programme laufen. Dabei werden Datentypen oftmals unterschiedlich dargestellt und sind zueinander nicht kompatibel. Die Darstellungsschicht stellt Mittel zur Verfügung, die es den Anwendungen ermöglichen, Begriffe eindeutig zu benennen, und legt Regeln fest, wie die Information auszutauschen ist und wie die einzelnen Daten zu codieren sind.

Anwendungssystem II

- Application Layer (7)
  - Stellt Services/Protokolle für Endbenutzer zur Verfügung
  - z.B. DNS, FTP, SMTP, HTTP

Schicht 7: Anwendung (Application Layer)

Auf dieser Schicht befinden sich häufig benötigte Protokolle und Dienste. Dazu gehören z.B. fertige Lösungen für den Dateitransfer (FTP), der Austausch elektronischer Post (E-Mail) oder sonstige Dienste. Anwendungen können dann auf diese Dienste und Protokolle zurückgreifen, ohne sich um die restlichen Details der Vernetzung zu kümmern.

Netzwerktopologien

Netzwerktopologien I

- Verschiedene Möglichkeiten, Rechner zu organisieren
- Bus
  - Jeder Rechner bekommt alle Nachrichten
  - Aufwand für Verkabelung gering
  - Bruch der Verkabelung → Bruch des Netzwerks
  - Ausfall eines Rechners ist egal

Unter Topologie eines Netzwerkes versteht man die räumliche Anordnung der Rechner und Leitungen eines Netzwerkes. Diese Anordnung hat entscheidenden Einfluss darauf, wie die beteiligten Rechner miteinander kommunizieren.

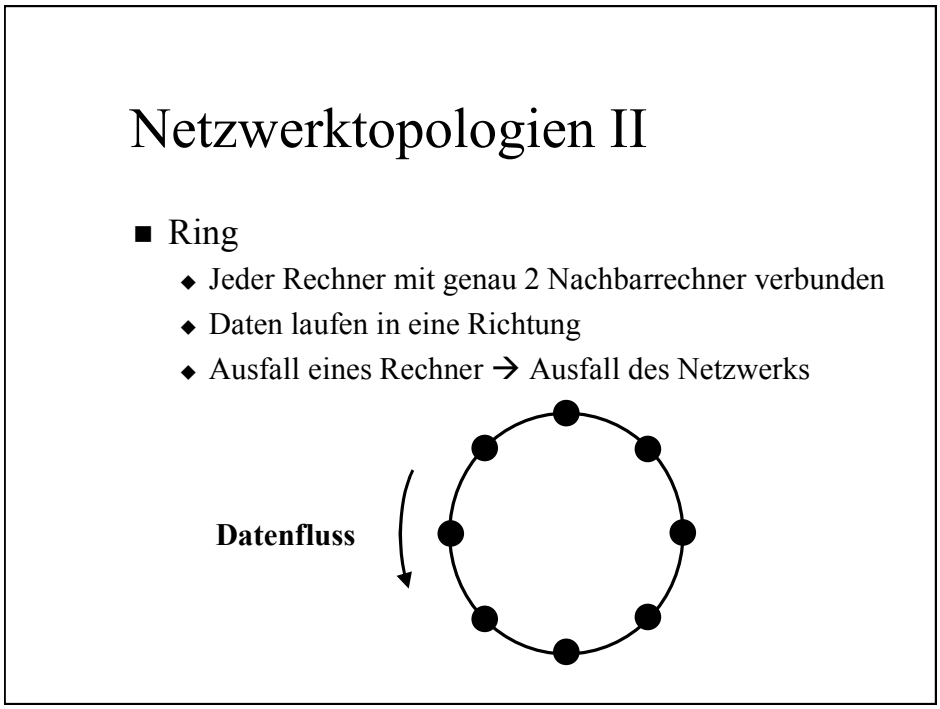
Bus-Topologie

Bei der Bus-Topologie sind alle Rechner an einem Strang angeschlossen. Die einzelnen Pakete werden dabei in den Bus gestellt und durchwandern diesen (in beide Richtungen) bis zum Ende des Busses. Dies hat zur Folge, dass jede Station alle Nachrichten mithört, aber nur die Nachrichten bearbeitet, die an sie gesendet worden sind. Zur Koordinierung der Zugriffe sind geeignete Maßnahmen notwendig.

Beim Ausfall einer Station bleibt der Rest des Netzwerkes funktionsfähig. Bricht jedoch ein Kabel des Busses oder ist die Busankoppelung einer Station defekt, so ist das gesamte Netzwerk funktionsunfähig.

Der große Vorteil dieser Topologie ist der geringe Verkabelungsaufwand, da nur ein Kabel von Rechner zu Rechner gelegt werden muss.

ⓘ Normalerweise durchwandern elektrische Signale in Form von Wellen den Bus. Diese Signale werden am Ende des Busses durch sogenannte Endwiderstände vom Bus genommen. Kann dieser Endwiderstand nicht erreicht werden, so entstehen *stehende Wellen*. Ein weiteres Benutzen des Netzwerks ist somit unmöglich.



Ring-Topologie

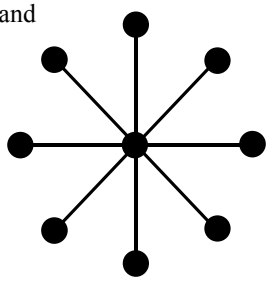
In einem Ring ist jede Station genau mit zwei Nachbarknoten direkt verbunden. Die Leitungen sind meist so ausgeführt, dass ein Knoten über eine Leitung Nachrichten empfängt und über die andere Leitung sendet. Jedoch gibt es eine festgelegte Richtung für den Datenfluss. Fällt eine Station aus oder bricht ein Kabel zu einer Station, so führt dies zu einem Netzwerkausfall.

Die Verkabelung muss bei dieser Topologie einen Ring ergeben. Deshalb muss der letzte Rechner wieder mit dem ersten Rechner verbunden werden, was manchmal zusätzlichen Verkabelungsaufwand mit sich bringt.

ⓘ In einem Ring wird normalerweise eine Berechtigungsmarke (das sogenannte Token) von Rechner zu Rechner gereicht. Der Rechner, der in Besitz dieses Tokens ist, darf die Leitungen für das Senden von Nachrichten benutzen. Damit ist gewährleistet, welcher Rechner zu einem bestimmten Zeitpunkt die Netzwerkleitung exklusiv benutzen darf. Bei einem Rechnerausfall kann jedoch das Token nicht mehr weitergegeben werden und ein Netzwerkausfall ist die Folge.

Netzwerktopologien III

- Stern
  - 1 Zentrale, mit der alle Rechner verbunden sind
  - Kommunikation über Zentrale
  - Ausfall von Zentrale → Ausfall von Netz
  - Ausfall von Rechner oder Netz beeinflusst Netzwerk nicht
  - Hoher Verkabelungsaufwand



Stern-Topologie

Die Stern-Topologie ist zwar die älteste Topologie deren Ursprung in der Host-Terminal-Vernetzung liegt, ist aber noch immer die am häufig genutzte Verkabelungsform. Immer öfters geht man dazu über, Bus- oder Ring-Netzwerke durch eine Stern-Verkabelung zu ersetzen.

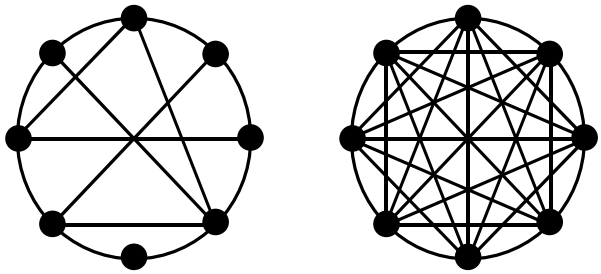
Bei dieser Form von Vernetzung gibt es eine Zentrale im Mittelpunkt, an der alle Teilnehmer angeschlossen sind. Über diese Zentrale läuft die gesamte Kommunikation. Fällt diese Zentrale aus, so zieht dies einen Komplettausfall des Netzwerkes mit sich. Andererseits sind Fehler dieser Art leicht zu diagnostizieren und bei einem Komplettausfall kann nur die Zentrale defekt sein. Ein Ausfall eines Rechners oder das Brechen eines Kabels beeinflusst nur den ausgefallenen bzw. daran angeschlossenen Rechner, das restliche Netzwerk bleibt unberührt. Da hier nur ein Rechner nicht mehr über das Netz zu erreichen ist, ist auch hier eine schnelle Fehlerfindung möglich – die Ursache liegt dann entweder in der Leitung oder in einem Defekt des Rechners selbst.

Dadurch, dass von jeder Station ein eigenes Kabel zur Zentrale gelegt werden muss, ist diese Topologie mit einem hohen Verkabelungsaufwand verbunden. Selbst wenn zwei Rechner nebeneinander stehen – aber weit entfernt von der Zentrale – muss für jeden Rechner ein eigenes Kabel zur Zentrale gelegt werden.

ⓘ Moderne Verkabelungskonzepte legen einen logische Ring- oder Bus-Topologie physikalisch als Stern aus und verbinden somit die Vorteile beider Topologien. Die Zentrale sorgt dafür, dass beim Ausfall einer Station oder einer Leitung nicht das gesamte Netzwerk betroffen ist.

Netzwerktopologien IV

- Teil- bzw. vollvermaschte Netze
  - Redundante Verbindungen werden zur Lastaufteilung verwenden
  - Erhöhte Sicherheit im Netzwerk



Teil- bzw. Vollvermaschte Netze

Diese Art von Topologie ist eine Art Verbindung der zuvor vorgestellten Arten von Vernetzung. Hier wird durch redundante Verbindungen eine Lastaufteilung und erhöhte Sicherheit gewährleistet.

Je nach Intensität der Verkabelung können bei Ausfall eines Rechners oder eines Kabels alternative Routen gewählt werden um so einen Komplettausfall des Netzwerkes zu verhindern. Diese Art von Topologie findet man in Netzwerken, von denen ein extremes Maß an Ausfallssicherheit gefordert wird, oder aber auch in Netzwerken, die durch Kopplung von mehreren LANs entstanden sind.

Übertragungsmedien

Übertragungsmedien I

- Zweidrahtleitung
  - Zwei isolierte, verdrillte Kupferleitungen
  - CAT5-Verkabelung (10BaseT / 100BaseT)
- Koaxialkabel
  - Innenleiter und Außenleiter
  - BUS-Verkabelung (10Base2, 10Base5)
- Lichtwellenleiter
  - Lichtimpulse werden übertragen
  - Sehr schnelle Datenübertragung

Informationen können durch eine Veränderung von physikalischen Größen wie Spannung oder Strom dargestellt werden. Dazu sind geeignete physikalische Medien zur Übertragung notwendig. Einige dieser Medien werden in weiterer Folge vorgestellt.

Zweidrahtleitung

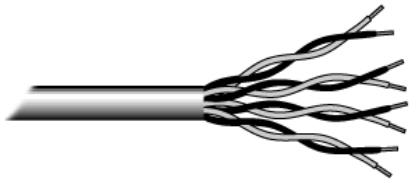


Abbildung 1-1: Ungeschirmte Zweidrahtleitung

Eines der ältesten aber immer noch oft eingesetzten Übertragungsmedien ist die verdrillte Zweidrahtleitung. Sie besteht aus zwei isolierten Kupferleitungen, die in der Regel verdrillt sind (ineinander verdreht). Diese Verdrillung mindert Einstreuungen und Störungen durch benachbarte Leitungen. Oft sind mehrere Leitungspaare gemeinsam mit einer geschirmten Ummantelung versehen, damit das gleiche Kabel auch für andere Zwecke eingesetzt werden kann. Diese Form von Leitung eignet sich sowohl für analoge als auch digitale Übertragung. Die Bandbreite hängt von der Güte der Leitungen und der Ausdehnung ab.

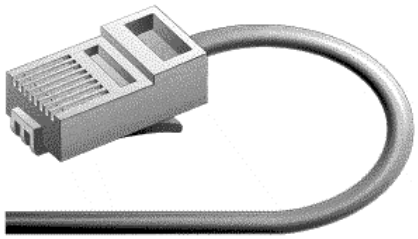


Abbildung 1-2: RJ45-Stecker und CAT5-Kabel (10BaseT, 100BaseT)

Zweidrahtleitungen werden bei der CAT5-Verkabelung, nach dem Ethernetstandard 10BaseT und 100BaseT eingesetzt. Dabei verwenden die Leitungen eine RJ45-Stecker, der auch in den USA als normaler Telefonstecker eingesetzt wird. Zweidrahtleitungen sind in der Regel billig, allerdings auch anfällig gegen Störungen. In der Netzwerktechnik werden diese Kabel mit Signalen im 200 MHz-Bereich gespeist, diese können aber maximal 100 Meter störungs- und verlustfrei übertragen werden.

Koaxialkabel

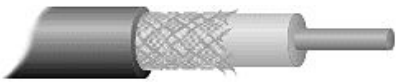


Abbildung 1-3: Koaxialkabel

Koaxialkabel bestehen aus einem Innenleiter aus festem Kupfer und einem zylindrischen Außenleiter aus Kupfergeflecht, der den Innenleiter umschließt. Dazwischen und auch um den Außenleiter befindet sich eine Isolierschicht. Diese Isolierschicht schützt das Kabel gegen Störfelder. Für die digitale Übertragung werden Kabel mit einem Wellenwiderstand von 50 Ohm verwendet. Das Kabel selbst ähnelt sehr einem Antennenkabel für Fernsehgeräte, jedoch verwenden diese 75 Ohm für analoge Datenübertragung. Koaxialkabel werden in der Ethernet-Verkabelung eingesetzt und arbeiten mit Übertragungsraten von 10 MBit/sec.

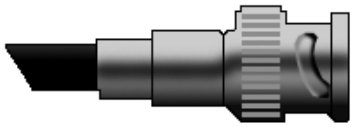


Abbildung 1-4: Koaxialkabel mit BNC-Stecker (10Base2)

Je nach eingesetztem Standard (10Base2 oder 10Base5) unterscheiden sich die Kabel in ihrem Durchmesser und ihrer maximalen Verkabelungsweite. 10Base5-Verkabelungen sind heutzutage nicht mehr im Einsatz. In Abbildung 1-4 zeigt die Anschlussart von 10Base2.

Abbildung 1-5: T-Stück (10Base2-Verkabelung)

Bei dieser Verkabelung wird das Koaxialkabel von Rechner zu Rechner gelegt. Die Netzwerkkarte wird durch ein sogenanntes T-Stück (Abbildung 1-5) mit den Kabeln verbunden. Die äußersten Enden des Kabels werden mit Endwiderständen (Abbildung 1-6) abgeschlossen.

Abbildung 1-6: Endwiderstand (10Base2-Verkabelung)

Diese Art von Verkabelung wird ausschließlich in Bus-Topologien eingesetzt. Dabei wird das Ende des Busses durch die Endwiderstände vorgegeben, die einzelnen Stationen werden über das T-Stück an den Bus angebunden.

Lichtwellenleiter

Abbildung 1-7: Lichtwellenleiter (100BaseFX)

Lichtwellenleiter übertragen Daten in Form von Lichtimpulsen. Ihre Bandbreite ist aufgrund der hohen Frequenz des Lichtes enorm hoch. Das Einspeisen der Lichtimpulse in den Leiter erfolgt durch eine LED (*light emitting diode*) oder eine Laserdiode. Der Empfänger besitzt eine Photodiode mit der er die Lichtimpulse wieder zurückwandeln kann.

Lichtwellenleiter sind – wenn sie richtig verlegt werden –resistent gegen jegliche Form von Störungen. Deshalb sind sehr hohe Übertragungsgeschwindigkeiten und sehr hohe Distanzen möglich (z.B. spezifiziert der FDDI-Standard eine Übertragung von100 MBit/sec über eine Länge von 200 km).

Übertragungsmedien II

- Richtfunk
  - Funktechnik zur Übertragung
  - Wenn traditionelle Verkabelung nicht möglich
  - Strahlfunk (in bestimmte Richtung)
- Satelliten
  - Interkontinentale Verbindungen
  - Hohe Übertragungsraten
- Wireless LAN
  - Neue Technologie
  - Kugelförmig ausbreitende Funkverbindung

Richtfunk

Richtfunk benutzt eine elektromagnetische Welle als Träger und moduliert dieser das zu übertragende Signal auf. Eingesetzt wird dies vor allem in unwegsamen, dünn besiedelten Gebieten oder über sehr weite Entfernungen, wenn seine konventionelle Verkabelung technisch nicht rentabel ist oder nicht möglich ist. Zwischen Sender und Empfänger ist jedoch eine Sichtverbindung notwendig, darum sind die Einsatzmöglichkeiten sehr beschränkt.

Satelliten

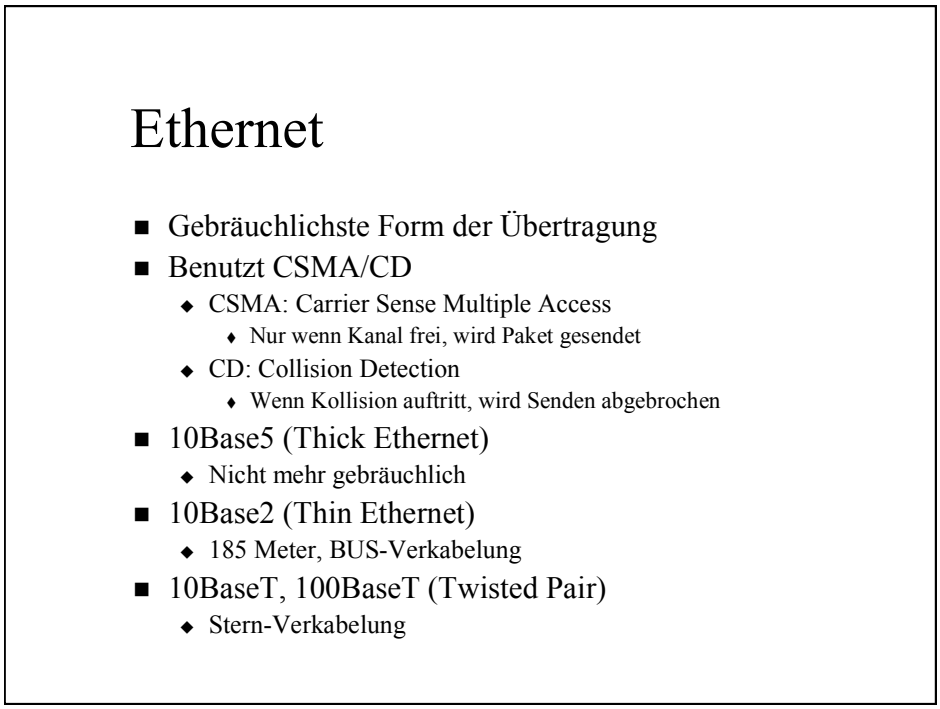
Kommunikationssatelliten eignen sich für interkontinentale Verbindungen und für Rundsendesysteme bei denen viele Empfänger die gleiche Nachricht hören. Wegen der großen Entfernung unterliegen die Signale bei geostationären Satelliten einer Verzögerung von ca. 200 msec. Für das Übertragen von Daten stehen Bandbreiten bis ca. 50 MBit/sec. Zur Verfügung.

Wireless LAN

Wireless LAN ist eine sehr junge Technologie und stellt eine Alternative zu verkabelten Netzwerken da.

Wireless LANs benutzen die Funktechnologie zur Datenübertragung. Anders als beim Richtfunk breiten sich hier die Funkwellen kugelförmig aus. Derzeit besitzen Wireless LANs durch das relativ junge Entwicklungsstadium noch sehr kurze Reichweiten und niedrige Übertragungsraten.

Das Ethernet-Protokoll



Ethernet ist das gebräuchlichste Kommunikationssystem das in LANs benutzt wird. Es ist ein kollisionserkennendes Protokoll und basiert auf einer ungesteuerten Übertragung. Das heißt, es ist nicht festgelegt, wer zu einer bestimmten Zeit senden darf. Die zugrundeliegende Leitung wird dabei allen Benutzern zur gleichen Zeit zur Verfügung gestellt. Im Konfliktfall, also wenn mehrere Stationen zugleich senden, werden Sendungen unbrauchbar da sich Signale überlagern. In einem solchen Fall erfolgen neue Sendeversuche, die wiederum konfliktbehaftet sein können. Diese Fakten führen dazu, dass die zur Verfügung stehende Bandbreite nur selten voll ausgenutzt werden kann.

Zugriffsverfahren

Um Kollisionen trotzdem gering zu halten, benutzt Ethernet das CSMA/CD-Verfahren. Bei CSMA (Carrier Sense Multiple Access) hört die Ethernetkarte auf der Leitung, ob diese bereits benutzt wird. Nur wenn die Leitung leer ist, d.h. es findet zur Zeit keine Übertragung statt, wird das Signal auf die Leitung gelegt. Diese Technik senkt das Konfliktrisiko bereits erheblich, jedoch ist es immer noch möglich, dass zwei Stationen zur gleichen Zeit senden, da beide festgestellt haben, dass die Leitung frei ist und somit benutzt werden kann. Eine weitere Verbesserung wird durch die Kollisionserkennung (Collision Detection – CD) erreicht. Die Netzwerkkarte hört während des Sendens mit und kann so feststellen, ob die Daten so auf die Leitung gelangen, wie sie gesendet wurden. Wird ein zweites Signal festgestellt (also sendet zur gleichen Zeit eine andere Karte), so wird die Sendung sofort abgebrochen. Da auch die zweite Karte diesen Konflikt mitbekommt (diese hört ja auch während des Sendens) werden alle laufenden Sendungen bei einem Konflikt sofort abgebrochen. Nach einer zufälligen Zeit wird versucht, die Sendung zu wiederholen. Da jede Station vor dem Senden die Leitung

abhört, kann eine Kollision nur am Beginn des Sendevorganges auftreten. Die Zeitspanne, innerhalb der eine Kollision möglich ist, ergibt sich aus der maximalen Ausbreitungszeit der Signale auf der Leitung.

Spezifikationen

Ethernet liegt in verschiedenen Spezifikationen vor, die sich in der Art des Mediums, der Geschwindigkeit und der maximalen Länge (ohne Verstärker) voneinander unterscheiden. Ein Teil der einzelnen Spezifikationen ist in Tabelle 1-1 angeführt.

Spezifikation	Kabeltyp	Geschwindigkeit	Maximale Länge
10BaseT	Ungeschirmte Zweidrahtleitung	10 MBit/sec	100 Meter
100BaseT	Ungeschirmte Zweidrahtleitung	100 MBit/sec	100 Meter
100BaseTX	Ungeschirmte Zweidrahtleitung	100 MBit/sec	220 Meter
10Base2	Dünnes Koaxialkabel	10 MBit/sec	185 Meter
10Base5	Dickes Koaxialkabel	10 MBit/sec	300 Meter
100BaseFX	Lichtwellenleiter	100 MBit/sec	200 km
1000BaseX	Ungeschirmte Zweidrahtleitung	1000 MBit/sec	100 Meter

Tabelle 1-1: Auszug aus Ethernet-Spezifikationen

ⓘ Ethernet selbst wird logisch immer als Bus gesehen, die Kabel oftmals aber durch Übertragungseinheiten physikalisch als Stern verlegt. Der Einsatz von Switches und Router begrenzt zudem die Kollisionsdomäne (der Bereich, in dem Kollisionen auftreten können).

! Wichtig ist, zu verstehen, dass trotz Einsatz eines HUBs das Ethernet-Protokoll logisch ein Bus ist. Ein Signal, dass von einer Station an die zentrale Einheit (HUB) gesandt wird, wird an alle an das HUB angeschlossenen Leitungen verteilt. Trotzdem werden alle Nachteile der Bus-Verkabelung durch die Stern-Verkabelung überwunden, da physikalisch ja ein Stern vorliegt.

Kopplung von Netzwerken

Kopplung von Netzen I

- Verschiedene Netze werden über Rechner verbunden (*relay)* – Unterscheidung auf welcher Schicht Netze verbunden werden
- Repeater
  - Arbeitet auf Schicht 1
  - Verstärkt Signal oder ändert Codierung des Signals
  - Bspl.: HUB
- Bridge
  - Arbeitet auf Schicht 2 (Sicherungsschicht)
  - Verbindet Teilnetze (auch unterschiedliche Topologien)
  - Selektive Paketweiterleitung (Bridges „lernen" Adressen der Teilnehmer → Netzwerkentlastung)
  - Bridge ist auch Repeater
  - Bspl.: Switch

Manchmal ist es wünschenswert und notwendig, dass verschiedene Netzwerke miteinander verbunden werden. Diese Kopplung wird durch ein Gerät bewerkstelligt, welches in der OSI-Terminologie als *Relais* bezeichnet wird. Solche Geräte werden anhand der Schicht klassifiziert, in der sie ihre Aufgabe bewältigen.

Repeater

Repeater sind einfache Verstärker, die elektrische Signale entweder aufbessern oder die elektrische Codierung des Signals verändern. Sie kennen die Struktur der Pakete nicht, ihnen ist lediglich die Codierung von Bits in elektrische Signale bekannt. Repeater sind auf Schicht 1 (Bitübertragung) angesiedelt.

Beispiele für Repeater sind HUBs, die lediglich ein ankommendes Signal verstärken und an alle angeschlossenen Stationen weiterleiten. Der Paketinhalt selbst wird vom HUB nicht analysiert oder verändert. Manchmal sind HUBs auch in der Lage, Signale von einer ankommenden Zweidrahtleitung (CAT5) in ein Koaxialkabel weiterzuleiten (10Base2).

Bridge

Bridges verbinden Netzwerke auf der Sicherungsschicht (Schicht 2). Diese Schicht kennt bereits die Hardwareadressen der einzelnen beteiligten Stationen und ist auch in der Lage, die Adressdaten der Pakete zu analysieren. Der Einsatz von Bridges ist dann sinnvoll, wenn die beteiligten Netze verschiedene Sicherungsschichten aber dieselbe Vermittlungsschicht besitzen. Aber auch wenn die zu koppelnden Netze die gleiche Sicherungsschicht benutzen, macht der Einsatz einer Bridge Sinn, da sie zur Lastentkoppelung dienen kann. Die Bridge kennt nicht nur die Adressdaten der Empfänger, sondern weiß auch, welcher Rechner an die Bridge angeschlossen ist.

Das Paket wird nicht mehr – wie bei einem Repeater – an alle anderen Ports weitergeleitet, sondern nur mehr an den Port der Bridge, an dem der Rechner angeschlossen ist oder über den der Rechner erreicht werden kann.

Bridges sind in der Regel intelligent, d.h. sie lernen durch Zuhören die Adressen der Teilnehmer und wissen später, an welchem Port welcher Rechner hängt. Die Zuordnung von Adresse und Rechner wird auf dieser Schicht mittels der eindeutigen Hardware-Adresse der Netzwerkkarte geregelt.

Bridges übernehmen zudem auch noch die Funktionalität eines Repeaters.

Beispiele für Bridges sind Switches, die Pakete nicht mehr an alle angeschlossenen Rechner weiterleiten, sondern nur mehr an das Port, an dem der Rechner hängt bzw. wenn mehrere Switches miteinander verbunden werden, an das Port über das der Rechner erreicht werden kann. Rechner, die an Switches angeschlossen werden, bekommen nur mehr Pakete, die auch wirklich an diese adressiert sind – alle anderen Pakete werden ihnen erst gar nicht zugestellt. Dadurch wird nicht nur das Netzwerk entlastet, es ist auch möglich, dass mehrere Rechner zur gleichen Zeit senden und empfangen, da der Switch mögliche Kollisionen verhindert.

Kopplung von Netzen II

- Router
  - Arbeiten auf Schicht 3 (unterschiedliche Vermittlungsschicht)
  - Verbinden verschiedene Topologien
  - Kennen das Datenübertragungsprotokoll
  - Aufgaben: Flusskontrolle, Vermeiden von Zyklen, Bestimmen der Wege
- Gateway
  - Für Netze, die sich bereits in der Anwendungsschicht unterscheiden
  - Übersetzt Datendarstellung zwischen Topologien

Router

Router werden eingesetzt, wenn die Netzwerke zwar dieselbe Transportschicht, aber verschiedene Vermittlungsschichten haben. Router sind auf Schicht 3 (Vermittlungsschicht) angesiedelt. Um ihre Aufgabe bewerkstelligen zu können, muss ihnen die Topologie der angeschlossenen Netzwerke bekannt sein. Zu ihren Aufgaben zählen das Bestimmen geeigneter Wege, die Flusskontrolle und das Vermeiden von Zyklen.

Anders als Bridges und Repeater, werden Router im Netz explizit adressiert.

Gateway

Gateways verbinden Netze, deren Protokolle sich bereits in den Anwendungsschichten voneinander unterschieden. Hier muss Kopplung über das Anwendungsprogramm erfolgen.

Gateways werden meist zur Verbindung unterschiedlicher Netzwerke verwendet und übersetzen die Datendarstellung zwischen den Netzen.

Kapitel

2

TCP/IP-Protokoll

Dieses Kapitel erläutert die TCP/IP-Protokollfamilie, zeigt deren Konzepte und Anwendungsgebiete. Dabei wird das Protokoll selbst vorgestellt, deren Konzepte und Anwendungsgebiete.

Das TCP/IP-Protokoll hat durch das Internet sehr stark an Bedeutung gewonnen. Ursprünglich war es das Standardprotokoll in UNIX-Systemen, löst aber zunehmend andere Protokolle ab und ist mittlerweile das am häufig genutzte Netzwerkprotokoll. Dieses Kapitel stellt die TCP/IP-Protokollfamilie vor, erläutert deren Konzepte und zeigt die Anwendungsgebiete. Im speziellen wird der Aufbau von IP-Adressen im Zusammenhang mit Subnet-Masken näher eingegangen, aber auch der Zusammenhang zu DNS-Namen erklärt. Neben dem IP selbst, werden auch die darauf aufgesetzten Protokolle TCP und UDP vorgestellt und näher erläutert.

Lehrinhalte und -ziele

Am Ende dieses Kapitels sollten Sie die TCP/IP-Protokollfamilie kennen und deren Konzepte. Sie sollten den Aufbau der Protokolle IP, TCP und UDP kennen und deren Einsatzgebiete. Darüber hinaus sollten Sie verstehen, wie IP-Adressen aufgebaut sind, welcher Zusammenhang zwischen IP-Adressen und Subnet-Masken besteht und wie IP-Adressen und DNS-Namen zusammenspielen. Letztendlich sollten Sie wissen, wie eine Kommunikation zwischen zwei Rechnern über TCP/IP funktioniert und welche Protokolle daran beteiligt sind.

Grundlagen zu TCP/IP

Eigenschaften

- TCP/IP eng mit UNIX verbunden
- Internet basiert auf TCP/IP
- Netzwerk ohne zentrale Verwaltung
- Bei Ausfall von Netzwerkteilen soll restlichen Netzwerk noch funktionieren
- → TCP/IP dazu geeignet
- Standardisierung mit RFCs (Request for Comment)
- www.rfc-editor.org

TCP/IP war von je her das Standardnetzwerkprotokoll in UNIX. Durch die rasante Entwicklung des Internets, das am Anfang eigentlich fast ausschließlich aus UNIX-Rechnern bestand, etablierte sich TCP/IP rasch zu einem Standard der auch schnell in anderen Betriebssystemen verfügbar war.

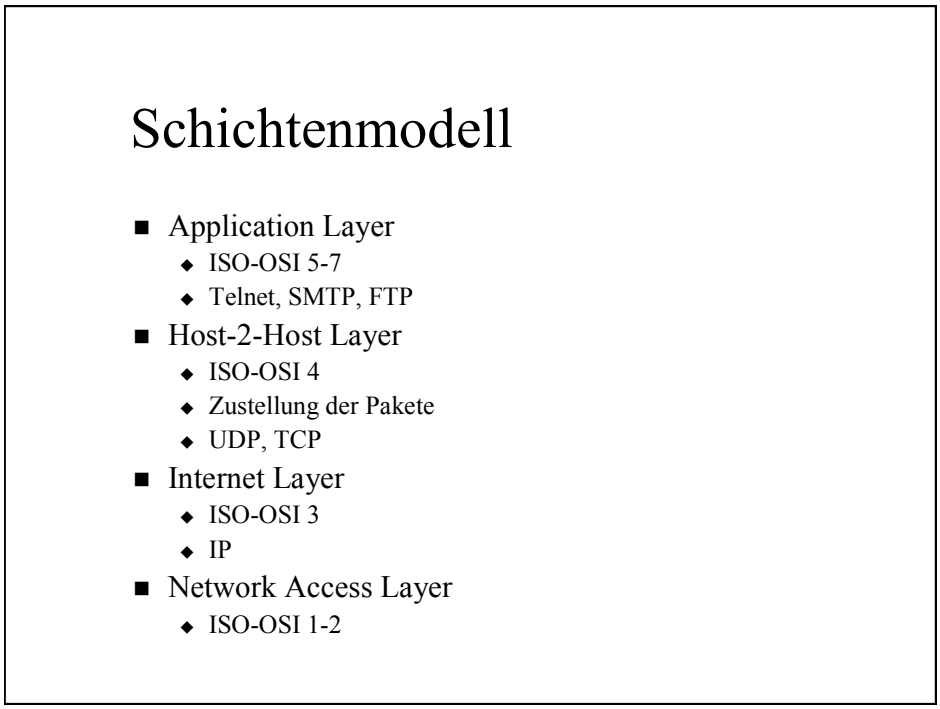
Die ursprüngliche Entwicklung initiierte das amerikanische Verteidigungsministerium (Department of Defence) in den frühen 70er Jahren. Damals konzentrierten sich Forschungsarbeiten auf die Entwicklung eines fehlertoleranten Netzwerkes, dass ohne zentrale Wartung zu betreiben ist. Das Ergebnis war das ARPANET und war der Vorgänger des heute bekannten Internets. Als Protokoll für dieses Netzwerk wurde das TCP/IP-Protokoll gewählt, nicht zuletzt weil es den Anforderungen des DoDs gerecht wurde.

TCP/IP ist ein Protokoll, dass dafür entwickelt worden ist, ohne zentrale Wartung auszukommen. Dies ist besonders wichtig bei einem unüberschaubaren Netzwerk wie dem Internet, wo täglich mehrere tausend Rechner hinzukommen und andere wieder wegfallen. Darüber hinaus ist es sehr fehlertolerant, d.h. auch nach einem Ausfall einzelner Knotenpunkte kann das Netzwerk seine Funktionalität noch erfüllen (wenn auch eingeschränkt) – in diesem Fall wählt das Protokoll selbstständig eine Alternativroute für die Übertragung.

Das TCP/IP-Protokoll hat viele verschiedene Teilaspekte, die dokumentiert und standardisiert sind. Die Standardisierung und Dokumentation erfolgt dabei anhand der sogenannten *Requests for Coments* (*RFCs*). Ein RFC ist im Prinzip ein Bericht, der einen bestimmten Teilaspekt beschreibt. Zuerst besitz jedes RFC einen Vorschlagscharakter und wird erst nach einer gewissen Diskussionsphase durch die Internetgemeine zu einem verbindlichen Dokument. Änderungen und

Klarstellungen zu einem RFC müssen dann wieder als neuer RFC eingebracht werden, ein bestehendes RFC kann nicht mehr verändert werden und dient sozusagen als Norm für Implementierungen, die den darin beschriebenen Teilaspekt realisieren. Durch diese Organisation steigt die Anzahl der RFCs schnell an. Jedes RFC enthält deswegen auch Verweise auf die Vorgängerversion oder auf andere, mit dem Thema zusammenhängende RFC-Dokumente. Die Liste von RFCs ist z.B. im Internet auf der Seite „www.rfc-editor.org“ verfügbar.

Die Abbildung von TCP/IP im Schichtenmodell



Das Protokoll TCP/IP besteht aus mehreren, hierarchisch angeordneten Schichten, die jeweils eine bestimmte Teilaufgabe übernehmen. Im vorigen Kapitel wurde bereits das ISO-OSI Referenzmodell vorgestellt, dass eine Aufteilung des Netzwerkverkehrs in verschiedene Schichten definiert. Die Schichten des TCP/IP-Protokolls lehnen sich an diese Schichten an, jedoch werden aus Simplifizierungsgründen lediglich vier statt der sieben Schichten verwendet. Dies entsteht dadurch, dass mehrere ISO-OSI Schichten zu einer Schicht im TCP/IP-Protokoll zusammengefasst werden.

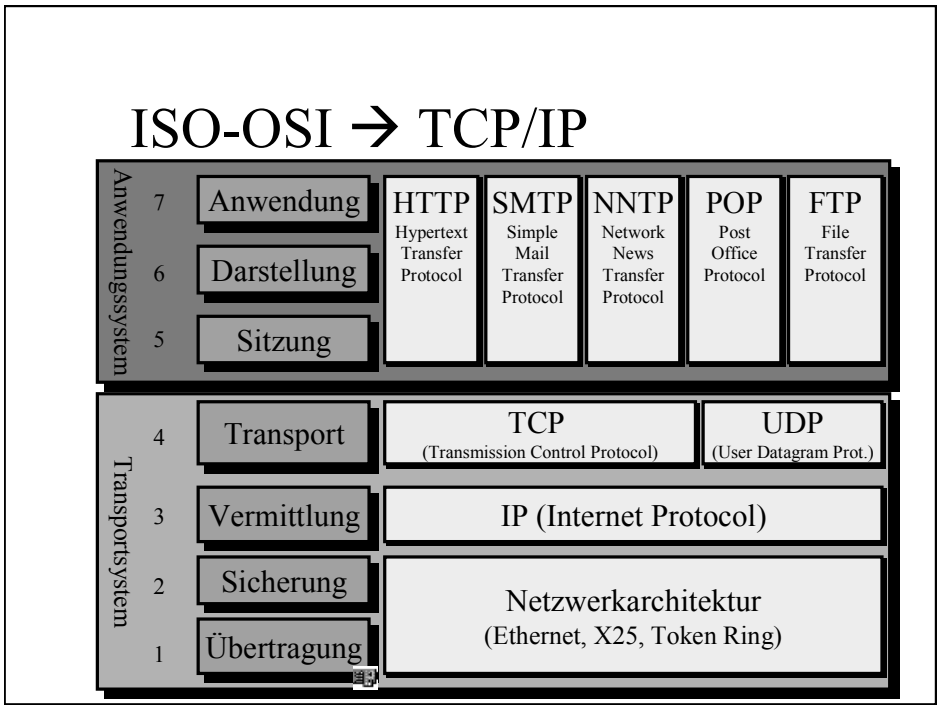
Im folgenden werden nun die einzelnen Schichten des TCP/IP-Protokollstacks näher erläutert.

Application Layer

Der Application Layer ist die oberste Ebene und entspricht den Schichten 5 bis 7 im ISO-OSI Referenzmodell. Hier sind die Applikationsprotokolle wie Telnet, SMTP (Simple Mail Transfer Protokoll), FTP und sonstige – aus dem Internet bekannte – Dienstprotokolle enthalten.

Host to Host Layer

Der Host to Host Layer oder Transport Layer ist die Basis der Applikationsprogramme. Diese Schicht geht bereits von einer existierenden Verbindung zwischen den Endteilnehmern im Netzwerk aus. Endteilnehmer sind auch unter dem Begriff ‚Hosts' bekannt. Der Host to Host Layer kümmert sich um die Zustellung der Netzwerkpakete zu den richtigen Prozessen innerhalb des Zielrechners. Dies ist deswegen notwendig, weil mehrere Anwendungen am Zielrechner zur gleichen Zeit unterschiedliche Dienste des Application Layers



benutzen können (z.B. E-mail und WWW). Ein ankommendes TCP/IP-Paket muss in diesem Falle eindeutig einer Applikation zugeordnet werden (ein WWW-Browser kann im Normalfall mit einem E-mail Paket nichts anfangen).

Diese Schicht verwendet für ihre Aufgabe eigene Protokolle, die auf dem Protokoll der darunterliegenden Schicht aufsetzen. Diese sind das UDP (*User Datagarm Protocol*) und das TCP (*Transmission Control Protocol*). Auf die Eigenschaften dieser Protokolle wird noch später in diesem Kapitel eingegangen.

Der Host to Host Layer entspricht der Schicht 4 des ISO-OSI Referenzmodells.

Internet Layer

Der Internet Layer ist für die Übertragung von Nachrichten zwischen den realen Endgeräten zuständig. Dazu benutzt er IP (*Internet Protocol*). Über dieses Protokoll können Nachrichten von zwei Endgeräten über ein Netzwerk – auch über mehrere Router hinweg – übertragen werden. IP ist für die Router transparent und kann von Routern entsprechend gelesen und verarbeitet werden. Die Aufgaben und Eigenschaften dieses Protokolls werden an späterer Stelle in diesem Kapitel noch genauer behandelt. Weitere Protokolle dieser Schicht sind ARP (Address Resolution Protocol) und ICMP (Internet Control Message Protocol). Diese Protokolle ermöglichen unter anderem die Zuordnung von IP-Adressen zu realen Netzwerkkarten-Adressen.

Im ISO-OSI Referenzmodell entspricht der Internet Layer der Schicht 3.

Network Access Layer

Diese Schicht ist die unterste Schicht im TCP/IP-Protokoll und entspricht somit den Schichten 1 und 2 des ISO-OSI Referenzmodells. TCP/IP setzt hier auf bewährte Schicht-2-Protokolle wie Ethernet, Token Ring und ähnlichen auf. Das Ethernet-Protokoll ist dabei das heutzutage gebräuchlichste Protokoll und wurde bereits im vorigen Kapitel näher vorgestellt.

Der Internet Layer

Internet Layer

- Entspricht ISO-OSI Layer 3
- In diesem Layer ist IP angesiedelt
  - IP ist verbindungsloses Protokoll
- Aufgaben
  - Routing der IP-Datagramme
  - Adressierung der Rechner
  - Fragmentierung der Datagramme
  - *Best Effort*-Zustellung

Wie schon angesprochen, entspricht der Internet-Layer der Schicht 3 des ISO-OSI Referenzmodells. Damit ist diese Schicht für die Kommunikation zwischen Geräten zuständig, die nicht direkt miteinander verbunden sind. Um diese Aufgabe bewerkstelligen zu können, sind in dieser Schicht verschiedene Protokolle angesiedelt. Im folgendem werden diese Protokolle erklärt.

Internet Protocol (IP)

IP ist ein verbindungsloses Protokoll. Das bedeutet, dass beim Senden keine Verbindung zum Zielrechner aufgebaut wird, sondern einfach probiert wird, die Nachricht zu übermitteln. Damit kann aber auch nicht garantiert werden, dass das gesendete Paket jemals den Zielrechner erreicht bzw. dessen Antwort wieder den Sender.

Das Internet Protocol hat folgende Aufgaben:

- Routing der IP-Datagramme
- Adressierung der Rechner
- Fragmentierung der Datagramme, falls dies gefordert ist
- Eine Punkt-zu-Punkt Sicherung durch Schicht 2 (wird durch das Ethernet-Protokoll gewährleistet) jedoch keine End-to-End Sicherung zwischen Sender und Empfänger. Das bedeutet, dass zwar eine Sicherung von Netzwerkkarte zu Netzwerkkarte gewährleistet wird, da sich aber auf dem Weg mehrere Router befinden, betrifft diese Sicherung jeweils nur das

betroffene Segment (also im ersten Schritt der Sendercomputer und der erste Router). Eine Sicherung zwischen dem Quellrechner und dem Zielrechner wird jedoch auf dieser Ebene nicht gewährleistet.

- Prüfsumme wird nur über den Header generiert, nicht aber über die übertragenden Daten

- Eine endliche Lebensdauer des IP-Datagramms vermeidet Zyklen und entfernt das Datagramm, falls dieses nach längerer Zeit nicht zugestellt werden kann um unnötige Netzlasten zu vermdeiden

- Best Effort Zustellung: Hier wird versucht, nach bestem Wissen und Gewissen das Datagramm dem Zielrechner zuzustellen, jedoch kann eine Zustellung nicht gewährleistet werden.

Internet Control Message Protocol (ICMP)

Das ICMP ist für Managementaufgaben zuständig. Dieses Protokoll wird von Routern verwendet, um unvorhersehbare Ereignisse zu melden. Darunter fallen z.B. die Überlastung (Flusskontrolle) oder der Ausfall des Zielrechners (Host kann nicht erreicht werden). Auch Hosts können mit solchen Paketen die Erreichbarkeit anderer Teilnehmer testen (z.B. durch das ping-Kommando). ICMP-Nachrichten werden in IP-Datagramme eingepackt.

Address Resolution Protocol (ARP und RARP)

Rechner werden im Internet und in UNIX-Netzen anhand ihrer IP-Adresse identifiziert. Dabei muss die IP-Adresse innerhalb eines Netzwerkes eindeutig sein. Jedoch ist die IP-Adresse nicht an eine bestimmte Netzwerkkarte gebunden – immerhin kann eine Netzwerkkarte ausgetauscht werden und die gleiche IP-Adresse verwendet werden oder eine andere IP-Adresse ein und derselben Netzwerkkarte zugeordnet werden.

Die Netzwerkkarte selbst wird auf Schicht 2 des ISO-OSI Referenzmodells über die sogenannte MAC-Adresse oder Kartennummer identifiziert. Diese MAC-Adresse ist eindeutig in der ganzen Welt, d.h. es kann keine zwei Netzwerkkarte mit der selben MAC-Adresse geben. Diese Adresse ist fix auf der Hardware der Netzwerkkarte installiert und kann auch nicht geändert werden.

Die Zuordnung zwischen Kartennummer und IP-Nummer eines Rechners wird nicht im Netzwerk zentral abgelegt, sondern mit dem *Adress Resolution Protocol* (ARP) ermittelt. Voraussetzung dazu ist, dass jeder Rechner die eigene IP-Nmmer und auch die eigene Kartennummer kennt. Da die IP-Adresse in eine Rechner- und eine Netzwerkadresse geteilt wird und alle Rechner innerhalb eines Netzwerkes die gleiche Netzwerkadresse haben, ist die MAC-Adresse nur für die Rechner im gleichen Netz interessant. Router leiten IP-Pakete anhand der Netzwerkadresse weiter, erst innerhalb des Netzes muss der Zielrechner anhand der realen MAC-Adresse gefunden werden.

Um die MAC-Adresse zu einer bekannten IP-Adresse zu erfahren, setzt ein Rechner einen Schicht-2-Boradcast ab. Bei einem Boradcast wird von einem Sender ein Netzwerkpaket ins Netz gesendet, dass an alle Rechner adressiert ist und von allen Rechnern im Netz bearbeitet. Somit kann jeder Rechner im Netz

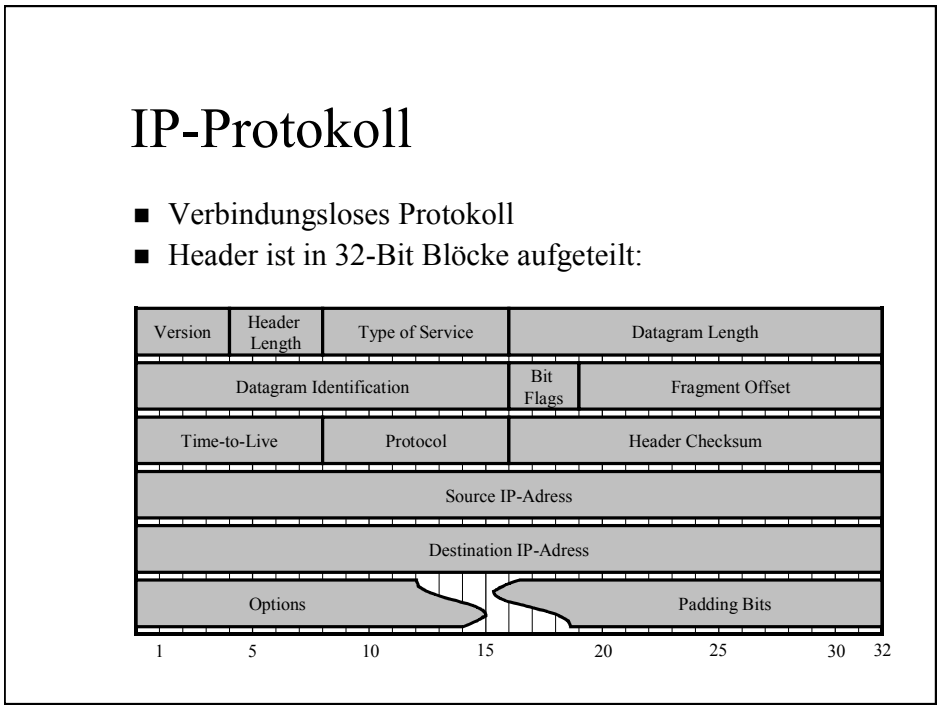
vergleichen, ob er derjenige Rechner ist, dessen IP-Adresse erfragt wird. Ist dies der Fall, so antwortet dieser Rechner und teilt dem Sender seine MAC-Adresse mit.

Der Internet Layer speichert die erfahrenen Daten in einem Cache ab. Damit werden weitere Zugriffe innerhalb des Netzwerkes beschleunigt.

In UNIX kann der aktuelle ARP-Cache mit dem Kommando „arp –a“ eingesehen werden.

Das *Reverse Adress Resolution Protocol* geht den umgekehrten weg und liefert zu einer MAC-Adresse die zugehörige IP-Adresse.

Das IP-Protokoll



Der Header des IP-Protokolls ist in Blöcken zu je 32 Bit unterteilt und besteht einerseits aus einem festen Teil mit 5x32 Bit und andererseits zu einem Teil variabler Länger der eventuelle weitere Optionen beinhaltet. Um wieder die Blockgröße von 32 Bit zu erreichen, wird der Platz hinter den Optionen mit Füllbits wieder auf die entsprechende Größe gebracht. Im folgenden werden die einzelnen Felder des Headers behandelt.

Version (4 Bit)

Dieses Feld enthält die Version des IP-Layers und bestimmt damit die Struktur der nachfolgenden Datenfelder. Dieses Feld erlaubt die gleichzeitige Verwendung unterschiedlicher Versionen des Internet Protocols. Derzeit ist weitläufig Version 4 in Verwendung, jedoch steht eine Ablösung durch Version 6 unmittelbar bevor.

ⓘ Neuere Versionen von Betriebssystemen (Windows 2000, Solaris 8, neuere Versionen von Linux-Distributionen) bietet bereits IPv6-Unterstützung an. Diese Protokollversion bietet nicht nur einige Performanceverbesserungen sondern auch einen viel größeren Adressraum. IP-Adressen in IPv6 sind 128 Bit lang – das zur Zeit genutzte IPv4 bietet nur 32 Bit lange Adressen an.

Header Length (4 Bit)

Da optional am Ende des Headers noch Optionen übergeben werden können, muss irgendwo festgehalten werden, wie langer der Header nun tatsächlich ist. Das Feld ‚Header Length' gibt dabei die Länge des Headers in 32-bit Worten an, also wie viele 32-Bit-Zeilen der Header umfasst.

Type of Service (8 Bit)

Dieses Feld enthält Informationen über die gewünschten Übertragungswege. Router berücksichtigen diese Werte bei der Auswahl der Wege. Das Feld unterteilt sich in folgende Einzelfelder:

- Precedence (3 Bit): Hier kann dem Paket eine Priorität (0 – 7) übergeben werden.

- Low delay (1 Bit): Ist dieses Bit gesetzt, so ist auf geringste Verzögerungszeiten zu achten (schnellste Übermittlung).

- High troughput (1 Bit): Ist dieses Bit gesetzt, so ist auf einen hohen Datendurchsatz zu achten. Eine Route, die den höchsten Datendurchsatz anbietet muss nicht zwangsweise auch die schnellste Übertragung anbieten. Manchmal erlauben gewisse Wege eine längere Paketgrösse als andere – schnellere Wege. Durch das Aufteilen der Pakete in viele kleine Pakete würde die Gesamtübertragung aber länger dauern als wenn das gesamte Paket ungeteilt über eine etwas langsamere Leitung geschickt würde.

- High reliability (1 Bit): Ist dieses Bit gesetzt, so ist eine Verbindung mit hoher Zuverlässigkeit zu wählen.

- Unused (2 Bit) Die letzten beiden Bit des Feldes ‚Type of Service' werden nicht benutzt.

Total Length (16 Bit)

Dieses Feld enthält die Gesamtlänge des IP-Pakets (IP-Datagrammes), d.h. hier steht die Länge des Headers und der Nutzdaten. Die maximale Länge eines IP-Paketes ergibt sich aus der Feldlänge, also 2^{16} Bytes.

Identification (16 Bit)

Der Wert in diesem Feld identifiziert das Paket eindeutig. Es wurde bereits angesprochen, dass es manchmal nötig ist, ein IP-Paket auf mehrere kleinere Pakete (Fragmente) aufzuteilen. Alle Fragmente haben dann den gleichen Wert im Identification-Feld.

Bit Flags (3 Bit)

In diesem Feld können Flags angegeben werden, die bei einer eventuellen Fragmentierung berücksichtigt werden. Benutzt werden allerdings nur zwei der drei Bits:

- Don't fragment, D-Bit: Ist dieses Bit gesetzt, so wird der Router angewiesen eine Alternative zu wählen, in der eine Fragmentierung des Paketes nicht nötig ist und als gesamtes übertragen werden kann.

- More fragments, M-Bit: Dieses Bit ist für das Zusammenstellen der Fragmente wichtig. Ein gesetztes M-Bit zeigt an, dass zu diesem Paket noch weitere Fragmente folgen.

Fragment Offset (13 Bit)

Dieses Feld bestimmt die Position des vorliegenden Fragmentes im gesamten IP-Paket. Der Wert dieses Feldes entspricht dem Offset in Bytes vom Anfang geteilt durch 8. Das ist deshalb so, weil ein Fragment immer einen Datenteil enthält, dessen Länge durch 8 teilbar ist. Um Platz bei der Indizierung zu sparen, wird hier die Länge geteilt durch 8 angegeben.

Time-to-live (8 Bit)

Dieses Feld definiert die maximale Zeit, die ein Paket für die Zustellung zum Zielrechner benötigen darf. Der Wert – angegeben in Sekunden – wird vom Quellrechner gesetzt. Da die tatsächliche Zeit schwer zu bestimmen ist, dekrementiert jeder Router den Wert dieses Feldes um einen bestimmten Wert – in der Regel um 1. Damit gibt dieses Feld eigentlich die Anzahl der möglichen *hops* – das sind die Sprünge über Router – zum Zielrechner an. Erreicht der Wert vor seiner Ankunft den Wert 0, so entfernt der aktuelle Router das Paket vom Netzwerk. Damit werden zirkulierende oder unzustellbare Pakete vom Netz weggenommen.

Protocol (8 Bit)

Dieses Feld verweist auf das höhere Protokoll. Dies ermöglicht die Zustellung im Zielrechner zur richtigen höheren Schicht. Konkret kann dieses Wert die Nummern 1 (ICMP), 6 (TCP) und 17 (UDP) enthalten. Details dazu findet man im RFC 1700 (Assigned Numbers).

Header Checksum (16 Bit)

Dieses Feld enthält eine Prüfsumme über den Header. Dies dient zur Sicherung des IP-Pakets.

! Die Prüfsumme wird jedoch nur über den Header gebildet, nicht über das gesamte IP-Paket. Fehler in den Nutzdaten können also vom Internet Procol selbst nicht festgestellt werden. Dies ist Aufgabe der nächsthöheren Schicht.

IP Adressen (32 Bit * 2)

IPv4 benutzt 32-Bit Adressen zur Identifizierung des Hosts. Im IP-Header befindet sich jeweils eine Angabe zum Sender (Source IP-Adress) und zum Empfänger (Destination IP-Adress). Diese Adressen sind für Router transparent, d.h. Router können sich die Adressinformation beschaffen und anhand der Information die richtige Route finden.

Options

An den Header können optional noch verschiedene Optionen übergeben werden. Diese behandeln meistens die Themen Sicherheit und Routing.

Fragmentierung

Fragmentierung

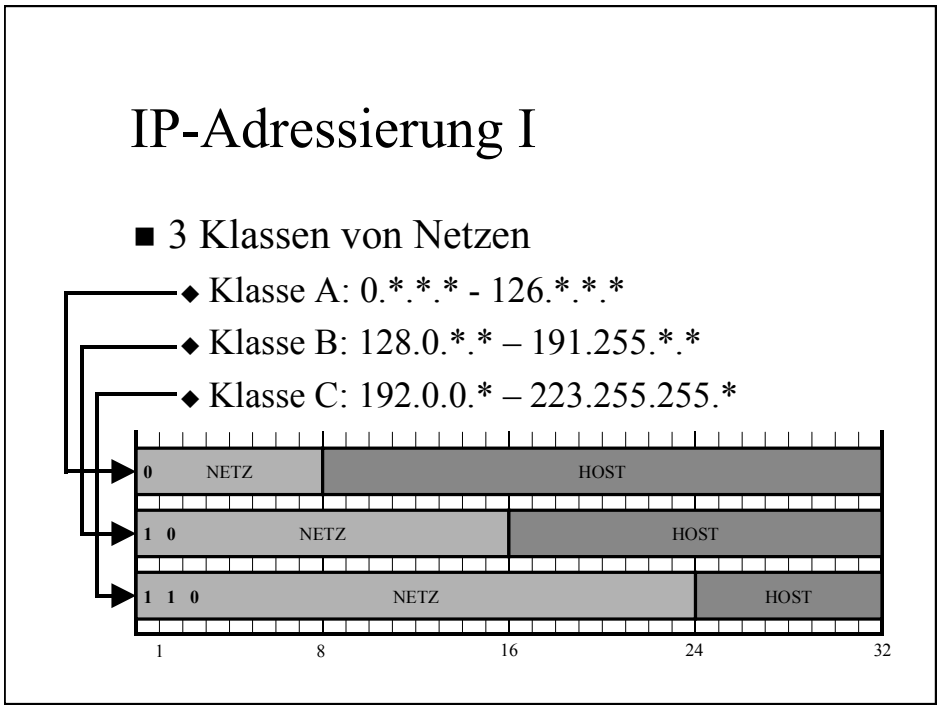
- Manchmal ist IP-Datagramm größer als Übertragungsmedium zulässt
  - IP Datagramm wird in Fragmente aufgeteilt
    - Router können Fragmente generieren
    - Router fügen Fragmente nicht mehr zusammen!
  - Zielrechner fügt Fragmente wieder zusammen
  - Fragmente haben im Feld ‚Identification' gleichen Wert
  - Fragmente geben im Feld ‚Fragment Offset' die Lage des vorliegenden Teils im gesamten Datagramm an.

Ein IP-Datagramm kann maximal 64 KByte groß werden. Das eventuell darunter liegende Ethernet-Protokoll lässt aber nur Frames mit maximal 1514 Byte zu. Überschreitet die Gesamtlänge des IP-Datagramms die maximal erlaubte Länge der darunter liegenden Schicht, so wird das IP-Datagramm mehrere Fragmente zerlegt. Es obliegt dem Zielrechner, die einzelnen Fragmente einzusammeln und wieder zu einem ganzen Datagramm zusammenzustöpseln. Erst wenn alle Teile des Fragmentes eingetroffen sind, wird das Datagramm an die nächst höhere Schicht weitergeleitet.

ⓘ Eine Fragmentierung kann auch bei einem Router geschehen, der Pakete empfängt und nun auf eine Leitung legt, die nur eine kleinere, maximale Paketlänge erlaubt. Router können jedoch Fragmente nicht mehr zusammenfügen, da es ja möglich ist, dass Teile des Fragmentes über alternative Routen geschickt worden sind und den Router niemals passieren werden.

Die einzelnen Fragmente haben im Feld ‚Identification' des IP-Headers den gleichen Wert stehen. Damit kann das Fragment eindeutig bestimmt und zugeordnet werden. Der Wert im Feld ‚Fragment Offset' bestimmt dabei die Lage des Fragmentes innerhalb des gesamten Datagramms. Anhand dieser Informationen kann der Zielrechner das Datagramm wieder korrekt zusammenfügen und an die nächsthöhere Schicht weitergeben.

Adressierung in IP-Netzwerken



Jeder Host oder Router wird in einem IP-Netzwerk anhand einer eindeutigen, numerischen IP-Adresse identifiziert. Diese Adressen sind in der heutzutage gebräuchlichen IP-Version (IPv4) 32 Bit (4 Byte) lang. Die Adressierung in IP und den darüber liegenden Schichten unabhängig von den Adressen der verwendeten Schicht 2 (MAC-Adressen).

ⓘ Router verbinden zwei unterschiedliche Netze und sind deswegen an beide Netzwerke angeschlossen. Falls es sich bei beiden Netzwerken um IP-Netze handelt, so besitzt ein solcher Router zwei unterschiedliche IP-Adressen – für jedes Netz eine.

Meist werden die einzelnen Bytes (also jeweils 8 Bit der Adresse) dezimal mit einem Punkt als Trennzeichen notiert (z.B. `10.17.1.68`). Dabei besteht die Adresse aus einem Teil der das Netzwerk spezifiziert und einem weiteren Teil, der den Rechner innerhalb des Netzwerkes identifiziert. Rechner innerhalb eines Netzwerkes führen den gleichen Wert im Netzwerkteil.

Routing

Das Routing basiert auf der Konvention, dass Rechner innerhalb eines Netzwerkes den gleichen Wert im Netzwerkteil der IP-Adresse führen. Jedes Netzwerk hat dabei einen unterschiedlichen Wert im Netzwerkteil. Wenn der Router ein IP-Paket empfängt, dann untersucht er nur den Netzwerkteil dieses Paketes. Anhand dieser Information kann der Router nun bestimmen, ob das Paket für eines der Netze, an denen er angeschlossen ist, bestimmt ist – und das Paket in das richtige Netz weiterleiten. Falls das Paket für keines der Netze bestimmt ist, bestimmt der Router anhand seiner Landkarte, an welches Netz das Paket übermittelt wird.

Klassifizierung von Netzen anhand IP-Adressen

Um Netzwerke unterschiedlicher Größe realisieren zu können, ist die Grenze zwischen Netzwerknummer und Hostnummer in den Adressen innerhalb gewisser Grenzen variabel. Im folgenden werden die Klassen von Netzwerken näher besprochen. Prinzipiell basiert die Klassifizierung darauf, dass geschaut wird, an welcher (binären) Stelle in der IP-Adresse das erste Mal die ‚0' vorkommt.

Klasse A-Netzwerke

Klasse A-Netzwerke zeichnen sich dadurch aus, dass die ‚0' gleich im ersten Bit der 32-Bit langen Adresse steht. Die folgenden 7 Bit dienen der Kennung für das Netzwerk. Somit realisiert nur das erste Byte den Netzwerkanteil der IP-Adresse, die restlichen 3 Bytes identifizieren die einzelnen Rechner des Netzwerkes (Hostanteil). Da nur 7 Bit für den Netzwerkanteil zur Verfügung stehen, können leicht die gültigen Netzwerknummern errechnet werden. Diese umfassen (dezimal gesehen) ‚0' bis ‚126', da der mögliche Wert ‚127' eine Spezialadresse ist.

Der Wert ‚127' im ersten Byte der IP-Adresse adressiert das lokale Netzwerk – unabhängig von der aktuellen Netzwerkklasse.

Klasse B-Netzwerke

Klasse B-Netzwerke zeichnen sich dadurch aus, dass die ‚0' im zweiten Bit der Adresse steht, somit muss im ersten Bit eine ‚1' stehen (also beginnt eine Klasse B-Netzwerkadresse mit ‚10'). Die folgenden 6 Bit und das nächste Byte (also die nächsten 14 Bit) bestimmen das Netzwerk. Da bereits 16 Bits für die Netzwerkadresse aufgebraucht wurden, bleiben noch 16 weitere Bits für die Hostadresse. Die möglichen Netzwerkadressenwerte stehen also in den ersten zwei Bytes der IP-Adresse und umfassen somit (dezimal gesehen) die Werte ‚128.0' bis ‚191.255'.

Klasse C-Netzwerke

Klasse C-Netzwerke zeichnen sich dadurch aus, dass die ‚0' im dritten Bit der Adresse steht, somit müssen die ersten beiden Bits eine ‚1' enthalten (also beginnt eine Klasse C-Netzwerkadresse mit ‚110'). Die folgenden 5 Bits und die nächsten beiden Bytes (also die nächsten 21 Bits) bestimmen das Netzwerk. Das verbleibende letzte Byte realisiert die Hostadresse. Es sind somit in einem Klasse C-Netzwerk theoretisch nur 256 verschiedene Rechner ansprechbar. Jedoch sind wiederum zwei spezielle Werte im letzten Byte für Spezialzwecke reserviert. Die möglichen Netzwerkadresswerte stehen also in den ersten drei Bytes der IP-Adresse und umfassen somit (dezimal gesehen) die Werte ‚192.0.0' bis ‚233.255.255'.

In der Theorie werden zusätzlich noch Klasse D und Klasse E-Netzwerke für Multicasts und künftige Nutzungen erwähnt. Diese sind jedoch für den täglichen Gebrauch kaum von Interesse und werden an dieser Stelle nicht näher erläutert.

Bei Netzwerken der Klasse A, B und C sind im Hostteil der Adresse zwei spezielle Werte reserviert:

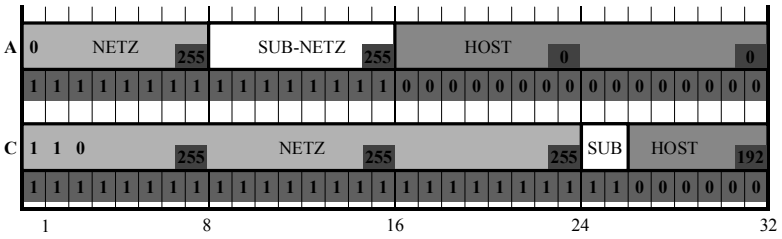
- Eine Hostadresse mit dem Wert ‚0' bezieht sich auf das Netzwerk und nicht auf einen Rechner.

- Eine Adresse in der alle Bits der Hostadresse auf ‚1' stehen, bezieht sich auf alle Rechner des aktuellen Netzwerkes (Broadcasts).

Diese Konventionen schränken die Anzahl der möglichen Rechner in einem Netzwerk zusätzlich ein, somit kann ein Netzwerk der Klasse C nur mehr 254 Rechner adressieren.

IP-Adressierung II

- Subnetting erlaubt Aufteilung eines Netzes in mehrere Subnetze (intern)



- Anzahl IP-Adressen beschränkt → IPv6 soll dieses Problem lösen

Subnetting

In manchen Fällen ist eine weitere Unterteilung eines großen oder mittleren Netzwerkes in mehrere kleine Netzwerke sinnvoll. Anhand von Subnetting kann die Grenze zwischen Netzwerkadresse und Hostadresse um beliebige Bits nach rechts verschoben werden (sprich der Netzwerkanteil kann vergrößert werden). Nach außen ändert sich nichts, die Router betrachten ja nur die eigentliche IP-Adresse. Lediglich die lokalen Router – also diejenigen die eine Verbindung zur Außenwelt herstellen – müssen ihre Entscheidungen anhand der geänderten Grenze treffen, d.h. darüber Bescheid wissen.

Die Grenzverschiebung erfolgt anhand der Angabe einer Subnet-Mask. Diese Maske ist – genauso wie die eigentliche IP-Adresse – 32 Bit lang. Steht in einem Bit eine logische ‚1', so wird das korrespondierende Bit in der IP-Adresse als Netzwerkteil der IP-Adresse angesehen. Steht in einem Bit der Subnet-Mask eine logische ‚0', so wird das korrespondierende Bit in der IP-Adresse als Hostteil der IP-Adresse gewertet. Die Angabe erfolgt auch hier dezimal, die Bytes werden durch einen Punkt getrennt. Die Subnet-Mask ‚255.255.255.0' würde z.B. bedeuten, dass die ersten 3 Byte der IP-Adresse als Netzwerkadresse angesehen werden, das letzte Byte der Host-Anteil ist (korrespondierend zu einem Klasse C-Netzwerk).

Eine solche Aufteilung würde Sinn machen, wenn ein Unternehmen viele kleinere Netzwerke hat (z.B. jede Abteilung hat ihr eigenes Netzwerk). Da interne Router mit dieser Netzmaske arbeiten, kann so eine Entlastung des Netzwerkverkehrs erreicht werden. Es ist aber auch denkbar, dass ein Netzwerk einer höheren Klasse (z.B. Klasse B) in mehrere Netzwerke einer niedrigeren Klasse (z.B. Klasse C)

aufgeteilt werden kann. Somit kann der Mangel von IP-Adressen vorübergehen hinausgeschoben werden.

IPv6

Das Problem der ausgehenden IP-Adressen kann mit Hilfe von Subnetting zwar hinausgeschoben werden, allerdings muss langfristig gesehen eine bessere Lösung gefunden werden. Zudem bestehen neue Anforderungen. All dies führte zur Entwicklung einer neuen Version des Internet Protocols, dem IPv6. Das neue Protokoll sollte sich für Milliarden von Hosts eignen, die Routing-Tabellen möglichst kurz halten, eine bessere Sicherheit bieten, Echtzeitanforderungen unterstützten (z.B. für Video-on-Demand) und in der Übergangszeit mit dem bestehenden Protokoll kooperieren können.

Die wesentlichen Änderungen gegenüber IPv4 sind die folgenden:

- Die IP-Adressen sind von 32 Bit auf 128 Bit erhöht worden.
- Der Header wurde vereinfacht und enthält nur mehr 7 Felder. Weitere Felder können in einem Erweiterungs-Header angegeben werden.
- IPv6 erlaubt die Authentifizierung von Hosts und die Verschlüsselung des Datagramms.

Für genauere Informationen zu IPv6 wird an dieser Stelle auf einschlägige Fachliteratur verwiesen.

Host-2-Host Layer

Host-2-Host Layer

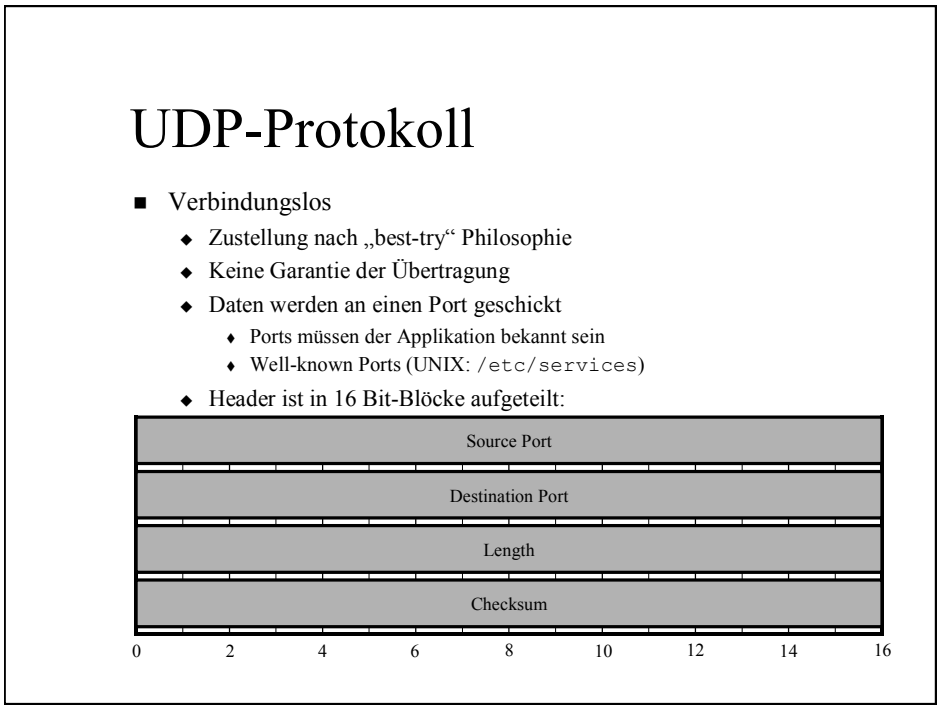
- Schicht 4 im ISO-OSI Modell
- Annahme: Existierende Verbindung (IP-Adressen von Internet-Layer)
- Verbindung zwischen Applikationen (Ports)
- Zwei Protokolle
  - User Datagram Protocol (UDP)
    - Verbindungslos, best-try (unsicher, schnell)
  - Transmission Control Protocol (TCP)
    - Garantiert Reihenfolge und Versenden → langsamer

Der Host to Host Layer verbindet Applikationen und geht dabei von einer bereits existierenden Verbindung (auf Basis des Internet Layer) der beteiligten Stationen aus. Der Host to Host Layer entspricht dabei der Schicht 4 des ISO-OSI Referenzmodells.

Bei IP werden die Stationen durch die IP-Adressen identifiziert. Nun kann es sein, dass zwei Rechner verschiedene Anwendungsdaten gleichzeitig austauschen (z.B. E-mail und Web). Deswegen muss der Host to Host Layer zusätzlich zur Unterscheidung der Herkunft (auf Basis von IP) noch die Zielanwendungen unterschieden. Dies geschieht anhand der Angabe von sogenannten Ports. Ein Port kennzeichnet die Applikation innerhalb der Rechner. Die sendende Applikation muss die Adressdaten der empfangenden Station selbst kennen. Die Ermittlung der Ports ist nicht Aufgabe des Host to Host Layers und fällt sowohl bei UDP als auch bei TCP in den Aufgabenbereich der übergeordneten Applikation.

Dabei kann der Host to Host Layer zwischen zwei verschiedenen Protokollen wählen: dem *User Datagram Protocol (UDP)* und dem *Transmission Control Protocol (TCP)*. UDP übernimmt im Wesentlichen die Eigenschaften von IP, es arbeitet verbindungslos und die Übertragung wird nicht garantiert. TCP hingegen erweitert die Funktionalität von IP und garantiert die Zustellung der Nachrichten in der richtigen Reihenfolge. Auch eine Flusskontrolle ist vorhanden um zu verhindern, dass ein schneller Sender einen langsameren Empfänger unnötig überlastet. Zur Unterscheidung der beiden Protokolle wird das Feld *‚Protocol'* im IP-Header verwendet.

Das UDP-Protokoll



UPD übernimmt die Fähigkeiten von IP und zeichnet sich durch einen minimalen Overhead aus. Durch das Wegfallen der Sicherung und weiteren Kontrollen zeichnet sich dieses Protokoll auch für einen guten Datendurchsatz aus. UPD arbeitet verbindungslos und stellt keine Beziehung zwischen den einzelnen Datagrammen her. Somit ist nicht garantiert, dass die Pakete in der richtigen Reihenfolge am Zielrechner eintreffen, aber auch die vollständige Übertragung selbst kann nicht garantiert werden.

Wie schon zuvor erwähnt, werden Daten an einen Port geschickt. Dieser Port definiert die Zielapplikation – und muss deswegen den beteiligten Applikationen bekannt sein. Es gibt aber auch eine Reihe standardisierter Ports, die sogenannten ‚Well-Known Ports'. Diese sind in UNIX in der Datei ‚/etc/services' eingetragen.

Der Port wird neben anderen Daten im Header des Protokolls übertragen. Der Header ist in Blöcken zu jeweils 16 Bit aufgeteilt. Die einzelnen Felder werden nun kurz vorgestellt.

Source Port (16 Bit)

Der Wert in diesem Feld kennzeichnet die sendende Applikation. Falls keine Antwort auf eine Nachricht erwartet wird, kann die Angabe hier auch entfallen. Dann hat das Feld den Wert ‚0'.

Destination Port (16 Bit)

Der Wert in diesem Feld kennzeichnet die angesprochene Applikation. Dieses Feld muss auf jeden Fall einen Wert enthalten.

Length (16 Bit)

Diese Feld enthält die Anzahl der Bytes im gesamten UDP-Datagramm, also sowohl die Länge des Headers als auch die Länge des Informationsteils.

Checksum (16 Bit)

In diesem Feld befindet sich eine Prüfsumme über das gesamte UDP-Datagramm. Dies ist deswegen notwendig, da die Prüfsumme von IP nur den IP-Header abdeckt, jedoch nicht den übertragenen Informationsteil. Anhand dieses Wertes kann der Empfänger feststellen, ob das UDP-Datagramm fehlerfrei übertragen worden ist.

Das TCP-Protokoll

TCP-Protokoll

- Verbindungsorientiert
  - Einhalten der Reihenfolge garantiert
  - Garantierte Zustellung
  - Fehlerbehandlung bei inkorrekter Zustellung
  - Flusskontrolle verhindert Überlasten des Empfängers
  - Daten werden an ein Port geschickt
    - Port muss der Applikation bekannt sein

UPD wird im Normalfall immer dann verwendet, wenn Daten sehr schnell übertragen werden sollen und eine Quittierung und eine Fehlerbehandlung nicht notwendig sind. Dies trifft vor allen bei einmaligen und kurzen Nachrichten zu (z.B. bei Internet-Telefonie). In den meisten Fällen wird jedoch gerade auf solche Eigenschaften besonders Wert gelegt (z.B. Datenübertragung). Auch das Einhalten der richtigen Reihenfolge ist meist wichtig. Um diese Anforderungen kümmert sich das TCP-Protokoll.

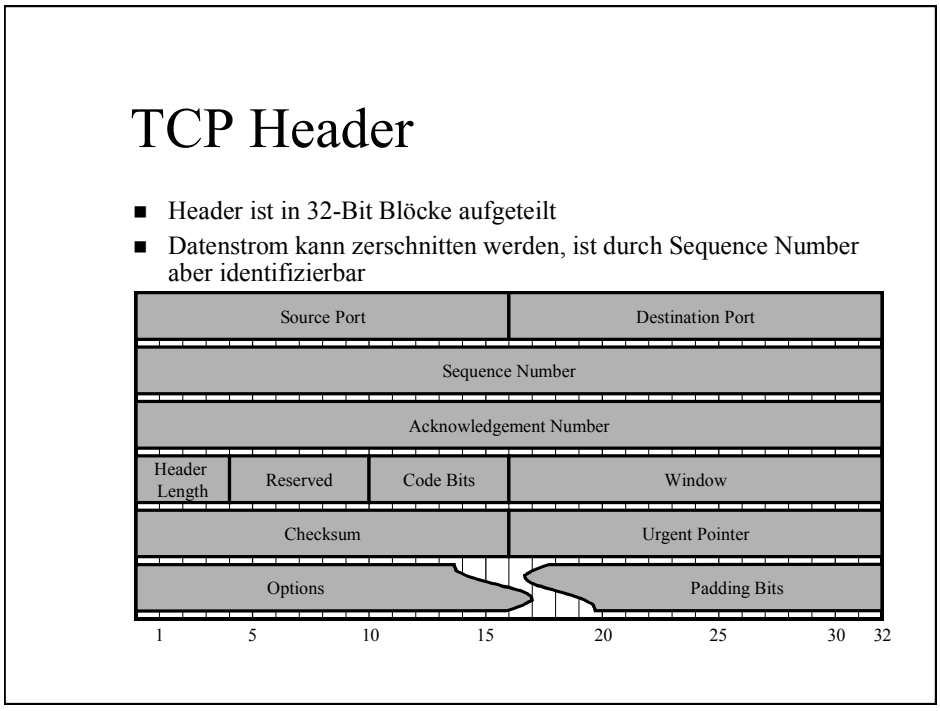
Eine TCP-Übertragung beginnt zuerst mit einem Verbindungsaufbau und endet mit einem Verbindungsabbau. Während die Verbindung steht, wird durch das TCP-Protokoll eine fehlerfreie Übertragung gewährleistet.

Die fehlerfreie Übertragung durch das TCP-Protokoll beinhaltet auch das Einhalten der Reihenfolge. Wie schon zuvor erwähnt, kann es passieren, dass die Pakete in verschiedener Reihenfolge beim Empfänger ankommen. TCP sorgt dafür, dass die Reihenfolge eingehalten wird und erst wenn alle Fragmente angekommen sind, wird das Paket an die nächsthöhere Schicht weitergeleitet.

Kommt es während der Übertragung zu einem Fehler, so sorgt TCP dafür, dass das fehlende oder unvollständige Paket neuerlich gesendet wird. Die Flusskontrolle verhindert dabei, dass der Empfänger eventuell durch zu viele eingehende Nachrichten überlastet wird.

Wie auch bei UDP werden bei TCP die Daten an einen Port geschickt, dass die Applikation identifiziert.

Der TCP-Header



Der TCP-Header ist in 32-Bit Blöcke aufgeteilt. Im folgenden werden die einzelnen Felder des TCP-Headers näher erläutert.

Source Port (16 Bit)

Der Wert in diesem Feld kennzeichnet die sendende Applikation. Anders als bei UDP ist hier jedoch eine Angabe zwingend erforderlich, da Kontrollinformationen wieder an diesen Port gesendet werden müssen.

Destination Port (16 Bit)

Der Wert in diesem Feld kennzeichnet die angesprochene Applikation. Wie auch schon bei UDP muss dieses Feld unbedingt einen Wert enthalten.

Sequence Number (32 Bit)

Der Wert innerhalb dieses Feldes identifiziert jedes Paket eindeutig. TCP betrachtet alle Daten als einen zusammengehörigen Strom, der durch einzelne Übertragungen in Teile zerschnitten wird. Die Sequence Number gibt die Position des ersten Bytes der vorliegenden Nachricht innerhalb des Datenstroms an, d.h. die genaue Position in der Gesamtübertragung. Beim Verbindungsaufbau wählt jede Station ihre Sequence Number nach eigenem Ermessen, bei einem Überlauf beginnt die Sequence Number wieder bei ‚0'.

Acknowledgement Number (32 Bit)

Dieses Feld dient zur Bestätigung von zuvor eingelangten Paketen. Die hier sendende Station bestätigt den Erhalt aller zuvor von ihr empfangenen Bytes bis hin zum Byte mit der hier angegeben Nummer – 1 (*Acknoledgement Number – 1).*

Das nächste erwartete Byte hat damit den Offset *Acknowledgement Number*. Dadurch, dass für diese Bestätigung ein eigenes Feld benutzt wird, kann die Station zeitgleich mit dem Bestätigen wieder selbst Daten senden.

Header Length (4 Bit)

Der Wert dieses Feldes gibt die Länge des Headers in 32-Bit Worten an. Wegen der variablen Optionen am Schluss kann die Länge des Headers variieren.

Reserved (6 Bit)

Dieses Feld ist für künftige Erweiterungen vorgesehen und wird nicht benutzt.

Code Bits (6 Bit)

Dieses Feld enthält einige Flags zur Beeinflussung der Übertragungseigenschaften:

- Urgent Pointer Field Valid (1 Bit): Ist das Bit gesetzt, enthält die Nachricht wichtige Daten. Die Übertragung erfolgt unter Umgehung der Flusskontrolle und wird sofort der Applikation zugestellt.

- Acknowledgement Field Valid (1 Bit): Ist dieses Bit gesetzt, so bestätigt der Sender den Empfang von Daten aus der Gegenrichtung.

- Push (1 Bit): Dieses Bit bewirkt die sofortige Weiterleitung des Pakets an die Applikation.

- Reset (1 Bit): Dieses Bit kennzeichnet die Beendigung der Verbindung ohne expliziten Verbindungsabbau.

- Sequence Number (1 Bit): Ist dieses Bit gesetzt, so enthält die Nachricht eine Initial Sequence Number. Ausgehend von dem Wert der im Feld Sequence Number angegeben wird, erfolgt nun die Übertragung. Diese Option wird normalerweise nur beim Verbindungsaufbau benötigt, um der Gegenstelle die gewählte Sequence Number mitzuteilen.

- End of Byte Stream From Sender (1 Bit): Dieses Bit kennzeichnet die Beendigung der Verbindung.

Window (16 Bit)

Dieses Feld dient der Flusskontrolle und enthält die Anzahl der Bytes, die der Empfänger aufnehmen kann. Ist die Differenz zwischen Acknowledge Number und Sequence Number größer als der im Feld Window angegebe Wert, so ist es dem Sender nicht mehr erlaubt, Daten an den Empfänger zu senden. Erst wenn die Differenz wieder den im Feld Window angegebenen Wert unterschreitet, darf die Sendung fortgesetzt werden.

Checksum (16 Bit)

Dieses Feld enthält – ähnlich dem entsprechenden Feld im UDP-Paket – eine Prüfsumme über das gesamte TCP-Paket. Anhand der Checksumme kann der Empfänger die fehlerfreie Übertragung des Paketes überprüfen.

Urgent Pointer (16 Bit)

Dieses Feld gibt die Anzahl der dringenden Daten im Paket an, falls das *urgent pointer field valid*-Flag gesetzt ist.

Options

Dieses Feld nimmt weitere Optionen auf, wie z.B. das Verabreden auf eine maximale Segmentgröße zwischen Sender und Empfänger.

Padding Bits

Der Header besteht aus einer Reihe von 32-Bit Worten. Da die Optionen keine bestimmte Länge haben, muss der Header gegebenenfalls auf ganze 32-Bit Worte aufgefüllt werden. Dies passiert mittels der Padding Bits.

DNS (Domain Name Service)

Domain Name Service I

- Jeder Rechner eindeutige IP-Adresse
  - Kompakt zu Speicher
  - Schwer zu merken
- DNS: Zuordnung Name ←→ IP-Adresse
- Zuordnung lokal (Datei `/etc/hosts`)
  - Praktikabel für kleine Netzwerke
- Verteilte Datenbank
  - Unterteilung in Top-Level Domains und Sub-Domains
  - Jede Domain hat eigenen Name Server

Bei Verwendung des TCP/IP-Protokolls wird jeder im Netz integrierter Netzwerkkarte eine innerhalb des Netzes eindeutige IP-Adresse zugewiesen. Diese Adresse identifiziert die Netzwerkkarte und somit den Rechner eindeutig. Darüber hinaus ist die Adresse die Basis für das Routing. Solche IP-Adressen sind kompakt zu speichern, sie sind jedoch für den Menschen schwer zu merken und wenig aussagekräftig.

Die *Domain Name Services* weisen jeder IP-Adresse einen sprechenden Namen zu. Gerade im WWW wird dieses Service sehr stark genutzt, da es Menschen unzumutbar wäre, sich die IP-Adressen der Server zu merken, auf denen Web-Inhalte abgespeichert werden. So ist z.B. zum Zeitpunkt der Niederschrift dieses Skriptum die Seite „www.spps-consulting.de“ mit der IP-Adresse ‚192.67.198.52’ verbunden. Obwohl die Seite in einem Webbrowser sowohl unter Angabe der IP-Adresse angezeigt werden würde, wird dies nur in den seltensten Fällen geschehen.

DNS bietet aber noch einen weiteren Vorteil. IP-Adressen sind aufgrund der Teilung in Netzwerkanteil und Hostanteil starr mit dem LAN verbunden. DNS hebt diese Bindung auf, da einem DNS-Namen ohne Probleme eine andere IP-Adresse zugewiesen werden kann. Somit ist es möglich, einen bekannten Rechner in ein anderes LAN zu transferieren und trotzdem immer noch mit dem gleichen Namen anzusprechen.

Speicherung der Adressenzuordnung

Ursprünglich wurde die Adressenzuordnung in jedem Rechner lokal gespeichert. Auf einem UNIX-System werden diese Daten in der Datei ‚/etc/hosts’ abgelegt. Diese Methode ist zwar für kleine, private Netzwerke durchaus praktikabel, in größeren Netzwerken ist der administrative Aufwand hierfür jedoch zu groß.

Darum gibt es auch noch sogenannte verteilte Datenbanken. Im folgenden werden beide Alternativen näher beschrieben.

Lokale Datenbank

Hier wird die Zuordnung in eine lokale Datei gespeichert. Im folgenden ist ein Auszug einer solchen Datei abgebildet.

```
# IP-Address   Hostname                        Additional Names
127.0.0.1      localhost loopback
10.17.1.67     SSW-1.ssw-network.com           SSW-1
10.17.1.68     SSW-Solaris.ssw-network.com     SSW-Solaris
10.17.1.70     SSW-DualServ.ssw-network.com    SSW-Dualserv
```

Abbildung 2-8: Auszug aus der Datei /etc/hosts

In dieser Datei wird zu der IP-Adresse zumindest der offizielle Name gespeichert. Weitere, alternative Namen (Aliases) können getrennt durch ein Leerzeichen optional angegeben werden. Auch im Internet wurde in seiner Anfangsphase die Namenszuordnung ausschließlich mit lokalen Konfigurationsdateien realisiert. Offizielle Rechnernamen vergab das NIC (Network Information Center). Es hielt eine offizielle Version der Datei ‚/etc/hosts' bereit, die laufend an alle Teilnehmer verteilt wurde.

Die durch die wachsende Anzahl von Rechnern entstehenden Nachteile sind klar:

- Kollisionen von Namen
- Hoher Verwaltungsaufwand in der Zentrale
- Konsistenz durch viele Änderungen ist nur schwer zu gewährleisten
- Vermehrter Datenverkehr am Netz durch große Datenbank und viele Hosts

Diese Nachteile und das ständig wachsende Internet zogen eine Weiterentwicklung – die sogenannte Domain Name Services (kurz DNS) – mit sich. Hier werden die Daten in global verteilte Datenbanken abgelegt.

Verteilte Datenbanken

Bei dieser Methode werden Namenszuordnungen in hierarchisch organisierten Datenbanken – den eigentlichen Domain Name Services (DNS) – abgelegt.

In DNS sind Rechnernamen hierarchisch aufgebaut. Statt eines einzigen Namens besteht ein Name aus einem Teil der den Rechner innerhalb seines logischen Umfelds identifiziert und einen Domänen-Anteil, der hierarchisch aufgebaut ist. Im Internet vergibt das NIC lediglich die Hauptdomäne (top-level domains) und tritt die Verwaltung für Sub-Domänen an lokale Verantwortliche ab. Beispiele für solche Hauptdomänen sind ‚.com', ‚.org', ‚.edu', ‚.at', ‚.de' etc. Lokale Verantwortliche können wiederum weitere Domänen- und Rechnernamen innerhalb der eigenen Domäne vergeben. Dadurch entstehen die heute verwendeten hierarchisch organisierten Domänennamen. Sie werden von hinten

nach vorne gelesen. Ganz hinten steht die Hauptdomäne, ganz vorne der Rechnername. Die einzelnen Bereiche (Hierarchien) werden durch Punkte voneinander getrennt.

Jeder Betreiber einer Domäne ist für die unter ihm befindlichen Subdomänen verantwortlich. Nach außen vertritt er den ganzen unter seiner Obhut befindlichen Teilbaum. Bei größeren Domänen delegiert er die Zuständigkeit für die Subdomänen an die darin befindlichen Institutionen. Jeder Verantwortliche verwaltet seinen Teil mit einem eigenen lokalen Name Server. Ein Name Server kennt die Rechner, die direkt in der eigenen Domäne angesiedelt sind. Bei Subdomänen kennt der Name-Server der übergeordneten Domäne entweder den Name Server der Subdomäne, oder aber alle Rechner der Subdomäne.

Dieses Schema erfordert jedoch, dass im ganzen Internet einige Name Server existieren, die auf der obersten Hierarchieebene angesiedelt sind. Diese Server sind die sogenannten Root Name Server und sind rund um den Globus verstreut. Ihr Datenbestand ist ident. Jeder Name Server kennt einen oder mehrere solcher Root Server.

Trotz der Möglichkeiten von DNS werden lokale Datenbanken heute immer noch häufig als ‚Backup' verwendet. Falls ein DNS-Server ausfällt, kann immer noch mit dem Netz gearbeitet werden. Außerdem liefern lokale Datenbanken auf Anfragen viel schneller eine Antwort als wenn erst ein DNS-Server gefragt werden muss.

Domain Name Service II

- Auf Unix Systemen resolver
  - Konfiguration `/etc/resolv.config`
  - Wird z.B. von gethostbyname() verwendet
- Workstation fragt lokalen Name Server
  - Kennt Rechner → liefert Antwort
  - Kennt Rechner nicht → liefert Adresse eines Root Servers, dieser kennt dann Adresse (über mehrere Name Server)

Resolver

Gewöhnliche Rechner nehmen die Dienste lokaler Name Server in Anspruch. Die in Workstations dafür vorhandene Komponente heißt Resolver. Der Resolver ist sozusagen der Client für den DNS-Server und benutzt diesen zur Auflösung von DNS-Anfragen. Im Prinzip enthält der Resolver nichts anderes als die IP-Adresse des lokalen DNS-Servers und eventuell noch den Namen der Domäne in der er sich befindet. In UNIX wird der Resolver in der Datei ‚/etc/resolv.config' konfiguriert.

```
domain          ssw-network.com
nameserver      10.17.1.70
```

Abbildung 2-9: Auszug aus der Datei /etc/resolv.config

Die Nennung mehrerer Name Server ist natürlich möglich, dies steigert die Fehlertoleranz bei Ausfall einer der angegebenen Name Server. Abhängig von der Konfiguration des Resolvers verwendet er zum Ermitteln von IP-Adressen aus Namen entweder die lokale Hosts-Datei, den lokalen Name Server oder beides. Wird eine Anfrage an den Resolver gestellt, setzt dieser ein ‚dns_query' an den lokalen Name Server ab. In dieser Query ist der Rechnername angegeben. Der Name Server liefert dann ein ‚dns_reply' mit der korrespondierenden IP-Nummer an den Resolver zurück.

Auflösung von DNS-Anfragen

Ein Name Server hält den Datenbestand über die lokal in seiner Domäne existierenden Rechner. Er muss seinen Clients jedoch über alle Rechner des gesamten Netzwerkes Auskunft geben können. Auch wenn er selbst diese Rechner

nicht kennt, so kennt er zumindest andere Name Server und kann die Anfrage an diese delegieren.

Bei der Auflösung einer DNS-Anfrage schaut der DNS-Server zuerst, ob er nicht selbst diesen Rechner kennt. Ist das der Fall, so liefert er sofort die korrespondierende IP-Adresse zurück. Kennt er den Rechner jedoch nicht, so leitet er die Anfrage an einen Root Server weiter. Dieser sendet entweder direkt die Antwort oder teilt dem lokalen Name Server die IP-Adresse eines untergelagerten Name Servers mit, der für die gewünschte Subdomäne zuständig ist. So wird die Anfrage von DNS-Server zu DNS-Server weitergeleitet, bis einer die Anfrage beantworten kann. Das Motto bei diesem Verfahren ist ‚Ich weiß es selber nicht, aber ich kenne jemanden, der es wissen sollte!'. Die schließlich erhaltene Antwort leitet dann der lokale Name Server an den anfragenden Client zurück. Außerdem merkt sich der lokale Name Server die Adresse um künftige Anfragen schneller beantworten zu können. Mit dieser Methode werden andere Name Server (besonders die Root Name Server) erheblich entlastet.

Ein Sonderfall im Namensbaum des Internet stellt die Domäne ‚in_addr.arpa' dar. Sie enthält die inverse Zuordnung, also die Abbildung von IP-Adressen zu Rechnernamen. Mit dieser Domäne kann ein beliebiger Applikationsserver den Domänennamen eines Rechners aus der IP-Adresse erfahren. Dies wird z.B. zum Prüfen von IP-Adressen auf Authentizität verwendet. Jeder Name Server muß auch auf solche Anfragen antworten können.