Computer Networks

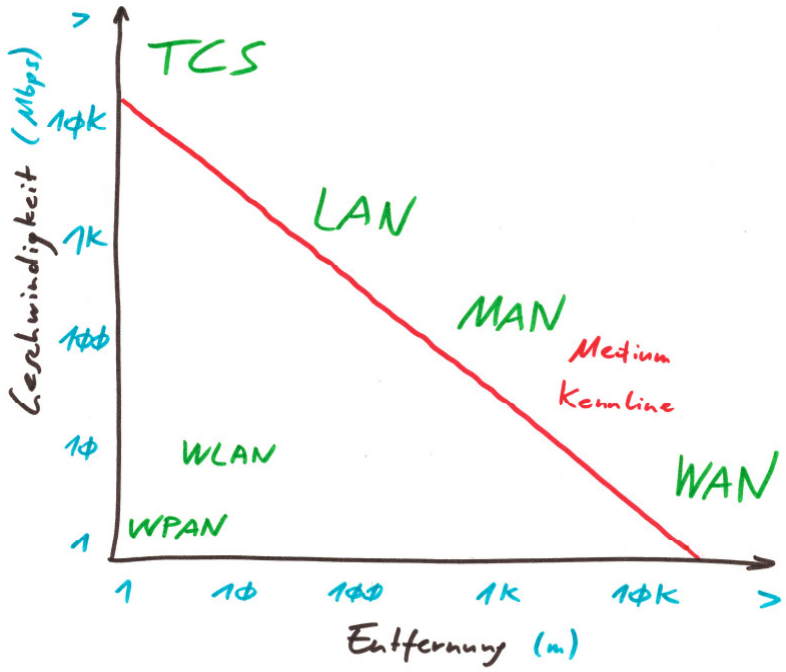
Christian Demuth

Ausgabe 2005

Abgrenzung LANs / MANs / WANs

Die Unterscheidung zwischen den “Lokalen Netzen” (“Local Area Networks”), den “Stadtweiten Netzen” (“Metropolitan Area Networks”) und den "Weitverkehrsnetzen" (“Wide Area Networks”) erfolgt gemeinhin anhand der Kriterien Übertragungsgeschwindigkeit (auch Übertragungsrate oder Datenrate genannt) und Übertragungsstrecke (maximal überbrückbare Entfernung).

Die sogenannte Medium-Kennlinie ist das Produkt der beiden Kriterien. Sie wird generell als “Bandbreite” bezeichnet. Ihre Einheit ist “km * MHz”. Damit kann man bei gegebenem Medium zwischen großer Entfernung oder großer Datenübertragungsrate wählen.



Skizze Medium-Kennlinie, Geschwindigkeit der Übertragung, Übertragungsgsdistanz

Bei den Lokalen Netzen geht man heute von einer Datenrate von ca 10 Mbps (“Megabit pro Sekunde”) bis ca 1 Gbps (“Gigabit pro Sekunde”) aus. In der Telekommunikation wird immer in Bits pro Sekunde gemessen, nie in Bytes pro Sekunde. Die Abkürzung “KB” ist in unserem Zusammenhang also falsch, richtig muß es immer “Kbps” oder "Mbps" oder "Gbps" heißen. Die Entfernungen, die heutige LANs überbrücken können, betragen bis zu einigen Kilometern.

Ferner sind LANs immer im “Privatbesitz”. Wenn also ein Teilstück eines LANs über öffentlichen Grund und Boden führt, dann ist das Netz eigentlich kein LAN mehr, sondern ein sogenanntes “Internet” (nicht zu verwechseln mit *dem* Internet, das aber seinen Namen auch zu Recht trägt, da es ebenfalls eine Menge von Subnetzen ist, die miteinander verbunden wurden). LANs sind damit immer organisationsweite und einer Organisation gehördende private Netze.

Außerdem charakterisiert das gemeinsame Übertragungsmedium die LANs. Da alle Stationen immer direkt miteinander verbunden sind, ergibt sich daraus der Vorteil, daß alle Stationen auch alle Datenpakete empfangen können. Die Realisierung eines Broadcasts (das ist das Senden an alle Stationen eines Netz-Segments) ist daher einfach möglich. Der Nachteil des gemeinsamen Mediums ist die Notwendigkeit einer Zugriffskontrolle auf selbiges, das sogenannte “Medium Access Control” Protokoll (kurz “MAC”). Der MAC steuert, wann welche Station senden darf, da durch das gemeinsame Medium zu einem Zeitpunkt immer nur eine einzige Station senden kann, ohne

daß das es zu Problemen kommt. In den neusten LANs ist dieses "Shared Medium" aus Effizienzgründen nicht mehr gegeben, es wird heute zumeist „pro Paket eine eigene Verbindung geschaltet“ (genannt Switched LANs, im Gegensatz zu den Shared Medium LANs).

Eine der Ur-Definitionen für LANs stammt aus der Ethernet-Spezifikation:

> “… independent devices to communicate with each other directly using a dedicated medium over medium distances (1-5km) with medium speed (1-10 Mbps)…”

Diese historische Definition stimmt in ihrem Kern größtenteils immer noch, auch wenn sich die Übertragungsgeschwindigkeit heutzutage mehr den 10 Gbps nähert, sich also in der 30jährigen Geschichte der LANs um mehr als den Faktor 1000 gesteigert hat. Aufgrund des Einsatzes von LANs als firmenweite oder organisationsweite Netze hat sich die Entfernung der einzelnen Knoten zueinander nicht wesentlich vergrößert. Erst mit der in letzter Zeit aufkommenden Bestrebung, die Netzwerk-Infrastruktur zu vereinheitlichen, werden Lokale Netze immer größeren Ausmaßes geschaffen – und damit auch die technische Entwicklung in diese Richtung getrieben.

Die Begriffe „WLAN“ („Wireless LAN“) und „WPAN“ („Wireless Personal Area Network“) werden in diesem Skriptum ebenfalls kurz betrachtet, die „TCS“ („Tightly Coupled Systems“), das sind Rechnerverbunde mit zumeist proprietären Hochgeschwindigkeitsverbindungen, dagegen nicht.

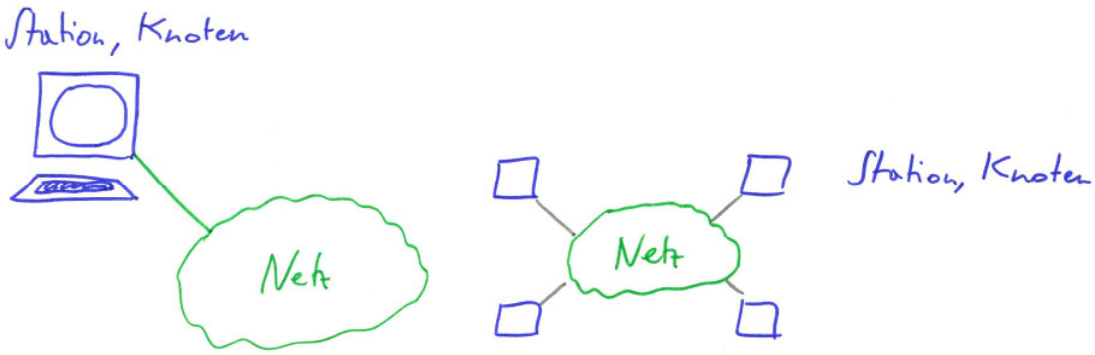
Geschichte der LANs

1960	Studien und Forschungsarbeiten zu Paktenetzen, die LAN-tauglich sind
1969	Das ARPANET stimuliert die Forschungsarbeiten für Subnetze, die sich später zum Internet verbinden werden
1973	Beginn der Idee eines universellen “Ethernet”. Vater des Ethernet ist Robert Metcalfe (Xerox). Einige Vorläufer/Prototypen waren bereits gebaut worden, Ethernet entstand aus den Verbesserungen, die in diesen Tests herausgefunden wurden.
1974	IBM definiert die “Systems Network Architecture”, “SNA”. SNA ist ein umfassendes hierachisches Netzwerksystem, das ursprünglich für Großrechner und “dumme” Terminals entwickelt wurde und später mit “peer-to-peer” Fähigkeiten erweitert wurde.
1976	Die CCITT definiert X.25 vulgo Datex-P, ein paketschaltendes WAN. Ethernet wird patentiert.
1979	Ethernet wird zum ersten Mal implementiert – eigentlicher Beginn der Geschichte der LANs
1980	Beginn des IEEE 802
1981	Token Ring wird erfunden
1984	Das OSI 7 Schichten Modell wird erfunden
1985	IEEE definiert die Normen 802.3 (aufbauend auf Ethernet) und 802.5 (aufbauend auf Token Ring)
1986	Erweiterung des 802.3 auf Telefonverkabelung
1990	FDDI
1992	DQDB
1993	ATM
1995	100 Mbps Ethernet
1998	1.000 Mbps Ethernet
2001	10.000 Mbps Ethernet

Häufig verwendete Begriffe

Knoten, Station und Netz

Wir bezeichnen ab hier die Rechnereinheiten, die in einem LAN mitwirken, als Knoten oder Stationen. Ein Netz ist in unserem Kontext immer ein LAN.



Skizze Station, Knoten und Netz

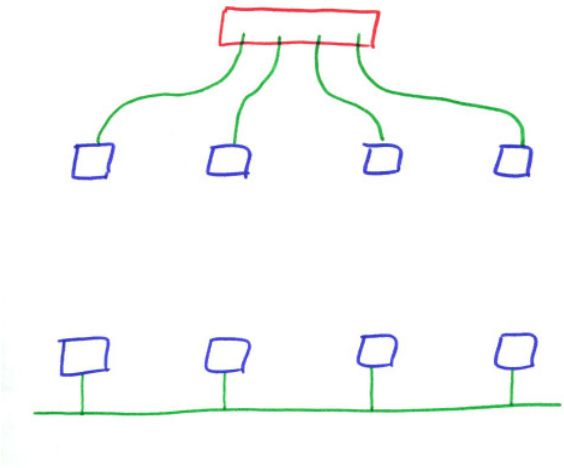
Subnetz

Im Protokoll TCP/IP wird von Subnetzen gesprochen. Hierbei handelt es sich um Netzwerke, die per Repeater oder Bridges zusammengefügt sind. Alle Knoten eines Subnetzes haben dieselbe IP-Netzadresse. Router im TCP/IP verbinden grundsätzlich Netze mit verschiedenen IP-Netzadressen. Die beiden Netze, die ein Router verbindet, sind damit automatisch Subnetze.

Der eigentliche Begriff Subnetz bezieht sich auf die künstliche Zerteilung eines Netzes in mehrere Subnetze mithilfe der sogenannten Subnetzmaske des IP.

Segment

Unter einem Segment versteht man “ein durchgehendes Stück Kabel”, an dem mehrere Stationen “hängen” können.



Mit der Koaxial-Verkabelung des Ethernet war es möglich, daß mehrere Stationen an einem Segment hingen. Da diese Verkabelungsart heute keine Rolle mehr spielt, kann man als “Segment” das Verbindungskabel zwischen einer Station und der nächsten Station bzw zwischen Station und Internetworking-Gerät bezeichnen.

Broadcast, Multicast und Unicast

Unter einem Unicast versteht man das gezielte Senden von Daten von der Sendestation an exakt eine adressierte Empfangsstation.

In Netzwerksystemen, bei denen mehrere Stationen das Paket unabsichtlich “zu sehen bekommen”, kann man auch Multicasts oder Broadcasts verwenden. Der Broadcast ist das Senden eines Datenpakets an alle (betriebsbereiten) Stationen eines Netzes. Er wird durch eine spezielle Adresse (meistens aus lauter binären “Einsen” bestehend) angezeigt. Ein Broadcast-Paket muß von jeder Station empfangen und bearbeitet werden, verbraucht daher in allen Stationen des Netzes CPU-Leistung.

Der Multicast ist das Senden an eine Gruppe von Stationen. Dabei wird der Netzwerkkarte eine oder mehrere (!) spezielle Multicast-Adresse(n) zugewiesen, unter der sich die Station angesprochen fühlt. Eine solche Station hat damit zumindest zwei zugewiesene Adressen, nämlich ihre Unicast-Adresse, unter der nur diese Station angesprochen wird, und die Multicast-Adresse. Die Multicast-Adresse kann von beliebig vielen anderen Stationen ebenfalls als Empfangsadresse zugewiesen werden. Damit wird der Overhead des Broadcasts umgangen und nur diejenigen Stationen, die an Multicast-Paketen interessiert sind, empfangen diese auch tatsächlich. Der Rest der Stationen bemerkt von diesen Paketen nichts.

Network Interface Card – NIC

Die NIC ist die Interface-Karte, die aus einem Rechner einen Knoten eines Netzwerks machen. An der NIC ist meist ein Kabel befestigt, das den Rechner in das Netz verbindet.

Die NIC gehorcht einem Schnittstellen-Standard für Rechner, wie zB ISA, EISA, PCI, PC-CARD (PCMCIA), etc. Es gibt aber auch Anbindungen von NICs an Rechner über andere NICs. ZB gibt es bereits USB-Ethernet-Karten. Hier wird die Ethernet-Karte über eine USB-Port an den Rechner gehängt. Dasselbe gilt auch für Fire Wire.

OSI 7 Schichten Modell

In den Achtzigerjahren wurde von der International Standards Organisation ein Referenzmodell für Telekommunikationssysteme geschaffen, das sogenannte “Open Systems Interconnect” oder kurz “OSI” Modell. Dieses unterteilt Telekommunikationssysteme intern in 7 Schichten:

Schichtnummer (“Layer”)	**Schichtname Englisch**	**Schichtname Deutsch**
L7	Application Layer	Anwendungsschicht
L6	Presentation Layer	Darstellungsschicht
L5	Session Layer	Sitzungsschicht
L4	Transport Layer	Transportschicht

L3	Network Layer	Netzwerkschicht
L2	Data Link Layer	Datensicherungsschicht
L1	Physical Layer	Physische Schicht
(L0)	Medium	Übertragungsmedium

Tabelle OSI “7 Schichten Modell”

Bei den LANs nach IEEE sind nur die untersten beiden Schichten des Modells implementiert. Wenn man sich netzwerkfähige Betriebssysteme ansieht, reicht die Implementierung bis in die Anwendungsschicht hinauf.

Adresse

Um eine Station in einem Netz anzusprechen, muß man sie “adressieren”. Dabei wird in dem Datenpaket, das man der Station zusenden möchte, die Zieladresse der Station in das Feld “Destination Address” (“DA”) geschrieben. Ferner wird die Adresse der absendenden Station in das Feld “Source Adress” (“SA”) geschrieben. Beides erfolgt im Treiber der NIC (unter Windows NT zB im NDIS) und ist für die Anwendung transparent.

Eine Station in einem Netz kann mehrere Adressen haben und damit auf mehrere Arten ansprechbar sein.

Adressen sind nur in den unteren Schichten (1 bis 3) wirklich relevant, man spricht daher von Schicht-1-Adressen, Schicht-2-Adressen und Schicht-3-Adressen. In den oberen Schichten werden oft andere Arten der Adressierung verwendet.

In der Schicht 1 und 2 ist die Erkennung der Adresse und das “Hinaufreichen” des Datenpakets an obere Schichten Aufgabe der NIC.

Port

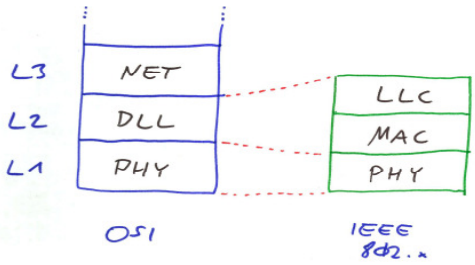
Unter dem Begriff Port werden zwei verschiedene Konzepte verstanden. Einerseits bezeichnet man die Ein/Ausgänge von Internetworking-Geräten (wie zB Repeatern, Bridges, Rouern, etc) als Ports. Andererseits gibt es den Begriff Port auch bei TCP und UDP. Dort wird allerdings mit einem Port eine Anwendung adressiert. Der Port im TCP/UDP ist also eine logische Adresse.

MAC

Der “Medium Access Control” ist die Steuerschicht der LANs, die den gemeinsamen Zugriff der Stationen auf ein gemeinsames Medium kontrolliert. Der MAC liegt in den OSI-Schichten 1 (oben) und 2 (unten)

LLC

Der “Logical Link Control” ist die Verbindungs-Schicht zwischen MAC und den Protokollen der OSI-Schicht 3. Der LLC liegt in der Schicht 2 oben.



Skizze MAC, LLC, OSI-7-Schichten

Frame

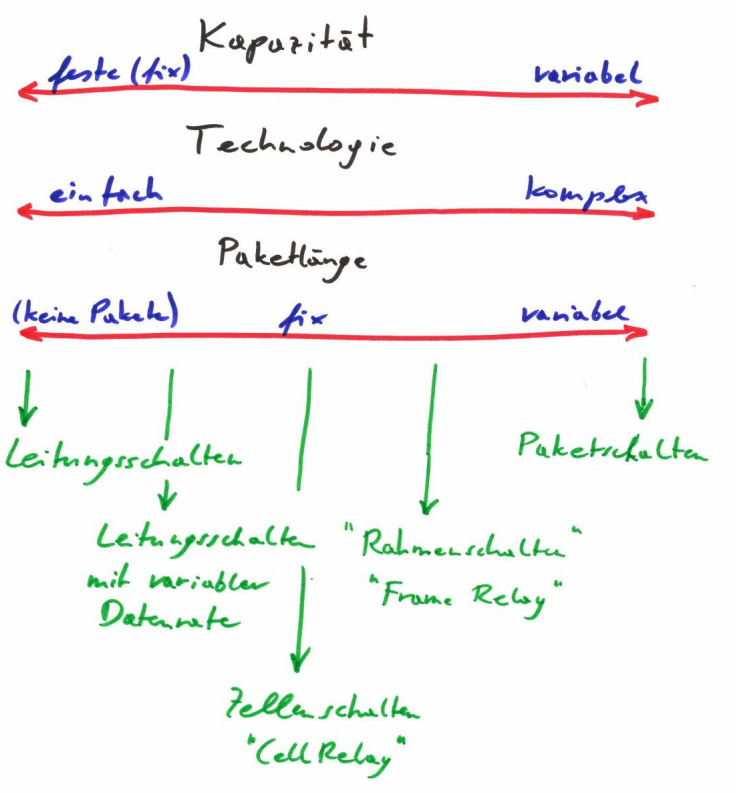
Als “Frame” bezeichnet man eine MAC-PDU, also ein Datenpaket der MAC-Schicht.

Internet

Wenn man unabhängige Netzwerke zusammenschließt, entsteht ein sogenanntes Internet. Charakteristisch hierfür ist die OSI-Schicht, in der der Zusammenschluß durchgeführt wird. Erfolgt das Zusammenfügen auf den Schichten 1 oder 2, so spricht man nach wie vor von einem LAN. Erfolgt der Zusammenschluß auf den höheren Schichten (zB per Router auf Schicht 3), so entsteht ein Internet.

Vermittlungsverfahren

Unter dem seltsam anmutenden Wort Vermittlungsverfahren versteht man die allgemeine Methode, wie Daten ihren Weg zum Ziel finden. Es hat nichts mit der tatsächlichen Wegfindung (dem Routing) zu tun, sondern beschreibt nur, wie die Daten zum Ziel kommen.



Leitungsschalten

Leitungsschalten ist das einfachste mögliche Konzept, es entspricht in etwa dem in der Telefonie gebräuchlichen Herstellen einer physischen Leitung. Leitungsschalten wird in der Datenübertragung heute kaum noch eingesetzt, nicht einmal mehr in der Telefonie gibt es eine physische Leitung zum Ziel. Bei dieser Methode sind Verbindungen zwingend vorhanden, es gibt nur einen Weg aller Daten zum Ziel. Die Bitrate (bps) ist praktisch immer fix eingestellt. Wenn eine Station nichts zu senden hat, wird die reservierte Bandbreite dieser Station verschwendet.

Die wichtigsten Vertreter des Leitungsschaltens im Datenbereich sind heute „SONET“ (Synchronous Optical Network“ und „SDH“ („Synchronous Digital Hierarchy“).

Cell Relay

Das Cell Relay (“Zellen-Schalten”) ist eine Weiterentwicklung aus der Technologie des Frame Relay und stammt ebenfalls ursprünglich aus dem Breitband-ISDN. Wieder werden Verbindungen aufgenommen, bevor Daten versendet werden können. Es gibt ebenfalls nur einen einzigen Weg aller Daten zum Ziel.

Cell Relay verwendet im Gegensatz zum Frame Relay kleine (53 Byte lange) Pakete (genannt “Cells”) mit wenig Overhead (nur 5 Byte Header). Größere Datenpakete müssen daher vorher auf 48 Byte Zellen segmentiert und nachher reassembliert werden.

Die Technologie geht von Verbindungen mit extrem geringer Fehlerwahrscheinlichkeit aus. Daher gibt es keine echten Prüfsummen mehr in den Cells und auch keinen Mechanismus zur Flußkontrolle.

Call Relay entwickelte sich, da die heutigen Medien (LWL) sehr geringe Fehlerraten aufweisen und speziell auch die Switch-Technologie weiter ausgereift ist. Die Bitrate des Cell Relay ist fix, die Datenübertragung isochron und daher echtzeitfähig.

Der wichtigste Vertreter heute ist ATM.

Frame Relay

Frame Relay entstand aus der Technologie des Packet Switching (Paketschaltens). Im Prinzip kann jedes X.25 System (in Deutschland und Österreich “Datex-P” genannt) auf Frame Relay “umprogrammiert” werden. Das zugrundeliegende Transportprotokoll des X.25, das sogenannte HDLC (“High Level Data Link Control”) bleibt ebenfalls vorhanden. Frame Relay wurde erfunden, um den im ISDN weithin ungenutzten D-Kanal (das ist der Signalisierungs-Kanal, also der Kanal, der die Steuerinformationen des ISDN parallel zu den B-Kanälen überträgträgt) zu verwerten.

Ein weiterer Motivator war und ist die Tatsache, daß Standleitungen wesentlich teurer sind und oft nicht voll ausgenutzt werden. Daher “versteckt” ein Frame Relay Betreiber seine Standleitungen in einem Frame Relay Netz und bietet seinen Kunden als Schnittstelle den Frame Relay Dienst an. Er kann über das Standleitungsnetz virtuelle Verbindungen für seine Kunden aufbauen und damit über einzelne Leitungen mehrere Verbindungen multiplexen. Damit wird die Auslastung der Leitungen verbessert und der Betreiber kann verschiedene Übertragungsraten anbieten, die evtl über ein und dieselbe physische Leitung laufen.

Dem Kunden steht damit eine Quasi-Standleitung mit hoher Datenrate zur Verfügung. Die

ursprüngliche Fähigkeit des X.25, eine beliebige andere X.25-Station explizit zu adressieren, ist im Frame Relay nicht mehr verfügbar.

Frame Relay wird auch heute noch oft als Verbindungsstrecke zwischen einzelnen LANs verwendet, um diese zB zu einem Firmen-Intranet zusammenzuschließen.

Datentransport in Frame Relay Netzen ist ausschließlich über Verbindungen. Es gibt nur einen Weg aller Pakete zum Ziel. Ein Frame Relay Netzwerk kennt kein Routing. Die zu verwendende Strecke wird im Vornherein festgelegt.

Die Länge der Pakete ist variabel und wird auf die Paketlänge der zu verbindenden Endnetze (zumeist LANs) angeglichen. Die aus den verbundenen LANs zum Transport ins Frame Relay weitergeleiteten Pakete werden in HDLC-Pakete verpackt und versendet. HDLC übernimmt dabei die Aufgabe der Fehlerkorrektur und der Flußkontrolle (“Congestion Control”).

Frame Relay reicht heute mit seiner Datenrate von 56Kbps bis ca 50 Mbps hinauf. Der Verzicht auf “Node-to-Node”-Flußkontrolle macht die höheren Leistungen möglich. Die Flußkontrolle des Frame Relay erfolgt in der Schicht-3 und damit direkt und ausschließlich zwischen Sendekonten und Empfangsknoten. Die einzelnen Teilstrecken sind nicht mehr mit Flußkontroll-Mechanismen versehen.

Die Bitrate des Frame Relay ist fest, die Verzögerungszeit dagegen nicht. Damit ist Frame Relay nicht für isochrone Datenübertragung geeignet.

Paketschalten

Das einzelne Versenden von Paketen ist - von der Implementierung her betrachtet - das komplexestes Konzept. Alle LANs arbeiten auf dieser Basis. In einem paketschaltenden Netz sind sowohl Verbindungen als auch Datagramme (und damit auch ein Broadcast und ein Multicast) möglich.

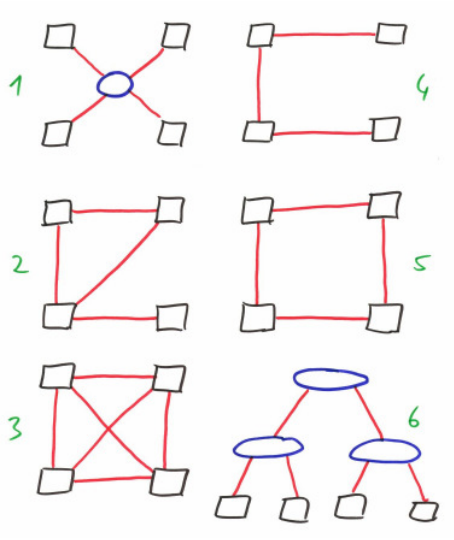
Sofern es die Topologie erlaubt sind unterschiedliche Wege der Datenpakete zum Ziel möglich. Welchen Weg ein Paket zum Ziel wählt, wird von Paket zu Paket einzeln bestimmt.

Die Bitrate ist grundsätzlich variabel, werden keine Daten gesendet, wird auch keine Netzkapazität benötigt. Damit entlastet eine Station, die nichts zu senden hat, das Netzwerk und stellt so anderen Stationen ihre Kapazität zur Verfügung.

Der wichtigste Vertreter heute ist Ethernet.

Physische Schicht - Topologien

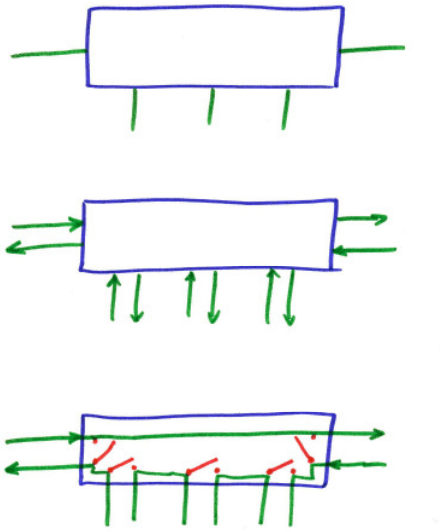
Unter Topologie versteht man hier die physische Anordnung der Stationen (oft auch Knoten genannt) in einem Netzwerk. Bei den Lokalen Netzen werden die folgenden Topologien eingesetzt:



Skizze Topologien

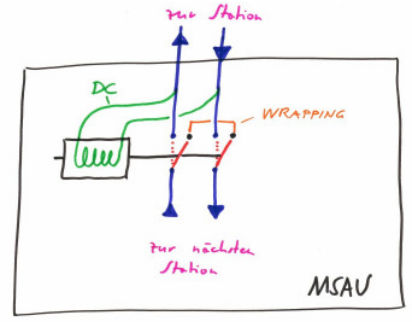
Historisch sind LANs aus einfachen Netzen entstanden und sollten immer auch kostengünstig sein. Damit wurden von je her die Topologien Bus und Ring bevorzugt, da sie einfacher und überschaubarer sind als Maschennetze oder Sternnetze und keine speziellen Internetworking-Geräte benötigen. Leider haben sie aber auch den Nachteil, daß ihre Redundanz und damit ihre Fehlertoleranz gering ist. Ein einfacher Leitungsausfall legt bereits das komplette LAN lahm. Daher wurden in letzter Zeit zusätzliche Mechanismen beigefügt (Ringsysteme) bzw bei den Bussystemen erfolgte die Reduktion auf eine Station pro Segment und damit der Übergang zum sternförmigen Netz.

Bei den Ringen kann man zB sternförmige Ringe bilden. Diese Topologie bewirkt die Bildung einer Art “Zentralstation”, die in diesem Fall allerdings sehr einfach gehalten werden kann. Diese Zentralstation wird oft “Hub” genannt. Ihr richtiger Name lautet im Deutschen “Ringleitungsverteiler” und im Englischen “Multiple Station Attachment Unit” – “MSAU”. Eine der berühmtesten MSAUs ist IBM’s “8228 Unmanaged MSAU”.



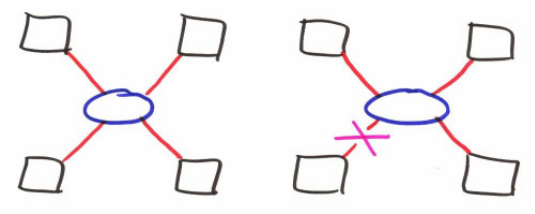
Skizze MSAU – optische Sicht – kabelmäßige Sicht – elektrische Sicht

Die Aufgabe der MSAU ist es, bei Ausfall einer Leitung bzw einer Station diese vom Ring abzutrennen und den Ring physisch wieder zu schließen. Die IBM 8228 MSAU bewerkstelligt dies auf rein passivem Weg, indem jede Station eine Gleichspannung auf die Kabel zur MSAU legen muß. Diese Gleichspannung überlagert die Wechselspannung, mit der die Information transportiert wird, die beiden Spannungen koexistieren und stören einander nicht. Während nun die Wechselspannung die Daten überträgt, dient die Gleichspannung dazu, ein Relais in der MSAU zu schließen. Damit bleib die Station physisch in den Ring eingebunden, solange sie die Gleichspannung auf den Kabeln aufrechthält. Wird die Station abgeschaltet oder das Kabel durchtrennt, fällt das Relais ab und schließt die Station kurz und damit aus dem Ring aus.



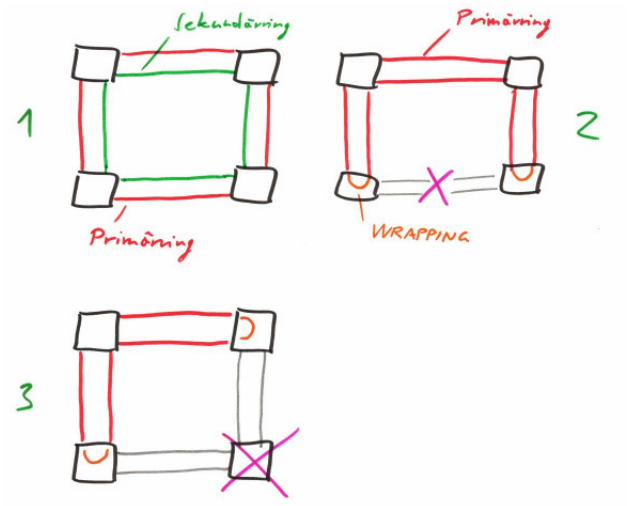
Skizze Innerer Aufbau der IBM 8228 MSAU

Sternförmige Ringe verkraften bereits den Ausfall einer Station bzw eines Kabelstücks:



Skizze Rekonfig sternförmiger Ring

Doppelringe scheiden die defekten Kabelsegmente bzw Stationen durch aktives “abtrennen” (“Ring Wrapping”) in den äußersten Stationen ab. Dabei wird in den beiden “Wrapping Stations” der primäre mit dem sekundären Ring verbunden. Ein Doppelringsystem rekonfiguriert sich daher im Fehlerfall zu einem Einfachring. Ein weiterer Fehlerfall führt zu einem erneuten “Wrappen”, dann zerfällt der Ring in zwei Subringe. Ein Doppelringsystem kann daher einen Stations- bzw Kabelfehler überstehen.



Skizze Rekonfiguration des Doppelrings

Der zweite Ring eines Doppelringsystems kann entweder “leer” laufen oder für den Datentransport verwendet werden. Normalerweise verzichtet man aus Gründen des Aufwands (und damit der Kosten) darauf, den zweiten Ring aktiv zu verwenden (außer im Rekonfigurationsfall).

Physische Schicht - Medien

Die Unterteilung der Medien erfolgt in “tatsächliche”, physische Kabel (“hardwire media”) und die kabellose Übertragung (“softwire media”). Der bei weitem größte Teil aller LANs verwendet heute Kabel zur Übertragung der Daten. Dies wird sich aber in nächster Zeit mit der Ausbreitung der Funk-LANs (zB IEEE 802.11) ändern.

Eine grobe Einteilung der Übertragungsmedien sieht folgend aus:

- Softwire Medien
  - Infrarot (optisch, nicht kohärent)
  - Laser (optisch, kohärent)
  - Funk (Radiowellen)
  - Mikrowellen (ultrakurze Radiowellen)
- Hardwire Medien
  - Metallkabel (“Copper Wire”)
    - Twisted Pair
    - Koaxiales Kabel
  - Lichtwellenleiter (“Fiber”)

Die hardwire Medien zerfallen in drei große Kabelsysteme: TP, KoAx und LWLs (siehe unten). Diese werden wir uns näher ansehen.

Die zuständigen Normungsgremien für Verkabelungen sind die EIA (“Electronic Industries Alliance”) und die TIA (“Telecommunications Industry Association”). Sie entsprechen in ihrere Rolle der IEEE, sind aber nbur für die Verkabelungssysteme zuständig und legen deren Charakteristiken und Anwendungsfälle fest.

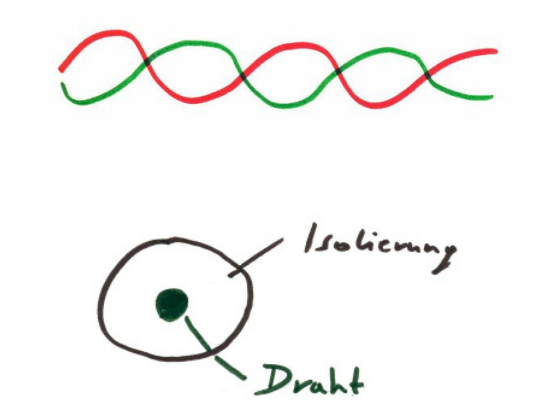
Da bei einer Neuinstallation eines lokalen Netzes der Kostenanteil der Verkabelung ein sehr relevanter ist, kommt der Auswahl des “richtigen” Kabels besondere Bedeutung zu.

Twisted Pair

Twisted Wire, auch Twisted Pair oder kurz “TP” genannt, stammt aus der Telefonie und ist ein bestens eingeführtes und in weltweiter Verwendung befindliches Kabel. Es ist daher sehr günstig, technisch ausgereift, überall und in enorm vielfältigen Ausprägungen und Bündelungen erhältlich und heute der Standard für sowohl LANs, als auch für die “last mile” der WANs und der Telefonie.

Unter der “last mile” wird die Strecke vom letzten Kommunikationsgerät zur eigentlichen Steckdose in einem Haushalt verstanden. Die “last mile” ist bis heute ausschließlich in “Kupfer-Technologie” gehalten, kaum Haushalte sind mit LWLs angebunden. Da vom letzten Kommunikationsgerät bis zu den Steckdosen in den Haushalten aber Millionen und Abermillionen von Kabel-Kilometern liegen und der Austausch dieser Kabel unwirtschafltich teuer ist, ist die “last mile” heute definitiv der Hemmschuh für den Hochgeschwindigkeits-Anschluß der Haushalte.

In seiner einfachsten Form besteht TP aus zwei isolierten Kabeln, die miteinander verdrillt sind.



Skizze TP längs und im Querschnitt

Für bessere Übertragungsqualität wird auf lange Strecken ein Draht verwendet (“Solid Core”). Sogenannte “Patch-Kabel” werden verwendet, um sich häufig ändernde Verkabelungen an einem vorgesehenen Punkt, dem “Patch Panel”, zu verändern. Da diese Kabel auf engstem Raum innerhalb des Panels verlegt werden müssen, sind sie nicht als Draht, sondern als Litze (“Stranded Core”) ausgeführt. Das macht sie flexibler, hebt aber die Verluste um 20%.

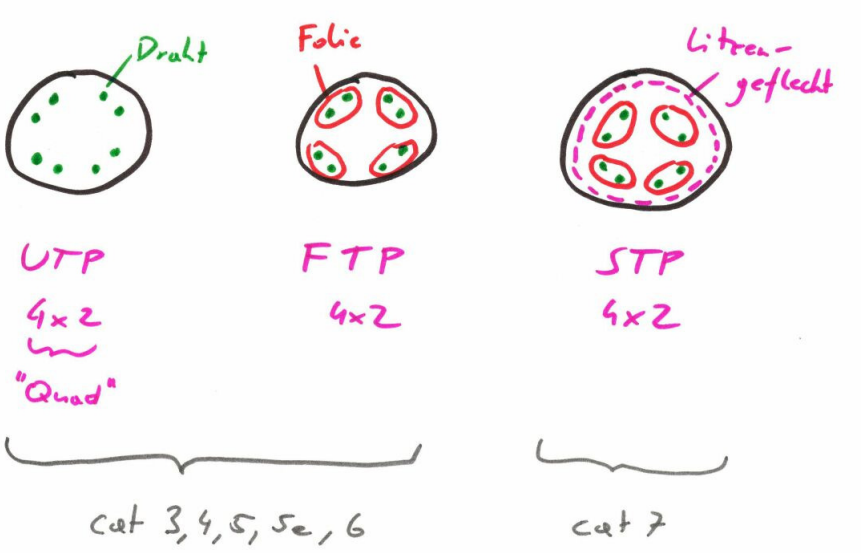
Die Anzahl der gegenseitigen Verdrehungen pro Fuß (“twists per foot”) charakterisiert das Kabel ebenso wie sein Durchmesser (gemessen in “American Wire Gauge”, “AWG”). Die AWG Kennzahl gibt an, wie oft der Kabeldraht bei der Fertigung durch eine immer kleiner werdende Öffnung gepresst wurde. Je höher die AWG-Zahl, desto dünner der Draht. Allgemeine TP-Drähte haben von 0,1 mm bis ca 3 mm Durchmesser. Gängige AWGs bei den LANs sind:

AWG	**mm Durchmesser**
22	0,6
24	0,5
26	0,4

Tabelle American Wire Gauge Einheiten

Einfaches TP wird heute auch als UTP – “Unshielded TP” – bezeichnet. Um nämlich die

elektrischen Eigenschaften des einfachen TP-Mediums zu verbessern, wurde recht bald eine billige, einfache Schirmung aus Alufolie um jedes Kabelpaar herumgelegt. Diese - Englisch als “foil” bezeichnete – Verbesserung des TP setzte sich rasch durch und wird als “Foiled TP” oder kurz “FTP” bezeichnet. Letztendlich wurden die Anforderungen an die TPs derart hoch (siehe unten unter Cat-7), daß man die FTPs zusätzlich mit “echten” Schirmungen aus Kupfer/Alulitze umfassen mußte. Das entstehene Kabel ist eigentlich ein KoAx-Kabel (siehe unten) mit einem oder mehreren TPs als Innenkabel statt einem einzigen Draht.



Skizze UTP, FTP und STP

Wie man sieht, kommt TP praktisch nie als “pures” 2x1-Draht Kabel vor, sondern wird minimal als 4x2 Drähte zusammengefaßt.

TP-Kabel haben zwischen 100 Ohm (UTP) und 150 Ohm (STP) Widerstand und dürfen normalerweise nur bis ca 100N (das sind nicht einmal 11 kg) Zugkraft belastet werden. Das “Einziehen” solcher Kabel in bestehende Verrohrungssysteme mittels Einziehfeder ist daher selten möglich, TP wird daher oft “gelegt” statt eingezogen.

Seit einigen Jahren werden TP-Kabel speziell für den Einsatz in LANs konzipiert und für die bei LANs gewünschten Charakteristika erstellt. Dazu hat sich die Kategorie-Schreibweise der TIA/EIA-568-A etabliert:

CAT	**max. MHz**	**Spezifikation**	**Steckverbinder**
3	16	2adriges Telefonkabel, “Voice Grade”	RJ-45
4	20	2adriges Telefonkabel, “Voice Grade”	RJ-45
5	100	2adriges Datenkabel, “Data Grade”	RJ-45
5e	100	4adriges Datenkabel	RJ-45
6	250	4adriges Datenkabel, auch “Class E” genannt	RJ-45
7	600	4adriges Datenkabel, auch “Class F” genannt, nur noch STP möglich	GG-45

Tabelle CAT-Spezifikationen beim TP

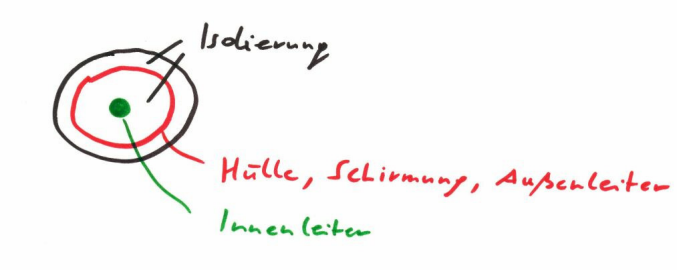
Die Steckverbinder der TPs sind universell die “Radio Jackets Type 45”, kurz “RJ-45”. Diese sind bis Cat-6 brauchbar, erst mit Cat-7 muß man auf die wesentlich komplexeren und teureren GG-45 umsteigen, die aber physisch (nicht jedoch von der Dämpfung) zu RJ-45 kompatibel sind.

Spätestens bei Cat-7 wurde klar, daß die Zukunft der Hardwire Medien nur der Lichtwellenleiter sein kann, da die Kosten des Cat-7 Kabels bereits nahe an den Kosten eines LWLs liegen, das TP aber bei weitem nicht die Leistung eines LWLs erbringen kann. Während Cat-7 bereits ein absolut ausgereiztes System ist, mit dem man derzeit an die 600 Mbps übertragen kann, ist der LWL bei 600 Mbps erst bei einem winzigen Bruchteil seiner Kapazität angelangt. Heutige LWLs kommen leicht in den Bereich von Terabits pro Sekunde bei Entfernungen von einigen Kilometern.

Koaxiales Kabel

Bei den LANs spielte das Koaxiale Kabel, kurz auch KoAx genannt, nur eine kurze Rolle, nämlich im Ur-Ethernet. Es ist gegenüber dem TP wesentlich leistungsfähiger, allerdings auch teurer, dicker und unhandlicher. Die Verlegungskosten liegen meist über dem des TPs, da das Kabel umständlicher in der Handhabung ist.

KoAx-Kabel wurden für die ursprüngliche Bus-Topologie des Ethernet verwedet. In dem Moment, wo sich diese pure Bus-Topologie als zu fehleranfällig herausstellte und man auf den Stern überging (ab 10BASETx), übernahm das TP die Rolle der führenden Verkabelung bei Ethernet und damit zugleich der Großteil aller LANs. In anderen LANs kam KoAx nie stark zum Einsatz.



Skizze Aufbau KoAx im Querschnitt

Der Durchmesser des Ur-Ethernet-KoAx (sogenanntes “yellow cable” oder auch treffend “thick wire” genannt, offizielle Bezeichnung “RG-59A”) war recht gewaltig und betrug bis zu 2,5 cm. Die spätere Ethernet-Form “Cheapernet” verwendete das wesentlich dünnere “thin wire” KoAx mit “nur” noch 0,8 cm Durchmesser (offiziell “RG-58A”).

Beide KoAx-Kabel haben einen Widerstand von 50 Ohm - im Gegensatz zum Antennenkabel, das ebenfalls ein KoAx-Kabel ist und dem “thin wire” zum Verwechseln ähnlich sieht, aber 75 Ohm Widerstand aufweist.

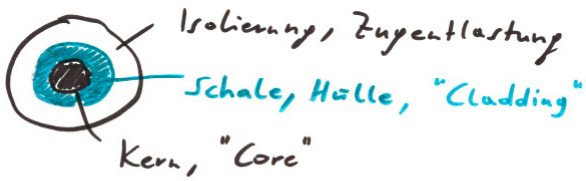
Das bei KoAx-Kabeln verwendete Steckersystem ist der N-Connector bei den “thick wires” bzw die BNC-Stecker bei “thin wires”. Beides sind Bajonettverschlüsse.

Koax-Kabel sind also an beiden Enden “offen” und müssen daher mittels Abschlußwiderständen elektrisch “geschlossen” werden. Diese Abschlußwiderstände – Im Englischen “terminator” genannt – verhindern, daß ein elektrisches Signal beim Auftreffen auf das offene Kabelende reflektiert wird (vergleichbar mit einer Welle, die gegen eine Mauer läuft). Der Abschlußwiderstand muß denselben Widerstand (genauer gesagt: dieselbe charakteristische Impedanz) wie das Kabel aufweisen. 50 Ohm Koax-Kabel benötigt daher auch 50 Ohm Terminatoren. Der Grund, warum ein Bussystem im Fehlerfall, zB beim Auftrennen der Leitung, nicht mehr funktioniert, liegt darin, daß

man nun zwei zusätzliche “Enden” hat, die nicht terminiert sind und die Übertragung stark beeinträchtigen.

Lichtwellenleiter

Das hardwire Medium der Zukunft ist aus heutiger Sicht eindeutig der Lichtwellenleiter, kurz “LWL” genannt. Im Englischen verwendet man die Bezeichnung “Fiber Optics Cable”, kurz “Fiber”.



Skizze LWL, Querschnitt

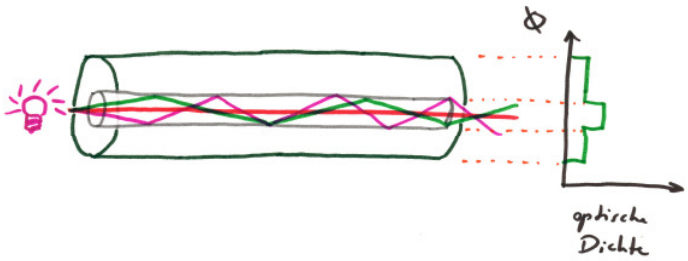
Lichtwellenleiter werden im Deutschen oft recht salopp als “Glasfaserkabel” bezeichnet. Dieser Ausdruck ist aber nur dann richtig, wenn es sich tatsächlich um ein Glasfaserkabel (Silizium) handelt. Teilweise wird aber Kunststoffkabel verwendet, da dies in der Fertigung billiger kommt.

Die Funktionsweise des LWL beruht auf der Totalreflektion der Lichststrahlen beim Übergang von einem optisch dichteren Medium (genannt der “Kern”, Englisch “Core”) in ein optisch dünneres Medium (genannt die “Hülle”, Englisch “Cladding”). Die Grenzschicht reflektiert Lichtstrahlen, die unter einem bestimmten Winkel eintreffen, total.

Als Lichtquelle wird in billigen Systemen eine LED verwendet, zumeist im “short wave” Bereich, also bei ca 850nm. Dieses Licht ist rot und sichtbar. Teurere Systeme verwenden Laserdioden (“Injection Laser Diodes”, “ILD”) , die im “short wave” oder auch im “long wave” Bereich bei 1310nm und 1550nm (Infrarot, für das menschliche Auge unsichtbar, daher auch geheimnisvoll “dark fiber” genannt) arbeiten. Der Vorteil der Laserdiode ist das kohärente Licht, das wesentlich effektiver übertragen wird und eine höhere Reichweite ermöglicht.

Multimode

Die Durchmesser der einfachen LWL liegen in Europa bei 50/100µm (Mikrometer Durchmesser des Kerns / Mikrometer Durchmesser der Hülle), in den USA bei 62,5/125µm. Das Durchmesser/Dichte-Diagramm eines einfachen LWL sieht folgend aus:



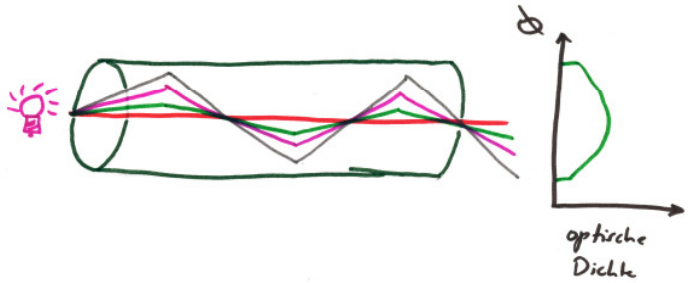
Skizze Multimode-LWL, Durchmesser/Dichte Diagramm

Man sieht deutlich den Sprung in der Dichte zwischen Kern und Hülle. Man nennt diese Art der

LWL auch “Multimode-LWL” (Englisch: “Multi Mode Fiber”, MMF). Multimode deutet auf die Moden hin, das sind quantenmechanische Begriffe, die wir uns näherungsweise als “Lichtstrahlen” vorstellen können. In einem Multimode-LWL können mehrere Moden zugleich durch das Kabel gelangen. Damit ist aber die Laufzeit der einzelnen Moden unterschiedlich, da eine Mode, die öfter totalreflektiert wird, auch länger durch das Medium wandert. Die zentrale Mode nimmt den kürzesten Weg durch den LWL. Die Laufzeitunterschiede bewirken, daß das in den LWL eintretende Licht beim Austritt aus dem LWL “verrauscht” ist. Technisch nennt man das die “modale Dispersion”. Damit ist aber die überbrückbare Entfernung für Datenübertragungen mit LWLs stark reduziert, da zwar das Licht kaum gedämpft wieder austritt, die Laufzeitunterschiede aber das Signal “verschmieren”. Aus einem kurzen Puls auf der Eingangsseite wird ein breiter Puls auf der Ausgangsseite.

Graduierlicher Multimode

Um dieses Phänomen zu bekämpfen, wurden die graduierlichen Multimode-LWLs erfunden. Bei dieser Technik gibt es keinen sprunghaften Übergang in der Dichte von Kern zu Hülle, sondern es erfolgt eine Reihe kleiner Übergänge, die die Dichte graduierlich vom Kern zur Hülle reduzieren. Eintreffende Lichtstrahlen werden – abhängig von ihrem Eintrittswinkel – an einer dieser Grenzen totalreflektiert. Wenn man nun die Übergänge sorgfältig wählt, erfolgt die Totalreflektion aller Lichtstrahlen immer an derselben Stelle im LWL. Die Lichtstrahlen werden quasi fokussiert, sie legen aber bei der Totalreflexion in einer weiter außen liegenden Grenzschicht einen weiteren Weg zurück als bei der Totalreflexion an einer inneren Grenzschicht. Diese Wegunterschiede werden durch die immer höher werdenden Lichtgeschindigkeit in den äußeren Schichten (wegen der immer geringeren Materialdichte) kompensiert.



Skizze graduierlicher Multimode-LWL, Durchmesser/Dichte Diagramm

Man erkennt die “weichen” Übergänge der Dichte. Die Fertigung dieser Art von LWL ist aber aufwendiger und damit teurer als einfache LWLs, die man auch Multimode LWLs nennt. Da ihre Effektivität aber noch immer nicht optimal ist, ging man daran, den Durchmesser der LWL zu verringern.

Monomode

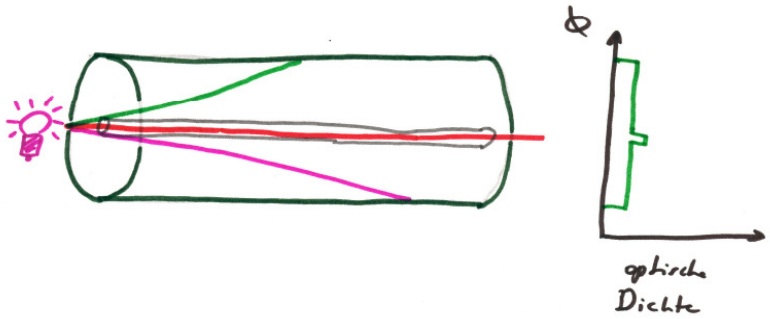
Es entstand der Monomode LWL (Englisch: “Single Mode Fiber”, SMF). Bei diesem wird nur eine einzige Mode durch den LWL gelassen. Um dies zu erreichen, muß man den Durchmesser des Kerns in die Größenordnung der Wellenlänge des zu übertragenen Lichts bringen. Genauer gesagt muß der Kerndurchmesser kleiner als ca 6 mal die Wellenlänge des verwendeten Lichts sein. Bei heutigen Monomode-LWLs wird ein Kern mit 9µm Durchmesser verwendet (Hülle 125µm). Ein kleinerer Kerndurchmesser ist in der Fertigung teuer und auch nicht nötig, da der SMF-Effekt bereits bei 9µm Kerndurchmesser eintritt, vorausgesetzt man verwendet Licht mit 1,3µm bzw 1,55µm Wellenlänge (“long wave”). Alle SMFs arbeiten daher als “Dark Fiber”, verwenden also

“unsichtbares Licht”.

Damit ist das Problem des verrauschten Signals, das im Multimode-LWL stark stört und vom graduierlichen Multimode-LWL teilweise und auf komplizierte Art verbessert wurde, vollständig verschwunden. Der LWL ist nur noch durch seine Verschmutzung im Glas/Kunststoff behindert (dies bekommt man mittlerweile sehr gut in den Griff) und einen als “chromatische Abberation” oder “chromatische Dispersion” bezeichneten Effekt. Dieser bewirkt, daß Licht verschiedener Wellenlänge sich unterschiedlich schnell ausbreitet. Das Ergebnis ist dieselbe “Verschmierung”, die auch bei der modalen Dispersion auftritt. Je reiner also die Frequenz des Lichtes ist, desto geringer wird dieser Effekt.

Kommerzielle SMF ermöglichen heute Übertragungsraten im Gbps-Bereich bei Längen von etlichen Kilometern (derzeit ca 40km bei 10Gbps).

Das Durchmesser/Dichte-Diagramm eines Monomode-LWL sieht folgend aus:

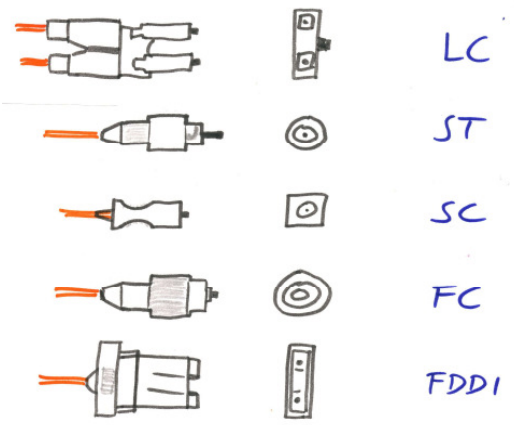


Skizze Monomode-LWL,Durchmesser/Dichte Diagramm

Man sieht hier wieder deutlich einen Dichte-Sprung und den kleinen Kernduchmesser.

LWL Steckersysteme

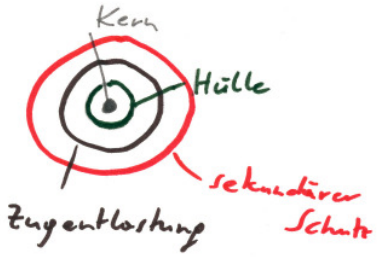
Während bei TP und KoAx die Steckersysteme universell sind, haben sich bei den LWLs vier wichtige Steckersysteme herausgebildet, die leider völlig inkompatibel zueinander sind.



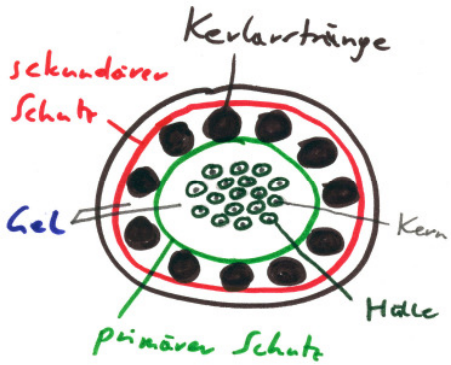
Skizze LWL Steckverbinder LC, ST, SC, FC, FDDI

LWLs werden in verschiedenen Bauformen verlegt: als Einzelleitung, als Pair (Leitung hin und Leitung zurück, getrennt ausgeführt) und als Trunk (Kabelbündel mit vielen LWLs). Da LWLs

immer nur unidirektional verwendet werden, muß man zB bei Ethernet immer LWL-Paare legen. Beim Token Ring und anderen einfachen Ringsystemen reicht dagegen ein einzelner LWL "zur Station hin" und einer "von der Station retour".



Skizze Einzel-LWL



Skizze "LWL Trunk Cable"

LWL Dämpfung

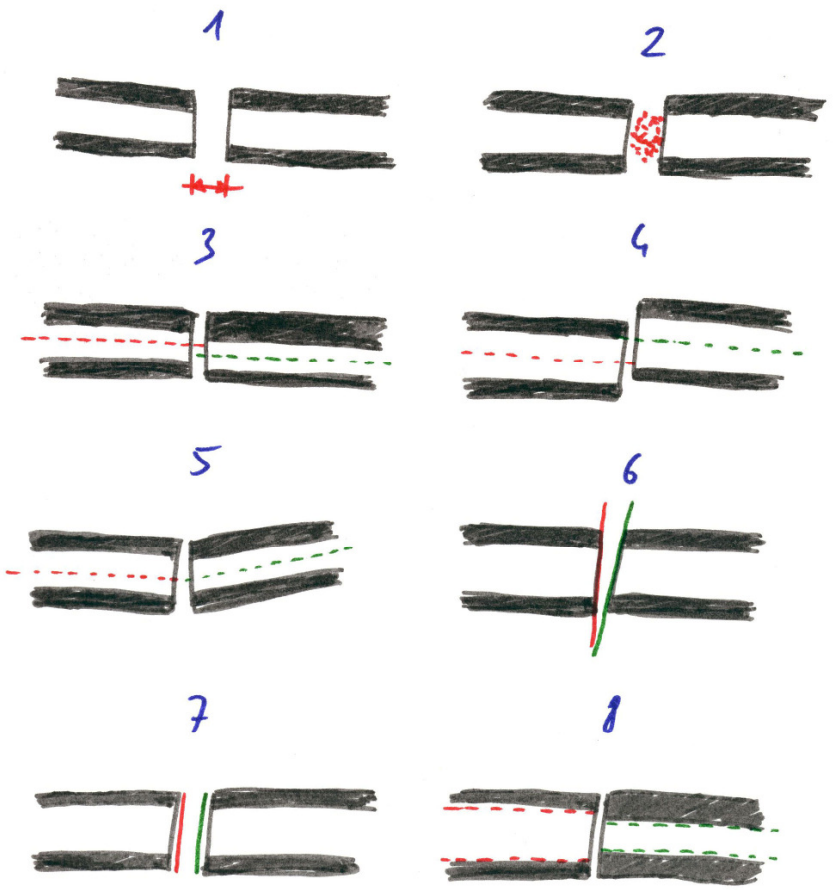
Mode	**Durchmesser (µm Kern / µm Hülle)**	**Lichtwellen-länge (nm)**	**Dämpfung (dB/km) bei MHz**	**Einsatz vorwiegend in:**
Multi mode	50 / 100	850 short wave	3,5	EU
Multi mode	62,5 / 125	850 short wave	3,5	USA
Multi mode	50 / 100	1200 long wave	1,0	EU
Multi mode	62,5 / 125	1200 long wave	1,0	USA
Single mode	9 / 125	1310 long wave	0,4	Universell
Single mode	9 / 125	1550 long wave	0,3	Universell

TabelleDämpfung beim LWL

Zum Vergleich: IBM Kabel Type 1 (TP) hat 46 db/km Dämpfung bei 16 MHz.

LWL Steckverbinder

Ein weiterer limitierender Faktor bei LWLs ist das “Splicing”, also das Verbinden von LWLs bzw der Übergang LWL zum Stecker. Hier können die folgenden Probleme auftreten:



Skizze Splicing Probleme beim LWL

1	zu großer Abstand zwischen den LWLs
2	Schmutz zwischen den LWLs
3	nicht axial zentrische LWLs
4	LWLs nicht axial zentrisch aneinandergefügt
5	LWLs nicht axial gerade miteinander verbunden
6	LWLs gerade verbunden, aber Kontaktfläche nicht plan
7	Material und damit Brechungsindex der beiden LWLs unterschiedlich
8	Kernduchmesser der beiden LWLs nicht gleich

Das Anfügen von Steckverbindern für LWL hat sich in letzter Zeit wesentlich vereinfacht und vor allem verschnellert. Der ursprüngliche Vorgang des Anbringens eines Steckverbinders sah folgend aus:

Die Plastikhülle des LWL einige cm entfernen.

Ein 2-Komponenten Epoxy-Harz anrühren.

Das angerührte Harz in den Stecker gießen.

Den LWL in den Stecker einführen, bis die Plastikhülle im Zugentlastungsbereich des Steckers sitzt. Der eigentliche LWL steht nun aus dem Stecker vorn einige cm heraus.

Den festen Sitz der Zugentlastung um die Plastikhülle sicherstellen (Crimp-Werkzeug oÄ).

Nun das Harz härten lassen. Bei den alten 2-Komponenten Harzen dauerte das eine halbe Stunde im Ofen bei einer vorgegebenen Temperatur (meist

um die 120° C).

Anschließend wird der Stecker abgekühlt und die vorne aus dem Stecker herausstehende Faser mit einem Cutter anritzen und abbrechen.

Dann erfolgt der Schleif- und Poliervorgang. Hierbei wird mit immer feineren Schleifmitteln die Fiber mit dem Kopf des Steckers plangeschliffen und anschließend poliert. Der Vorgang des Planschleifens und Polierens entscheidet über die eigentliche Qualität der Verbindung. Wird hier schlampig gearbeitet, so kommt es zu einem der oben im Bild angegebenen Fehlerquellen und damit zu Lichtverlusten, die Leistungs- und Bandbreitenverlusten entsprechen.

Nun wird die polierte Fläche peinlichst genau gesäubert. Dafür werden fuselfreie Tücher bzw Wattestäbchen – in 99% (Isopropyl) Alkohol getaucht – verwendet.

Mit einem Mikroskop (Vergrößerung 100x bis 200x) wird die Fiber-Fläche auf Planheit überprüft und gegebenenfalls bei Schritt 8 weitergemacht.

Abschließend wird der LWL auf Lichtverlust getestet. Dafür benötigt man ein eigenes Gerät. Bei MMF genügt die Messung des Lichtverlustes bei Lichtdurchtritt, bei SMF muß auch die Reflexion an der polierten Fläche gemessen werden.

Dieses klassische Verfahren hat den Vorteil, daß der Stecker optimal haltbar ist und die Materialkosten pro Stecker gering bleiben. Der Nachteil ist der große Zeitaufwand (10 Minuten für die Bearbeitung plus 20 Minuten im Ofen), das Hantieren mit dem 2-Komponenten Harz und der Einsatz des Ofens.

Neuere Verfahren versuchen eine schnellere und einfachere Bearbeitung zu erzielen, jedoch sind die so entstehenden Stecker teurer pro Stück und die Haltbarkeit ist geringer. Bei Verwendung von Cyanacrylat (aka “Superkleber”) dauert das Verkleben nur noch Sekunden und funktioniert ohne Ofen, was aber auch die Zeit für die Bearbeitung auf Sekunden reduziert und einen erfahrenen Verarbeiter voraussetzt. Es gibt auch Varianten mit Aushärtung per UV-Licht, was schneller geht als normales Epoxy und dem Bearbeiter mehr Verarbeitungszeit läßt, aber eine spezielle UV-Apparatur voraussetzt. Auch Epoxys, die bei Raumtemperatur aushärten, gibt es bereits. Spezielle 2-Komponenten Systeme tauchen den LWL in die eine Komponente und in den Stecker wird die zweite Komponente eingefüllt. Die Aushärtung erfolgt dann anaerob (durch Abscheiden der Sauerstoffzufuhr an der Klebestelle). Ganz schnelle Verfahren verwenden ein reines Crimp-System für sowohl den LWL als auch die Plastikhülle. Das Festhalten des LWLs erfolgt dann nur mittels Verklemmen, was aber den LWL für Dauerbelastung wenig geeignet macht.

Man sollte LWLs nie stärker biegen als den Durchmesser einer Faust (eine klassische “Faust”-Regel). Außerdem ist es extrem wichtig, die offenen Enden der LWL sauber zu halten, da selbst mikroskopische Kratzer oder Staubpartikel bei einem LWL mit µm Kerndurchmesser riesige Lichtmengen vernichten bzw eine sogenannte “Luftbrücke” (Englisch “Air Gap”) bilden, an der der Brechungskoeffizient der Luft plötzlich (übel) mitspielt.

Wesentlich bei der Bearbeitung von LWLs ist, daß man die Bruchstücke und Abfallstücke der LWLs sorgfältigst in einem eigenen Behälter oder besser auf Klebeband deponiert und immer eine Schutzbrille trägt. LWLs im µm Bereich können leicht unter die Haut gelangen, eingeatmet werden oder ins Auge gelangen, was sehr unangenehme Folgen haben kann.

Ferner ist es ganz wichtig, NIE in ein offenes LWL-Ende bzw in einen Stecker hineinzuschauen. Die Laserleistung heutiger LWLs kann die Netzhaut schädigen. Das ist besonders bei “Dark Fiber” im 1310nm bzw 1550nm Bereich gefährlich, da man das Infrarotlicht nicht sieht.

Photonik

Derzeit wird bereits an den WDM-LWLs (“Wave Division Multiplexing”-LWLs) geforscht. Hierbei wird Licht verschiedener Frequenzen zugleich über einen LWL übertragen. Derzeit ist es möglich, bis zu einigen Hundert verschiedene Wellenlängen zugleich zu übertragen. Damit steigert sich die Datenrate der LWLs in den Petabit-Bereich (Pbps, 10^15 bps).

Als unerwartetes Problem kommt derzeit die unzulängliche Leistung der heutigen Rechner hinzu.

Eine CPU ist – auch wenn sie noch so spezialisiert ist – nicht in der Lage, einen Petabit-pro-Sekunde Datenkanal dauerhaft mit Daten zu versorgen. Bridges, Switches und Router werden daher in nächster Zeit nicht mehr mittels Elektronik, sondern mittels einer neuen Technologie namens "Photonik" ausgestattet werden. Dabei übernehmen "Lichtschalter" im wahrsten Sinne des Wortes die Rolle der Datenweiterleitung anhand von Dateninhalten. Damit entfällt die Umwandlung "optisch-elektronisch-optisch" und die damit einhergehende "langsame" Verarbeitung der Daten auf elektronischem Weg.

Softwire Medien

Bei den Softwire-Medien gibt es heute die Techniken Radio, Mikrowelle, Infrarot und Laser.

Bei Infrarot und Laser müssen Quelle und Ziel aufeinander einjustiert werden, Radio und Mikrowellen werden mehr gestreut gesendet.

Infrarot und Laser durchdringen Glas und sind damit auch für Übertragungen im Innenbereich von Gebäuden teilweise tauglich. Mikrowellensender werden normalerweise im Außenbereich installiert. Mikrowellen-Strahler sind aufgrund ihrer Radio-Charakteristik bei der Post meldepflichtig.

Wegen seiner einfacheren Handhabung und der lizenzfreien Öffnung bestimmter Frequenzbereich im Low-GHz-Spektrum (zB 2,400-2,4850 GHz, sogenannter „ISM"-Bereich, „Industry, Scientific, Medical" oder 5 GHz) werden sich in nächster Zeit für die LANs wahrscheinlich die Radio-Techniken durchsetzen (Stichworte "Blue Tooth" und IEEE 802.11 „WiFi"). Beim Richtfunk ("Point-to-Point") ist Laser bzw Infrarot schneller und effektiver verwendbar.

Eine neue Klasse von Übertragungsarten stellt das sogenannte „Ultrawideband" System dar. Hierbei wird ein sehr großes Spektrum an Übertragungsfrequenzen zugleich verwendet. Es handelt sich um ein nicht-modulierendes Verfahren, also um Basisband-Übertragung (siehe weiter unten). Die verwendete Sende-Energie ist aber sehr gering, und zusammen mit der Verteilung der bereits geringen Energie auf ein sehr weites Spektrum (von 0Hz bis weit in den GHz-Bereich hinein) ist die in einem bestimmten spektralen Bereich anzutreffende Energie äußerst gering. Dennoch kann hiermit ein leistungsfähiges Kurzstrecken-Kommunikationssystem (PAN) aufgebaut werden, indem man viele Frequenzen zugleich verwendet.

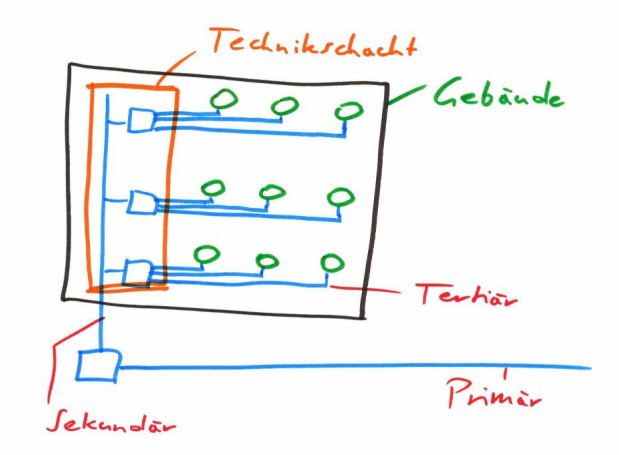
Verkabelungskosten

Studien in letzter Zeit haben ergeben, daß bei großen Verkabelungsvorhaben die Kostenstruktur ca folgend ausfällt:

Kabelkosten:	10% der Gesamtkosten
Arbeitszeit:	50-60% der Gesamtkosten
NICs, Geräte, etc	Rest

Tabelle Kostenverteilung bei der Netzwerkinstallation

Damit ist klar, daß man bei der Verlegung neuer Kabel immer danach trachten sollte, die besten leistbaren Kabelsysteme zu verlegen, da ein baldiges Neuverlegen durch die hohen Arbeitskosten nicht rentabel ist. Dies erklärt auch den Trend, heute bereits LWLs in Bündeln zu verlegen, auch wenn dies immer noch die teuerste Kabelvariante ist.

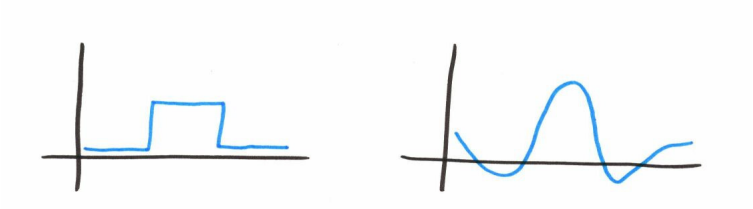


Skizze Primär, Sekundär-, Tertiär-Verkabelung

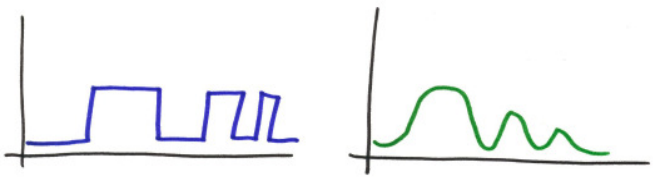
Physische Schicht – Übertragungstechnik und Kodierung

Basisband und Breitband

Bei der Basisband-Übertragung wird das gesamte verfügbare Frequenzspektrum des Mediums für die Übertragung eines Signals ausgenutzt. Man sendet dabei die digitale Information (Spannungspuls, Lichtpuls) direkt per Verstärker in das Medium. Damit nutzt man zwar das gesamte Spektrum des Mediums aus, verschenkt aber viel von seiner Leistungsfähigkeit. Da dieses Verfahren sehr günstig zu realisieren ist, war es bei LANs immer bevorzugt verwendet.



Skizze Rechtecksignal "vorher" und "nachher"



Skizze Rechtecksignalfolge "vorher" und "nachher"

Bei der Breitband-Technik hingegen wird zuerst ein Trägersignal gebildet. Dieses ist eine Welle einer bestimmten, festen Charakteristik (genauer: eines bestimmten Frequenzbereichs). Auf dieses Trägersignal wird dann per Modulation das Datensignal aufgebracht. Dabei stehen die Amplitudenmodulation (AM), die Frequenzmodulation (FM) und die Phasenmodulation (PM) zur Verfügung. Auf der Gegenseite wird das Datensignal durch Demodulation wieder zurückgewonnen. Die dabei verwendeten Geräte heißen Modems (für MOdulator/DEModulator).

Modems im herkömmlichen Sinne (also solche, die in der digitalen Übertragung über

Telefonletungen verwendet werden) arbeiten mit AM, FM und PM im Audionbereich.

Im Hochgeschwindigkeitsbereich arbeiten sogenannte “Breitband-Modems” oder “Kabel-Modems”. Kabel-Modems sind eine spezielle Form, da hierbei die Information auf einem bestehenden System mitübetragen wird (derzeit primär dem Kabelfernsehen).

Um gestörte oder unbrauchbare Frequenzbereiche gezielt umgehen zu können, wird normalerweise nicht nur eine einzige Trägerfrequenz verwendet, sondern eine Vielzahl von solchen. Damit kann man unbrauchbare Frequenzbänder einfach ungenutzt lassen.

Bei LANs hat sich universell die Basisband-Methode durchgesetzt. Das letzte Breitband-System war eine spezielle Ethernet-Variante und ist schon vor langer Zeit “ausgestorben”. Mit dem voranschreitenden Umstieg auf LWLs ist die Verwendung von Breitband im LAN-Bereich in nächster Zeit nicht wahrscheinlich, da ein LWL auch im Basisband-Betrieb ausreichend Übertragungsleistung bereitstellt.

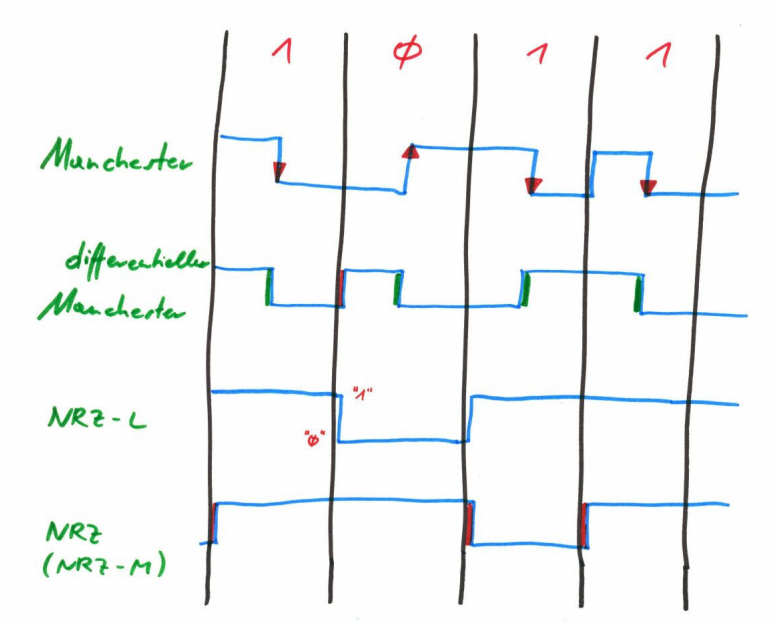
Kodierung

Wenn man Information moduliert oder unmoduliert übertragen will, muß man sie entsprechend kodieren. Diese Kodierung erfolgt bei den LANs heute im Basisband-Verfahren, bei Metallkabeln normalerweise durch Spannungswechsel, bei LWLs durch “Licht-an/Licht-aus” (Amplituden-Modulation).

Die Empfangsstation muß in der Lage sein, das gesendete Signal zu reproduzieren. Dazu müssen Sender und Empfänger über sehr genau gehende und hinreichend synchrone Uhren verfügen, oder die Taktinformation muß vom Sender zusammen mit den Daten mitgesendet werden.

Man arbeitet zB bei Analog-Modems mit synchronen Uhren. Dafür sind Start-Bits und eine auf beiden Seiten eingestellte Uhr (“Baudrate” genannt) notwendig. Dies ist im Bereich der Übertragung von einigen tausend Bits pro Sekunde ohne weiteres machbar. Bei LANs, wo in Millionen Bit pro Sekunde bis zu Miliarden Bit pro Sekunde gerechnet wird, müßten die Uhren sehr präzise synchron laufen und hinreichend oft synchronisiert werden.

Man hat sich daher bei den ersten LANs dafür entschieden, die Taktinformation gleich in jedem einzelnen übertragenen Bit mitzusenden. Die primären und ursprünglichen Kodierungsformen der LANs sind “Manchester” und “Differential Manchester”. Später kam das einfache NRZ (“No Return to Zero”) hinzu, das allerdings keine Synchronisationsinformation per se mitführt:



Skizze Manchester / Differentieller Manchester / NRZ

Manchester wechselt IMMER in der „Mitte“ jedes Bits von LO auf HI oder umgekehrt. LO auf HI kodiert eine „Null“, HI auf LO kodiert eine „Eins“. Zu Beginn eines Bits muß gegebenenfalls ein zusätzlicher Pegelwechsel eingefügt werden (wenn mehrere „Einsen“ oder mehrere „Nullen“ hintereinander gesendet werden).

Differentieller Manchester kodiert eine „Null“ durch einen Wechsel von HI auf LO ODER von LO auf HI am „Anfang“ eines Bits. Die Richtung des Wechsels ist egal, daher der Name „differentieller“ (nur die Änderung betrachtender) Manchester.

NRZ (“No Return to Zero”) kodiert in seiner einfachsten Form (NRZ-Level oder NRZ-L) eine „Eins“ duch ein HI und eine „Null” durch ein LO.

NRZ-M als differentieller Code kodiert eine „Eins“ durch eine Wechsel (HI auf LO ODER LO auf HI) am „Anfang“ des Bits.

Reiner Manchester überträgt zB beim Ur-Ethernet 10Mbps Daten und 10Mbps Taktinformation, insgesamt also 20Mbps.

Der differentielle Manchester verhält sich genauso, nur wird das Vertauschen der Kabel (Signal und Masse) durch die Übertragung der Information in der Änderung der Spannung egalisiert.

Beide Manchester-Varianten kennen sogenannte “Non-Data-Symbols”, genannt “J” und “K”. Diese den Manchester-Regeln widersprechen Spannungverläufe werden ausgenutzt, um spezielle Zustände auf der Leitung anzuzeigen (siehe Ethernet und Token Ring).

NRZ-Kodierungen verschwenden wesentlich weniger Bandbreite, da keine zusätzliche Synchronisierungs-Information mitgesendet wird. Eine Folge von „1en“ oder „0en“ ist daher für Verfahren, die NRZ-Kodierung verwenden, sehr ungünstig, da keinerlei Synchroninformationen übertragen wird.

Vorkodierung

Durch die immer höher werdenden Datenraten mußten dann doch die Synchronisierungs-Informationen verringert werden, um Teile des Overhead (100% bei Manchester!) in nutzbare Leistung umzusetzen. Man verwendet daher eine Vorkodierung und sendet dann im NRZ-Verfahren. Dabei kodiert ein Pegelwechsel eine “Eins”, das Fehlen eines Pegelwechsels stellt eine “Null” dar. Klarerweise muß unbedingt verhindert werden, daß zu viele „Nullen“ hintereinander gesendet werden, da die „Einsen“ im Datenstrom die Synchronisationsinformation mitführen. Dies ist die Aufgabe der Vorkoderung.

Wenn man reines NRZ-M verwendet, kommt beim Senden von Nullen kein Pegelwechsel im Medium vor. Damit verliert der Empfänger nach kurzer Zeit die Synchronisation zum Sender, da er die „Nullen“ nicht mehr richtig auseinanderhalten kann.

Am Beispiel 4B/5B erfolgt die Vorkodierung folgendermaßen: der Datenstrom wird in 4-Bit Gruppen unterteilt und jeder dieser 4-Bit-Gruppen wird eine 5-Bit Gruppe eindeutig und fest zugeteilt. Die 5-Bit Gruppe wird dabei so gewählt, daß die Anzahl der „Nullen“ bei jedem beliebigen Hintereinandersenden von 5-Bit Gruppen kleiner oder gleich drei bleibt.

```
0000  => 11110
0001  => 01001
…
1111  => 11101
```

Damit muß der Empfänger in der Lage sein, maximal drei „Nullen“ und damit maximal drei Zeiteinheiten ohne Synchronisationsinformation mit dem Sender synchron zu bleiben. Dies ist auch bei hohen und höchsten Übertragungsraten noch relativ einfach realisierbar.

Ferner entstehen zusätzlich zu den 16 kodierten Halbbytes weitere 16 „Nicht-Daten-Symbole“ („non Data Symbols“, so wie das „J“ und das „K“ bei der Manchester-Kodierung).

```
QUIET => 00000
IDLE  => 11111
J     => 11000
K     => 10001
T     => 01101
R     => 00111
…
```

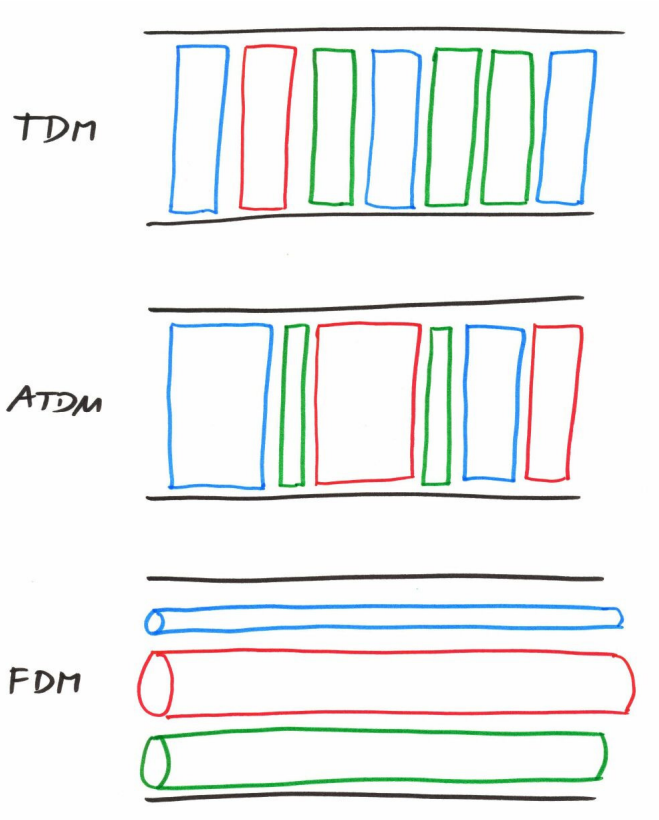
Durch diese Vorkodierung von 4 Bit auf 5 Bit (daher auch “4B/5B”) wird nur mit 25% Overhead gearbeitet.

Multiplexen

Um Informationen mehrerer Stationen zugleich oder quasi zugleich übertragen zu können, muß man die Daten “überlagern”, neudeutsch “multiplexen”. Dabei werden die Daten entweder wirklich zugleich gesendet. Dann muß man dafür sorgen, daß sie sich nicht gegenseitig stören. Die einfachste Methode dafür ist, mehrere Frequenzbänder zu bilden (siehe Breitband) und diese einzelnen Stationen entweder fix oder nach Bedarf zuzuweisen. Dieses Verfahren nennt man “Frequenzmultiplexen” bzw “Frequency Division Multiplexing”, FDM.

Alternativ kann man anstatt in der Frequenz auch “in der Zeit” multiplexen. Dann werden die sendewilligen Stationen “der Reihe nach” drangenommen. In jeder dieser Zeiteinheiten darf die Station dann ihre Daten senden. Man nennt dieses Verfahren “Zeitmultiplexen” bzw “Time Division Multiplexing”, TDM.

Es gibt zwei TDM-Varianten. Bei Synchronous TDM (STDM) wird in festen, immer gleich langen “Zeitscheiben” gesendet. Bei Asynchronous TDM (ATDM) sind die Zeiteinheiten dagegen unterschiedlich lang.



Skizze (S)TDM, ATDM und FDM

Speziell bei LWLs gibt es eine weitere Variante des Multiplexens, das “Wave Division Multiplexing” (“WDM”). Hierbei werden Lichtstrahlen verschiedener Wellenlänge zugleich in einen LWL gesendet. Dies entspricht in etwa dem FDM, nur eben mit verschiedenen Lichtwellenlängen (=Farben).

Datensicherungsschicht - Medium Access Control

Der MAC ist ein wesentlicher, die Charakteristik des LANs bestimmender Teil. Die Medium Access Control erfahren lassen sich in drei große Gruppen unterteilen:

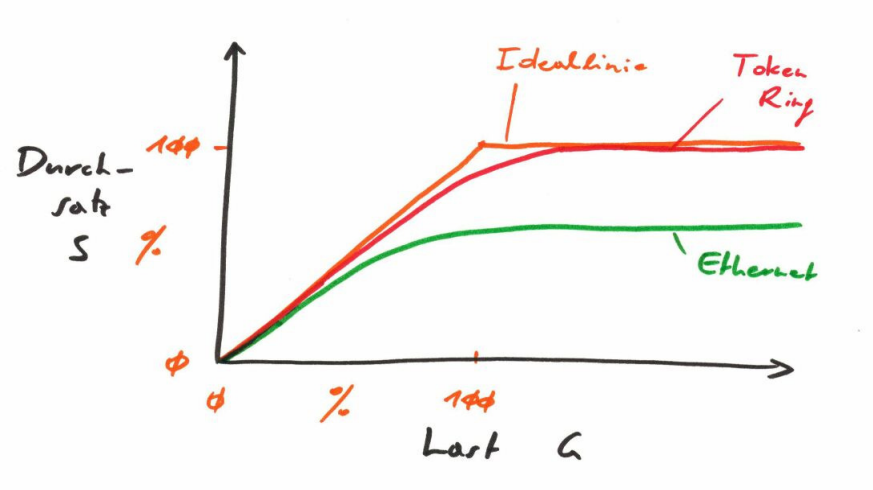
Wahlfreier Zugriff (Englisch: Contention)	Die Stationen eines Netzsegments entscheiden selbständig (also ohne untereinander zu kommunizieren), welche Station als nächstes sendet. Contention-Verfahren sind immer stochastischer Natur. Genaue Berechnungen oder die Bestimmung von Obergrenzen, zB: wann wer als nächster senden kann, wann spätestens er das nächste Mal senden kann oder wie oft jede Station in einem Zeitintervall senden kann, sind nicht möglich, lediglich statistische Aussagen können getroffen werden.
Verteilt gesteuerter Zugriff	Die einzelnen Stationen kooperieren, um den nächsten Sender zu bestimmen. Zu diesem Zweck müssen sie untereinander kommunizieren. Diese “im Geheimen” (also ohne daß der Benutzer etwas merkt) ausgetauschten Kontrolldaten nennt man gemeinhin MAC-PDUs (“Protocol Data Units”). Sie stellen einen Overhead dar, da sie Netzkapazität verbrauchen, aber keine Nutzdaten transportieren. Verteilt gesteuerte Mechanismen sind aber im Allgemeinen wesentlich besser berechenbar als wahlfreie Verfahren. Ober- und Untergrenzen für bestimmte Systemereignisse sind eindeutig bestimmbar. Diese Verfahren sind wesentlich besser für die Übertragung von Echtzeitdaten wie zB Sprache, Audio, Video etc geeignet.
Zentral gesteuerter Zugriff	Bei diesen Verfahren gibt es eine spezielle Station, die den Zugriff auf das Medium koordiniert. Diese spezielle zentrale Station ist natürlich besonders abzusichern, da sie einen “Single Point of Failure” darstellt. Aus diesen Gründen werden zentral gesteuerte Zugriffe bei LANs selten eingesetzt. Die meisten verteilt gesteuerten MACs haben allerdings eine oder mehrere Stationen, die das LAN monitoren. Diese geben auch einem verteilt gesteuerten MAC Charakterzüge eines zentral gesteuerten MACs. Reine zentral gesteuerte Systeme sind zB Daisy Chains, Selektions- und Polling-Verfahren. Sie werden

	häufig bei Bussystemen (zB SCSI, USB; fire Wire) eingesetzt.

Tabelle Klassen von MAC Verfahren

Last und Durchsatz

Im folgenden Diagramm wird die Last, die auf ein LAN einwirkt, und der daraus resultierende Durchsatz dargestellt:



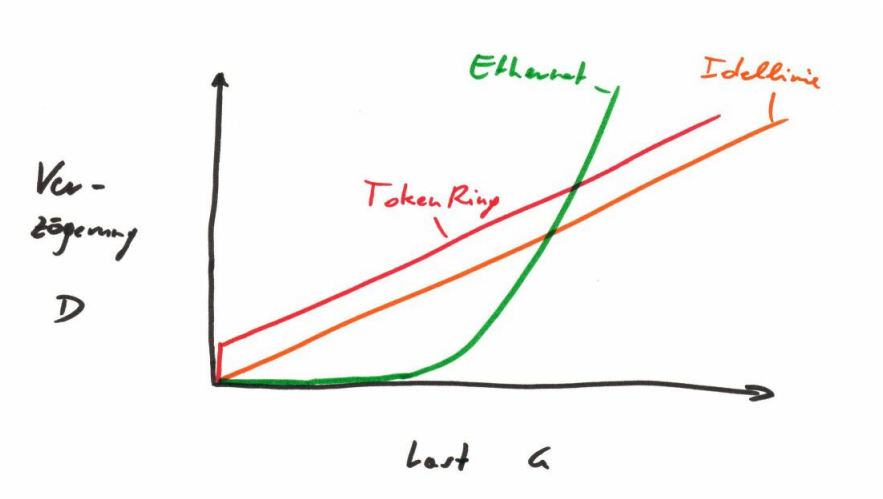
Skizze Last zu Durchsatz

Man sieht die Idealkurve, bei der der Durchsatz der Last linear eins-zu-eins folgt um dann bei 100% stabil zu bleiben. Mehr Last kann klarerweise nicht mehr als 100% Durchsatz erzeugen. Aus einem mit 10Mbps getakteten MAC kann man nicht 12Mbps Leistung herausholen.

In schlimmen Fällen führt mehr Last sogar zu weniger Durchsatz. Dies muß klarerweise vom MAC verhindert werden. Sowohl Ethernet als auch Token Ring nähern sich der Idealkurve, erreichen sie aber nie ganz. Bei Ethernet sind es die Kollisionen, die Kapazität kosten, beim Token Ring ist es das Protokoll, da Datenpakete zur Zugriffssteuerung versendet werden müssen.

Last und Verzögerung

Wenn man die Verzögerung dem Durchsatz gegenüberstellt, sieht man die etwas hinterhältige Reaktion von wahlfreien MACs wie zB Ethernet. Hier ist in einem “ruhigen” System die Verzögerung beeindruckend kurz, um dafür aber beim Übergang in die Sättigung dramatisch anzusteigen. Ein verteilt gesteuerter MAC wie zB der des Token Ring hat auch im Ruhezustand eine Grundverzögerung, da eine Station immer auf das Token warten muß, bevor sie senden darf. Dafür steigt aber die Verzögerung mit der Ringlast linear an.

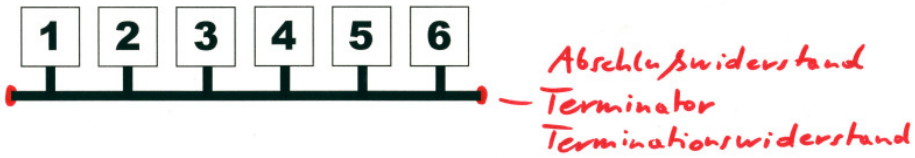


Skizze Last und Verzögerung

Bei Ethernet gilt nach wie vor die Grundregel, daß ein im Bereich 40-50% Last laufendes Ethernet in den Sättigungsbereich kommt, in dem die Verzögerung dann rasch anwächst und der Durchsatz praktisch gleichbleibt. Dieses Verhalten tritt auf überbelasteten Segmenten auf, also wenn sich zu viele Stationen ein Segment teilen müssen. In diesem Falle ist die Auftrennung des Segments in Untersegmente (zB per Bridge, Switch oder Router) angebracht. Durch die geringere Zahl an Stationen (und die damit verringerte Anzahl von Kollisionen) sind die einzelnen Segmente dann weniger belastet.

Kollision

Unter einer Kollision versteht man das “Zusammenprallen” zweier oder mehrerer Datenpakete auf dem Medium. Dabei werden alle beteiligten Pakete unrettbar zerstört. Kollisionen treten im Allgemeinen nur bei Contention MACs auf, können aber in besonderen Situationen auch bei anderen MACs vorkommen.

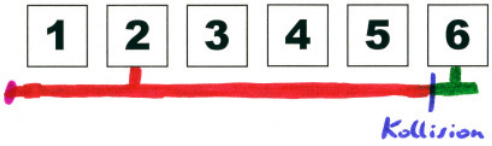


Segment mit 6 Stationen, busförmiger Verkabelung und Abschlußwiderständen

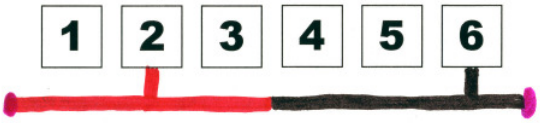
Station 2 sendet ein Datenpaket aus. Das Paket wandert bidirektional im Kabel Richtung Endwiederstand.

Das Paket wird "links" vom Abschlußwiderstand "vernichtet". Die Stationen 1 und 3 haben das Paket bereits erkannt und beginnen mit dem Empfangen.

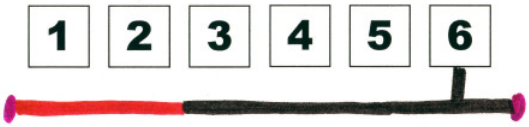
Die Stationen 4 und 5 haben das Paket nun auch erkannt und beginnen ebenfalls, dessen Daten in den Eingangspuffer zu kopieren. Kurz bevor das Paket die Station 6 erreicht, beginnt die Station 6 zu senden. Dies darf sie, da sie ja (noch) keine Daten auf dem Medium vorgefunden hat.



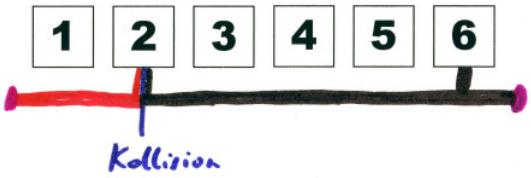
Das Datenpaket der Station 6 kollidiert mit dem Datenpaket der Station 2. Keine einziger der 6 Stationen dieses Segments erkennt zu diesem Zeitpunkt die Kollision.



Nun mischen sich die Datenpakete der Stationen 2 und 6, indem sie miteinander "verschmelzen" (und sich dabei zerstören). Die Stationen 6, 5 und 4 haben die Kollision bereits erkannt.



Die Station 2 hat die Kollision immer noch nicht erkannt und sendet weiter.



Nun hat die Kollisionsfront auch die Station 2 erreicht und führt dort zum Abbrechen des Sendevorganges.

Die Kollisionsdaten "füllen" das gesamte Segment, alle Stationen wissen nun von der Kollision. Die Kollisionsdaten werden ebenfalls von den Abschlußwiderständen eliminiert.

Skizze Kollision in einem busförmigen LAN Segment.

Das Ziel aller Contention MACs ist es, Kollisionen zu vermeiden oder zumindest deren Folgen zu mildern. Da man bei Contention MACs Kollisionen nicht ausschließen kann (außer wenn zB ein Segment nur mit einer einzigen Station belegt wird – dies ist eine präferierte Methode, die Anzahl der Kollisionen in einem LAN zu verringern), muß man versuchen, Kollision frühzeitig zu erkennen und alle Stationen darüber informieren. Daher gibt es den Begriff des “Collision Windows”, oft auch die “Collision Domain” genannt. Das ist diejenige Zeitspanne, innerhalb derer in einem Netz alle beteiligten Stationen eine Kollision erkannt haben müssen (“twice the maximum network end-to-end propagation delay”). Das Collision Window ist errechenbar aus:

CW = 2 * ü * d / v

ü = Übertragungsgeschwindigkeit auf dem Medium in bps
d = Distanz zwischen den am weitesten auseinanderliegenden Knoten des Netzes in m
v = Mediumlichtgeschwindigkeit in m/sec, typisch 0,6-0,7c
2 = der “round trip”, also Weg hin und zurück, muß berücksichtigt werden
CW wird in Bit angegeben

Je schneller man daher sendet oder je größer die Distanz wird, desto größer wird das Collision Window, desto “später” (in Bits gerechnet) erkennen die Stationen, daß eine Kollision aufgetreten ist. Definitionsgemäß wird das Collision Window fix vorgegeben, daraus errechnen sich dann "ü" und "d" für ein LAN.

Wählt man nun das Collision Window groß, so kann es passieren, daß von einem Datenpaket bereits viele Bits gesendet wurden, und dann eine Kollision eintritt (Die Wahrscheinlichkeit einer Kollision steigt mit der Anzahl der gesendeten Bits). Damit ist das Paket zerstört und muß als ganzes nochmals gesendet werden. Es wird viel Netzkapazität durch häufige Kollisionen verschwendet, auch die Verzögerung (Latenz) nimmt zu.

Alternativ kann man das Collision Window klein wählen. Dann werden die Kollisionen sehr früh erkannnt und treten auch wesentlich seltener auf, aber auch "d" oder "ü" oder beide müssen dann entsprechend klein gewählt werden. Das LAN wird daher entweder langsam oder seine Gesamterstreckung wird klein.

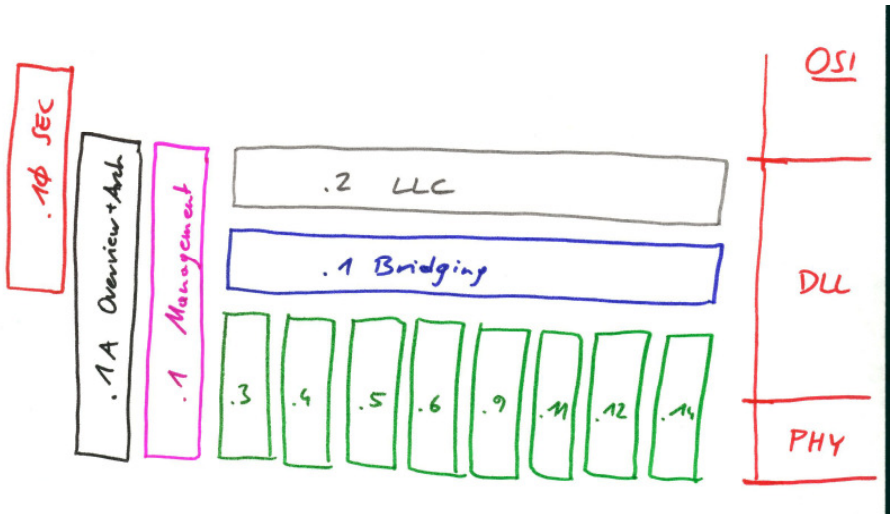
Bei Ethernet verwendet man als Collision Window seit jeher 512 Bit (also ein klein CW), bei den Gbps-Varianten 4096 Bit (512 Byte, also das achtfache). Generell gilt im Ethernet, daß eine Kollision zweier oder mehrerer Datenpakete nur innerhalb des Collision Windows auftreten darf. Das ist eine wichtige Implementierungsvorgabe für dieses LAN! Ist das Window "vorbei" (also 512 Bits bzw 512 Byte des Paketes wurden erfolgreich gesendet), kann die Station den Rest des Pakets senden, ohne auf weitere Kollisionen achten zu müssen, da nun definitionsgemäß keine Kollision mehr passieren kann. Die ersten 512 Bit / Byte sind also die kritische Phase, nur in dieser können Pakete und damit Bandbreite vernichtet werden! Ist ein Kabelsegment des Ethernet zu lang, treten Kollisionen nach den ersten 512 Bit des Paketes auf – diese werden dann von den beteiligten Ethernet-NICs nicht mehr erkannt und die Korrektur dieses Problems muß auf der nächsten Schicht (LLC oder L3, L4, etc) erfolgen! Aber: je höher die Schicht, desto ineffizienter wird die Fehlerbehandlung (MAC: Collision Detection erfolgt "in Hardware". L3 - zB IP: dieser Fehler wird überhaupt ignoriert. L4: - zB TCP "resend": alles "in Software" realisiert).

Es gilt bei den mit Kollisionen arbeitenden MACs ähnlich wie bei der Mediumkennlinie: das Produkt aus Geschwindigkeit und Entfernung ist konstant. Je schneller das Contention LAN, desto kleiner seine maximale Erstreckung. Wenn bei einem Collision Window von 512 Bit ein 10Mbps Ethernet noch 2,5km durchmessen konnte, ist ein 100Mbps Ethernet auf 200m eingeschränkt. Ein 1-Gbps Ethernet wäre auf 20m Durchmesser reduziert, was allerdings für ein LAN inpraktikabel ist.

Daher wurde bei den Gbps-Ethernet-Varianten das Collision Window auf 4096 Bits aufgestockt, was wieder knapp 200m Netzdurchmesser ergibt. Dafür beträgt die minimale Paketgröße in so einem Netz jetzt 512 Byte, was für manche Anwendungen (zB telnet) einen bedeutenden Overhead darstellt! Als Ausgleich wurde der "Packet Burst Mode" eingeführt, bei dem eine Station solange senden darf, wie sie sendebereite Pakete in der lokalen Warteschlange liegen hat. Der Burst ist auf 64 Kb beschränkt, dann muß die Station aufhören zu senden.

IEEE – Institute of Electrical and Electronics Engineers

Das IEEE – sprich "ai-trippl-i" – ist das älteste Normungsgremium, das sich mit Lokalen Netzen beschäftigt hat. Im IEEE wurden und werden nach wie vor eine Vielzahl von Normen erstellt. Die IEEE-Untergruppe 802 beschäftigte sich in den achtziger Jahren mit der Normung von LANs, die ersten effektiven LAN-Normen kamen 1985 heraus: IEEE 802.3 CSMA/CD (aka Ethernet), IEEE 802.4 Token Bus und IEEE 802.5 Token Ring. Ferner wurde mit 802.2 die gemeinsame Schnittstelle der LANs zu den oberen Schichten festgelegt und mit 802.1 die Integration in das damals bereits weitgehend fertiggenormte ISO-OSI-Modell ("International Standards Organisation" - "Open Systems Interconnect") durchgeführt. Die meisten Normen der IEEE 802.x wurden nachträglich auch zu ISO-OSI-Normen. Sie erhielten die Bezeichnung ISO 8802.x, wobei das x der IEEE-Norm dem x der OSI 8802-Norm entspricht.



Skizze IEEE 802.x und OSI

IEEE 802 nennt sich seit einigen Jahren "LMSC" – "LAN MAN Standards Committee".

IEEE 802 Aufbau

Die IEEE befaßt sich seit 1980 mit der Erstellung von Standards auch im LAN-Bereich. Ihre wichtigsten Standards kamen 1985 konsolidiert heraus und umfaßten:

IEEE 802	Overview and Architecture
IEEE 802.1	Architecture, Management and Internetworking
IEEE 802.2	Logical Link Control
IEEE 802.3	CSMA/CD
IEEE 802.4	Token Bus

IEEE 802.5	Token Ring

Tabelle ursprüngliche IEEE 802.x Standards

Im Laufe der Zeit kamen weitere Standard-Gruppen hinzu. Heute sieht das "Gebäude" der IEEE 802.x folgend aus:

802.1	Higher Layer LAN Protocols Working Group. The IEEE 802.1 Working Group is chartered to concern itself with and develop standards and recommended practices in the following areas: 802 LAN/MAN architecture, internetworking among 802 LANs, MANs and other wide area networks, 802 overall network management, and protocol layers above the MAC & LLC layers. Untergruppen der .1 Gruppe sind zB: Abgeschlossen: 802.1D MAC bridges 802.1G Remote MAC bridging 802.1Q Virtual LANs Aktiv: 802.1s Multiple Spanning Trees 802.1t 802.1D Maintenance 802.1u 802.1Q Maintenance 802.1v VLAN Classification by Protocol and Port 802.1w Rapid Reconfiguration of Spanning Tree 802.1x Port Based Network Access Control
802.2	Logical Link Control Working Group (Inactive). The IEEE 802.2 Working Group develops standards for Logical Link Control. This working group is currently in hibernation (inactive) with no ongoing projects.
802.3	Ethernet Working Group. Abgeschlossen: 802.3 Trunking Study Group 802.3 Higher Speed Study Group. 802.3 DTE Power via MDI Study Group. 802.3z-1998, Gigabit Ethernet 802.3aa-1998, Maintenance Revision #5 (100BASE-T) 802.3ab-1999, 1000BASE-T 802.3ac-1998, VLAN TAG 802.3ad-2000, Link Aggregation 802.3 Ethernet over LAPS liaison Ad hoc. Aktiv: P802.3, Ethernet in the First Mile Study Group 802.3ae, 10Gb/s Ethernet 802.3af, DTE Power via MDI 802.3ag, Maintenance Revisions #6. 802.3rev, Conformance Test Maintenance #1 802.3 Static Discharge in Copper Cables Ad hoc.
802.4	Token Bus Working Group (Inactive). The IEEE 802.4 Working Group develops standards for Token Bus. This working group is currently in hibernation (inactive) with no ongoing projects.
802.5	Token Ring Working Group.
802.6	Metropolitan Area Network Working Group (Inactive). The IEEE 802.6 Working Group develops standards for Metropolitan Networks. This working group is currently in hibernation (inactive) with no ongoing projects.
802.7	Broadband TAG (Inactive). The IEEE 802.7 Broadband Technical Advisory Group (TAG) developed a recommended practice, IEEE Std 802.7 - 1989, IEEE Recommended Practices for Broadband Local Area Networks (Reaffirmed 1997). This group is inactive with no ongoing projects. The maintenance effort for IEEE Std 802.7 - 1989 is supported by 802.14.
802.8	Fiber Optic TAG. The IEEE 802.8 Technical Advisory Group (TAG) develops recommended practices for fiber optics. Letzter Eintrag 1997.
802.9	Isochronous LAN Working Group. The IEEE 802.9 Working Group develops standards for Isochronous LANs. Letzter Eintrag 1997.

802.10	Security Working Group. The SILS working group has completed its work in providing standards for LAN/MAN security. The group is in "hibernation" but can still be contacted for assistance using the links listed on this page.
802.11	Wireless LAN Working Group. The IEEE 802.12 Working Group develops standards for Demand Priority. Letzter Eintrag 1997. Derzeit sehr heißes Thema.
802.12	Demand Priority Working Group. Letzter Eintrag 1997.
802.13	Not Used
802.14	Cable Modem Working Group. In hibernation (?)
802.15	Wireless Personal Area Network (WPAN) Working Group. Inkludiert Bluetooth, "High Rate WPAN" und "Low Rate WPAN"
802.16	Broadband Wireless Access Working Group. Welcome to the Web Site of the IEEE 802.16 Working Group on Broadband Wireless Access Standards. The mission of Working Group 802.16 is "to develop standards and recommended practices to support the development and deployment of fixed broadband wireless access systems." Hiermit ist im Speziellen ein Wireless MAN gemeint, also der Zugang in ein stadtweites oder sogar überregionals Breitband-Netzwerk im 10 - 60 GHz Bereich.
802.17	Resilient Packet Ring Working Group. Dual Counter Rotating Rings, both rings carry traffic all of the time; Media Independence, scalable in bit- rate, # nodes, span distance; OC-48c & OC-192c SONET/ SDH as PHY or 1Gbps & 10 Gbps Ethernet as PHY. Der Standard-Vorschlag ist eine Alternative zu sowohl GB-Ethernet als auch SONET. Sehr junge Untergruppe der 802.

Tabelle IEEE 802.x mit ihren jeweiligen "Mission Statements"; Stand 2001

IEEE 802 Adressen

Im IEEE 802.x wurde auch der Aufbau von Adressen genormt. Eine IEEE 802.x oder MAC Adresse ist immer 16 oder 48 Bit lang.

Die 16-Bit Adreßform wurde kaum je eingesetzt, da sie den Nachteil habt, daß man bei jeder Station eine eigene eindeutige Adresse explizit einstellen muß. Dies macht die 16-Bit Adreßform aus Sicht des Netzwerkadministrators unattraktiv. Ferner ist es aus Performance-Sicht praktisch unerheblich, ob 16 oder 48 Bit für die Adressen verwendet werden.

Die 48-Bit Adreßform hat noch einen zusätzlichen Vorteil: jede Karte hat vom Werk aus eine eindeutige individuelle MAC-Adresse “einprogrammiert”. Diese fixe MAC-Adresse ist weder änderbar noch löschbar.

Jede 48-Bit-MAC-Adresse hat den folgenden bitweisen Aufbau:

```
ccccccug cccccccc cccccccc xxxxxxxx xxxxxxxx xxxxxxxx
```

Die “c”-Bits sind der Prefix, der für jeden Hersteller eindeutig vorgegeben wird (sogenannter “Organization Unique Identifier”). Sämtliche von einem Hersteller prouzierten NICs haben denselben Prefix.

Das “u”-Bit ist das “Universally Defined / Locally Defined” (“U/L”) Bit. Bei den fest vorgegebenen MAC Adressen ist es immer auf “0” gesetzt.

Das “g”-Bit ist das “Individual Address / Group Address” (“I/G”) Bit. Ist es auf “1” gesetzt, liegt eine Gruppenadresse (Multicast Adresse) vor. Ansonsten ist diese MAC-Adresse eine Individuelle (Unicast) Adresse.

Die “x”-Bits sind vom Hersteller der Karte eindeutig zu vergeben und werden normalerweise aufsteigend nummeriert.

Normalerweise verwendet man in IEEE 802.x Netzen “u=0” und “g=0”, also individuelle unicast MAC Adressen, da dies den geringesten Aufwand in der Konfiguration der NICs bedeutet. Beim IEEE 802.5 Token Ring wird die lokal administrierte Variante bevorzugt. Dafür wird “u=1” und “g=0” gesetzt und die restliche Stationsadresse (MAC-Adresse) dann frei vergeben. Wird zusätzlich mittels Source-Routing Bridges gearbeitet, muß bei der Festlegung der Stationsadresse die folgenden Konvention eingehalten werden:

```
rrrrrrug rrrrrrrr xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

Die 14 “r”-Bits kennzeichnen nun die Ringnummern der Station. Jeder einzelne Ring eines gebridgten Token Ring Systems muß eine eindeutige eigene Ring-Nummer erhalten. Die “x”-Bits sind frei vergebbar, müssen aber innerhalb eines Rings eindeutig sein (das wird für jede neu in den Ringe kommende Station geprüft, siehe Token Ring).

Ferner ist es möglich und oft auch der Fall, daß eine Station mehrere MAC-Adressen zugewiesen erhält. Zusätzlich zur vorgegebenen individuellen Adresse werden zusätzliche, sogenannte funktionale Adressen vergeben. Diese werden – wie der Name bereits sagt – aufgrund von speziellen Funktionen der jeweiligen Station zugeteilt. Im IEEE 802.5 werden diese Funktionalen Adressen folgend gebildet:

```
rrrrrrug rrrrrrrr bxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

Die “r”-Bits geben wieder wie gewohnt den Ring an. Es wird aber nun “u=1” und “g=1” gesetzt. Damit wird eine lokal verwaltetet Multicast-Adresse gebildet. Diese ist entweder bit-Signifikant (“b” = 1) oder nicht. Bei einer bit-signifikanten Adresse steht jedes der 31 “x”-Bits für eine von 31 verschiedenen Funktionen. Vordefiniert sind hierbei die folgenden Werte für “x”:

```
00:00:00:01    aktiver Ring Monitor
00:00:00:02    Ring Parameter Server
00:00:00:08    Fehler-Monitor
00:00:01:00    Bridge
…
00:08:00:00    und größer: benutzerdefinierte Funktion
```

Bei der nicht-bitsignifikanten Version (“b” = 0) steht jeder Zahlenwert von “x” für eine Funktion. Damit sind potentiell 2^31 Funktionen ansprechbar. Wenn der Wert von “r” in der DA gleich “0” ist, dann ist immer “dieser Ring” (also der lokale Ring) gemeint.

IEEE 802.3 CSMA/CD – Ethernet

Ethernet ist das “Ur-LAN”, das älteste definierte LAN-System, das noch in Betrieb befindlich ist und zugleich den Markt der LANs mit über 86% Marktanteil (Stand 1998) dominiert. Die Stärken des Ethernets liegen vor allem in seiner Einfachheit, die über die letzten 20 Jahre unverändert blieb. Erst mit der Einführung der Gbps Ethernet LANs 1998 begann man an den “Grundfesten” des Ethernet (512 Bit Collision Window, CSMA/CD-MAC-Verfahren) zu rütteln.

Designziele für das Ur-Ethernet (1973):
Einfachheit
geringe Kosten
Kompatibilität
Fairness im Zugriff der Stationen auf das Medium

“non-repudiation” (Nichtverweigerung des Dienstes)
geringe Verzögerungszeiten bei kleiner Last
Stabilität des Durchsatzes (monoton steigend mit der Last)

Tabelle Designziele des ursprünglichen Ethernet

Die ursprüngliche Einfachheit des Systems und seine zugleich ungeahnte Skalierbarkeit hat den Siegeszug des Ethernet bewirkt. Aus dem anfänglich simplen, busbasierten 10Mbps LAN, das weder Fehlertoleranz noch Echtzeitfähigkeit noch Management-Werkzeuge kannte, wurde das marktbeherrschende LAN-System, das durch Einfachheit und damit günstigen Preis und durch estmögliche Ausreizung seiner Möglichkeiten alle anderen Systeme – obwohl diese zumeist wesentlich besser sind – dominiert.

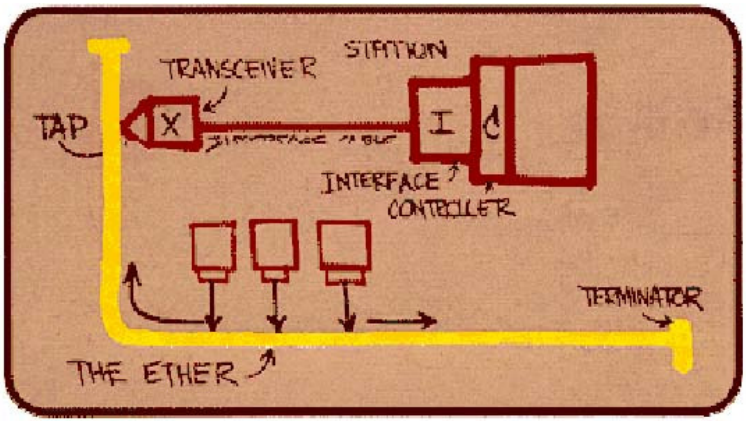
Ethernet-NICs sind heutzutage in Einzelstücken ab ca 8 US$ zu haben und zählen mittlerweile zu den billigsten Komponenten in PC-Gesamtsystemen. Dies wurde durch die Schlichtheit des Systems und die darausfolgende “Economy of Scale” in beeindruckender Art und Weise erreicht und von keinem anderen LAN-System bisher nachvollzogen.

Ethernet hat heute alle anderen LAN-Systeme zu Nischenprodukten degradiert und punktet auch in Bereichen, in denen es eigentlich markante Schwächen aufweist, zB bei der Übertragung von Echtzeitdaten. Sein Contention-MAC ist immer noch derselbe wie vor 20 Jahren und daher immer noch nicht echtzeitfähig. Die Lösung dieses Problems lautet im Original “throw bandwidth at delay”, man versucht also, die unangenehmen statistischen Verzögerungs-Effekte des Ethernet-MACs nicht zur Wirkung kommen zu lassen, indem man weit unterhalb des Sättigungsbereichs bleibt. Das klappt aber nur, wenn die Bandbreite und damit der maximal mögliche Durchsatz des Systems sehr hoch sind. Und genau diesen Weg geht Ethernet heute, indem die 10Gbps für das Jahr 2002 angekündigt wurden.

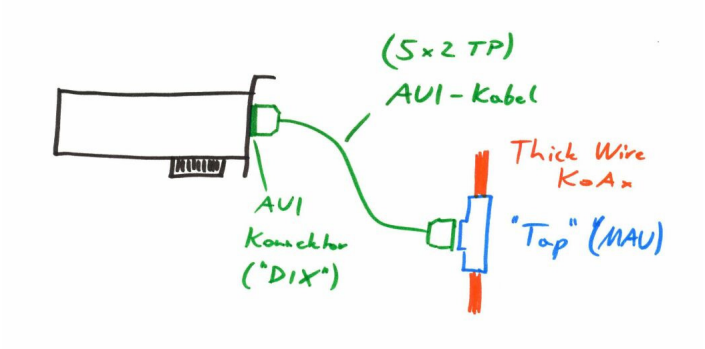
Ethernet ist damit drauf und dran, auch MAN- und WAN-Protokolle wie Frame Relay und ATM vom Markt zu drängen, obwohl diese technisch wesentlich leistungsfähiger und moderner sind. Es könnte sein, daß es in einigen Jahren nur noch ein einziges Netzwerk weltweit gibt, eine Art “Welt-Ethernet”, das zugleich die LANs der einzelnen Subnetze darstellt und auch den Internet Backbone umfaßt.

Standard-Ethernet

Aus dem Jahre 1979 stammend ist Ethernet ein – in EDV-Zeiteinheiten gemessen – Uralt-System aus der Frühzeit der EDV. Es wurde von den Firmen Digital, Intel und Xerox gefördert und initial unterstützt. Einer der verwendeten Stecker trägt immer noch den Namen “DIX-Connector” (DIX für “Digital-Intel-Xerox”).



Ursprünglich als busbasiertes System mit KoAx-Verkabelung und 10Mbps (für damalige Verhältnisse recht schnell) konzipiert, folgte rasch eine “Billigversion”, da das Original für ein LAN zu teuer in der Installation war und man die Verbreitung aufgrund zu hoher Kosten gefährdet sah. Das Ur-Ethernet oder korrekter: “Standard-Ethernet” verwendete “thick wire” KoAx (RG-59A) mit Segmenten bis zu 500m, bis zu 100 Stationen pro Segment und schloß die NICs per “Drop Cable” an das KoAx-Kabel an:

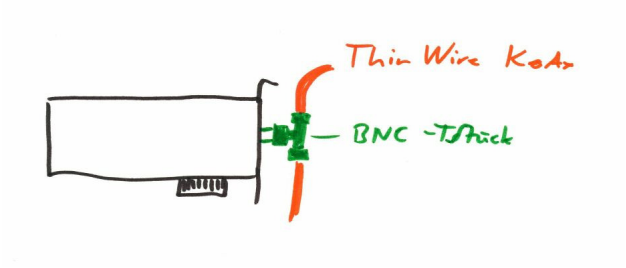


Skizze Standard Ethernet

Das eigentliche Problem war das Drop Cable, auch “Transceiver Cable” oder im IEEE-Standard dann “Attachment Unit Interface Cable” (“AUI-Cable”) genannt. Dieses war – gegenüber dem KoAx-Kabel des Segments – ein 10-poliges UTP-Kabel, das bis zu 50m lang sein durfte, und mußte daher von hoher Güte sein. Dies war damals in den Achzigern nicht einfach realisierbar und daher sehr teuer. Und alles, was teuer war, wurde im Ethernet möglichst rasch umgangen. Daher fand sich bald der neue Standard “Cheapernet” (auch “thin wire Ethernet” genannt) ein.

Cheapernet

Beim Cheapernet wurde ein billigeres KoAx-Kabel vorgeschrieben (RG-58A). Dieses muß zudem in Schleifen zu allen NICs geführt werden, was die Verwendung der teuren AUI-Kabel vermied, allerdings viel Kabel verbrauchte. Ein weiterer Effekt des billigen Kabels waren nur noch 185m maximale Segmentlänge und nur noch maximal 30 Stationen pro Segment.

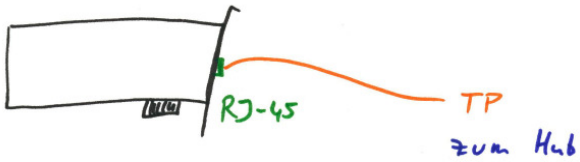


Skizze Cheapernet

Cheapernet wurde dennoch häufig eingesetzt und bereitete den eigentlichen Siegeszug des Ethernet vor. Allerdings war es immer noch sehr fehleranfällig und bei einfachen Leitungsausfällen stand das gesamte LAN still. Der Grund dafür liegt in den elektrischen Eigenschaften der Bus-Topologie. Da ein Bus zwei "offene Enden" hat, müssen diese terminiert werden. Eine durchtrennte Leitung hat aber zwei "zusätzliche" Enden, die nicht terminiert sind und beide Teilstücke funktionieren daher nicht mehr zuverlässig.

Durch die Reduktion der Stationen pro KoAx-Segment von 100 (bei Standard-Ethernet) bzw 30 (bei Cheapernet) auf eine einzige wurde das Problem des Leitungsausfalls zwar nicht für die einzelne betroffene Station behoben, aber für das gesamte LAN sehr wohl. Es entstand das sternförmige (hierarchische) Ethernet. Im Gegensatz zB zum Doppelring, der einen Leitungsausfall unbeschadet übersteht, ist beim sternförmigen Ethernet bei einem Leitungsausfall ein Netz-Abschnitt weggetrennt und damit nicht mehr erreichbar. Dennoch setzte sich die neue Topologie beim Ethernet praktisch sofort durch. Damit erhielt aber ein Gerät, das bis dato nur zur Leitungs-Verlängerung von LANs eingesetzt wurde, eine zentrale Bedeutung: der "Hub".

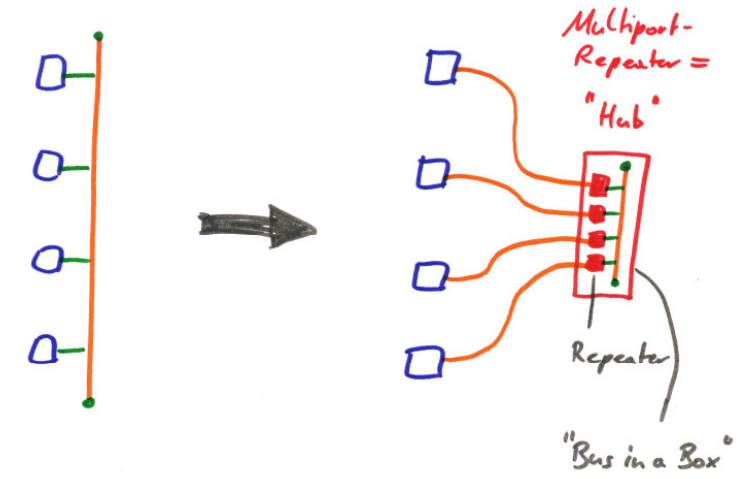
Twisted Pair Ethernet



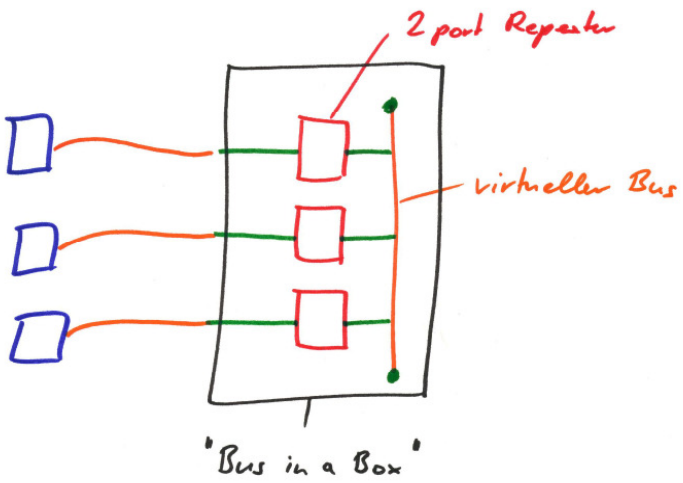
Skizze TP-Ethernet.

Der Hub ist in Wirklichkeit ein Multiport-Repeater. Da im sternförmigen Ethernet pro Segment nur noch eine Station angeschlossen werden darf, muß am anderen Ende des Kabels ein Repeater liegen. Dieser hat die Aufgabe, bei Leitungsausfall das Netz zu rekonfigurieren, einfach gesprochen den betroffenen Port abzuschalten, zu terminieren und ein Lämpchen ("link down" oder ähnlich) aufleuchten zu lassen. Ansonsten funktioniert er wie jeder andere Repeater auch, ist also ein bidirektionaler bitweiser Transmitter/Receiver, aber eben mit mehreren "Ports" (das sind die Stecker für die Kabel, die zu den Stationen oder zu anderen Repeatern gehen).

Oft wird ein Repeater auch als "Bus in a Box" bezeichnet, also als eine Art superkurzes KoAx-Kabel, das in einem Gehäuse steckt und an das die einzelnen TP-Segmente per 1-zu-1 Repeater angeschaltet werden.



Skizze Ein Segment eines Bus-LANs wird zu einem "Bus in a Box"



Skizze innerer Aufbau eines "Bus in a Box"

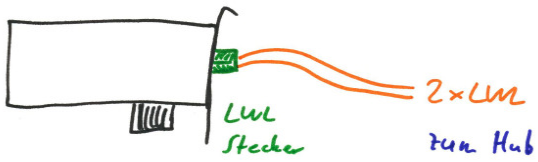
Außerdem erfolgte der Umstieg vom KoAx-Kabel auf TP, da man nun von Seiten der Elektronik in der NIC in der Lage war, 10Mbps auf einem billigen Cat-3 TP zu übertragen. Da TP auch in der Telefonie verwendet wird, wurde die Umstellung auf TP-Ethernet noch zusätzlich beschleunigt, indem man nun die bereits in den Gebäuden (zumeist sehr großzügig) verlegten Telefonkabel quasi als “bereits fertig verlegte” LAN-Medien verwenden konnte. Mit TP wurde allerdings 100m als maximale Distanz bis zum Repeater festgelegt (anstatt 500m bzw 185 beim KoAx). Zugleich wurde der ungeschirmte, billige RJ-45 Steckverbinder aus der Telefonie übernommen.

Von den 8 Kabeln, die im RJ-45 definiert sind, werden beim TP-Ethernet nur 2x2 Kabel verwendet: Signal und Masse fürs Senden, Signal und Masse fürs Empfangen. Die Kabel Nummern 1 & 2 und 3 & 6 werden benutzt, das mittlere Paar 4 & 5 (das immer bei RJ-45 für die Telefonie verwendet wird) bleibt frei.

LWL Ethernet

Mit 10BASEF wurden 1993 auch LWLs für den Einsatz im Ethernet freigegeben. Ethernet über LWL entstand aus den “Fiber Optic Inter Repeater Links” (FOIRL, ca 1987), das waren proprietäre

Repeater-zu-Repeater Verbindungen auf LWL Basis. FOIRLs waren für die Überbrückung großer Distanzen (bis 2km) konzipiert und daher immer Full Duplex betrieben. Diese Technik nahm nun ebenfalls in das Ethernet Einzug.



Skizze Ethernet auf LWL Basis

Fast Ethernet auf TP Cat-5

Im nächsten Schritt erfolgte die Umstellung von 10Mbps auf 100Mbps bei gleichzeitiger Übernahme aller sonstigen Eigenschaften des 10Mbps TP-Ethernets. Es entstand das sogenannte "Fast Ethernet". Für die Übertragung sind lediglich bessere (mindestens Cat-5) TP-Kabel notwendig. Ferner wird mit 4B/5B vorkodiert, damit wird der Overhead der Übertragung auf 25% reduziert. Dennoch liegt das 100Mbps Datensignal durch die 25% Overhead für 4B/5B brutto bei 125 Mbps, also über der Spezifikation des Cat-5 mit 100Mbps. Daher wird mithilfe einer Kodierung namens "MLT-3" ("Multi Level Transition with 3 Levels") ein trinäres Signal ("Eins", "Null" und "Minus Eins") zur Übertragung verwendet.

Bei TP-Ethernet und LWL-Ethernet gibt es immer getrennte Leitungen zum Senden und Empfangen, man kann daher in bestimmten Situationen komplett auf das CSMA/CD verzichten und im sogenannten "Full Duplex Mode" arbeiten. Mit dem Fast Ethernet wurde auch der "Full Duplex MAC" eingeführt. Dieser Modus ist nur solange verfügbar, wie sichergestellt ist, daß es zu keinen Kollisionen kommen kann. Repeater sind natürliche Grenzen für Full Duplex Segmente, da bei Repeatern immer mehrere Ports Daten senden können und ein Repeater die Daten an alle Ports weitersenden muß. Full Duplex spielt aber bei gebridgten/geswitchten Netzen (siehe Internetworking) und bei reinen Point-to-Point Verbindungen auf Ethernet-Technologie eine Rolle und verdoppelt die Übertragungsrate.

Dadurch, daß auf Full Duplex Strecken kein CSMA/CD eingesetzt wird, gibt es dort auch keine Collision Windows und damit keine Längenbegrenzung aus der Regel CW = 2 * ü * d / v mehr. Nur noch die Qualität des Mediums entscheidet über die Streckenlänge! Im Halb Duplex Betrieb beträgt die maximale Entfernung der äußersten Netz-Knoten 200m, die Enfernung von der Station zum Repeater 100m. Es darf höchstens ein Repeater zwischen zwei Stationen liegen, bei den schnelleren Klasse-2 Repeatern dürfen es zwei sein, die über ein maximal 5m langes TP verbunden sind.

Fast Ethernet auf TP Cat-3

Bei dieser ersten Sonderform über Cat-3 - TP genannt 100BASET4 - werden drei Kabelpaare zugleich zum Senden und das vierte für die Kollisionserkennung verwendet. Diese Ethernet Variante über alle 4 Kabelpaare eines Cat-3 TPs hat sich nicht durchgesetzt, obwohl es mit den "schlechteren" Cat-3 Kabeln aus der Telefonie auskommt. 100BASET4 beherrscht außerdem den Full Duplex Modus nicht. Fast alle Kunden hatten bereits Cat-5 verlegt oder planten den Umstieg auf Cat-5, um dann 100BASETX zu verwenden.

Genauso erging es dem Nachfolger 100BASET2. Er braucht nur zwei Kabelpaare im TP und nur Cat-3 TP und kennt Full Duplex Betrieb. Er wurde dennoch – wie auch 100BASET4 – von 100BASETX ersetzt. 100BASET2 verwendet eine quintäre Kodierung namens "Pulse Amplitude

Modulation - 5 Level" ("PAM5"). Die 5 Levels werden bezeichnet mit: -2, -1, 0, 1, 2 und kann damit über 2 Bits pro Taktschritt übertragen. Offene Frage: warum kann man auf einem Cat-3 mit 16 MHz * 5 Levels insgesamt 100 Mbps übertragen?

Beide Fast Ethernet Varianten sind insofern abwärtskompatibel zum TP-Ethernet, als sie vor dem Einfügen in den Ethernet-Hub die verfügbare Geschwindigkeit des Hubs testen. Ist der Hub mit 100Mbps betrieben, so hängen sich die NICs mit 100Mbps ein. Ist er mit 10Mbps betrieben, so schalten sich die NICs auf 10Mbps (auf "normales" TP-Ethernet) zurück. Dieser Vorgang wird "Autonegotiation" genannt. Er funktioniert nur nur im TP-Ethernet und nur zwischen 10 und 100 Mbps.

Fast Ethernet auf LWL

Ferner gibt es noch das Fast Ethernet auf LWL-Basis. Es entstand aus den Erfahrungen mit der Übertragungstechnik im FDDI und verwendet ein ähnliches elektrisches Interface.

Es gilt die 2-1-Regel bzw die 3-2-Regel für Router. Die maximale Segmentlänge darf bei Klasse-1-Repeatern 272m, bei Klasse-2-Repeatern 320m und bei 2 Klasse-2-Repeatern 228m betragen. Als maximale Entfernung zwischen zwei beliebigen Knoten im gerouteten LAN sind 412m definiert.

Fast Ethernet auf LWL unterstützt keine Autonegotiation und kann daher nur mit 100Mbps Internetworking-Geräten zusammenarbeiten.

Gigabit Ethernet

Bei 1000Mbps wäre das Collision Window des Ethernet wieder 1/10 desjenigen von 100BASEx, und damit 20m. Da dies aber selbst für kleine LANs viel zu gering ist, wurde das Collision Window auf 512 Byte erhöht. Die "Aufstockung" von zu kurzen Paketen erfolgt aber nicht wie man glauben könnte im "Pad" Feld des Frames, sondern nach der Prüfsumme in einer sogenannten "Packet Extension". Dies ist aus Kompatibilitätsgründen notwendig, um das MAC-Frame Format des Ethernets einzuhalten. Der MAC ist für die Aufstockung des Frames auf 4096 Bit (512 Byte) zuständig, es werden Non Data Symbols dafür verwendet.

Um die Verschwendung von Kapazität bei kleinen Paketen zu minimieren, darf ein Gigabit Ethernet MAC mehrere Pakete sofort hintereinander senden, sofern die Pakete zur Übertragung bereits fertig bereit stehen. Dieser sogenannte "Burst Mode" kann bis 64Kb lang andauern. Damit wurde auch die maximale Paketlänge des Ethernet von bisher 1518 Byte auf 64 Kb vervielfacht.

Als Verkabelung kommt nun primär LWL (MMF, dann 1000BASESX genannt und SMF, dann 1000BASELX genannt) zum Einsatz, wobei viel von der Technologie des "Fibre Channel" übernommen wurde. Die SX-Varianten sind bis zu 500 Metern verwendbar, die LX-Varianten bis 5000m im Full Duplex Verfahren (also ohne CSMA/CD).

Die Vorkodierung nennt sich 8B/10B und ist ein enger Verwandter des 4B/5B, was die Effizienz und die allgemeine Methode betrifft. 8B/10B wurde ursprünglich bei der IBM in den Achtzigern entwickelt und wird auch im FibreChannel verwendet.

Aus Kostengründen wird auch noch Cat-5 TP Kabel unterstützt. Das nennt sich dann 1000BASECX, alle vier Kabelpaare werden verwendet wie beim 100BASET4 (100 Mbps+25% vom 4B/5B, das Ganze mal 4 Kabelpaare ergibt 500Mbps), zusätzlich werden wie bei 100BASET2 mit der 5-Level-Kodierung 2 Bit pro Taktschritt übertragen (ergibt 2x500=1000Mbps), es ist aber trotzdem Full Duplex fähig! Ein komplexer und ultraschneller DSP ("Digital Signal Processor") in der NIC ist erforderlich, um dies alles zu bewerkstelligen.

Das Gigabit Ethernet verwendet denselben Autonegotiation Mechanismus wie das Fast Ethernet, aber sowohl auf TP als auch auf LWLs.

10 Gigabit Ethernet

Im 10Gbps Ethernet wären selbst mit 512 Byte großem Collision Window wieder nur 20 Meter machbar. Da dieses "Über-Ethernet" aber als Backbone des Internets fungieren soll (und damit in den MAN / WAN Bereich vordringt), wurde vollständig auf den Half Duplex Mode und daher auf das CSMA/CD verzichtet.

Da dies das erste Ethernet ohne CSMA/CD-MAC wäre, stellt sich die Frage, warum man es überhaupt noch als "Ethernet" bezeichnet. Es hat mit dem Ur-Ethernet technisch nichts mehr gemein, weder Topologie, noch Medium, noch Kodierung und nun auch nicht mehr den MAC. Nur das Frame-Format ist noch weitgehend erhalten geblieben.

Es wurden drei verschiedene Versionen der Physischen Schicht (PHY) definiert, zwei LAN-konforme mit "shared medium" und eine auf dem OC-192 (also auf einem WAN) basierende. Es wird generell eine 64B/66B Vorkodierung verwendet.

Für den Betrieb mit "privaten" Medien (also als reines LAN) werden drei getrennte (alternative) Lichtwellenlängen vordefiniert: Short mit 850nm, Long mit 1310nm und Extralong mit 1550nm. Daraus ergeben sich die 10GBASE-SR, 10GBASE-LR und 10GBASE-ER Formen des 10GB-Ethernets.

Mit Verwendung von WDM ("Wave Division Multiplexing") werden vier Lichtwellenlängen zugleich verwendet und damit quasi vier getrennte LWLs implementiert. Es werden nur 1310nm Medien verwendet. Der Standard wird wahrscheinlich 10GBASE-LX heißen.

Bei der letzten Variante des physischen Layers als WAN wird sogar die Geschwindigkeit des 10Gigabit-Ethernet auf 9,2942 Gbps herabgesetzt, da die Geschwindigkeit des OC-192 vordefiniert ist (9,9533Gbps) und durch die 64Bit auf 64Bit Vorkodierung noch ein wenig Leistung verloren geht. Diese WAN-Form des 10GB-Ethernets wird in "Long Wave" und "Extralong Wave" verfügbar sein und 10GBASE-LW bzw 10GBASE-EW heißen.

10Gbps Ethernet gibt es generell nur noch im Full Duplex Betrieb, damit sind die überbrückbaren Entfernungen ausschließlich von der Leitungsqualität abhängig und liegen damit im Bereich etlicher Kilometer. Die primäre Verwendung des 10Gbps Ethernet wird damit vorerst im Backbone-Bereich liegen, nicht am Desktop. Der Full Duple Betrieb schließt die Verwendung von Repeatern grundsätzlich aus, da alle Stationen, die an einem Repeater hängen, eine Collision Domain bilden. 10GB Ethernet kann also nur noch entweder geswitcht oder Punkt-zu-Punkt eingesetzt werden.

Wellenlänge (nm)	**LWL-Typ**	**Kerndurch-messer (nm)**	**Max. Distanz (m)**
850	MMF	50	65
1310	MMF	62,5	300
1310	SMF	9	10.000
1550	SMF	9	40.000

Tabelle 10Gbps Ethernet LAN-Medien und erreichbare Entfernungen

Ethernet und IEEE 802.3 CSMA/CD

IEEE griff relativ bald die damal wichtigsten Produkte im LAN-Bereich auf und normierte sie sozusagen “im Nachhinein”. Während die IEEE-Norm “802.5 Token Ring” den IBM Token Ring direkt und praktisch nur bis auf die Namensänderungen übernahm, wurde Ethernet etwas modifiziert zu “IEEE 802.3 CSMA/CD”. Diese Änderungen sind heute (gottseidank) kaum noch relevant und – wichtiger – kaum noch irgendwo bemerkbar. Dennoch sind Ethernet und 802.3 unterschiedlich im Frame-Format, jedoch sind heute beide Frame-Formate von sowohl der Industrie als auch vom IEEE 802.3 unterstützt. In Anbetracht der extremen Anpassung und Wandlung des Ethernet ist das MAC-Frame-Format heutzutage das einzige, was in allen Ethernet-Varianten seit 1979 wirklich gleichgeblieben ist.

Wir verwenden die Begriffe Ethernet und IEEE 802.3 CSMA/CD in dieser Abhandlung immer synonym. Da Ethernet der kürzere Begriff ist, hat er sich besser eingebürgert und wird auch hier zumeist verwendet.

Ethernet		802.3	
Bytes	**Feld**	**Bytes**	**Feld**
8	Preamble	7	Preamble
		1	Start Delimiter (SDEL)
6	Destination Address (DA)	2/6	Destination Address (DA)
6	Source Address (SA)	2/6	Source Address (SA)
2	Type	2	Length
0-1500	User Data (Schicht-3-Daten)	0-1500	User Data (Schicht-3-Daten)
		0-46	“Pad” (Auffüllen auf 512 Bit)
4	Checksum (CCITT CRC 32)	4	Checksum (CCITT CRC 32)

Tabelle Frameformat des Ethernet und des IEEE 802.3

Die 8-Byte Preamble des Ethernet ist mit der 7-Byte-Preamble und dem 1-Byte-SDEL des 802.3 identisch. Sie werden im 802.3 nur schematisch getrennt, um mit anderen Protokollen des 802 (zB dem 802.5) konform zu gehen. Die Präambel ist eine Folge von “10101010…” mit 7*8 = 56 Bit. Der SDEL besteht ebenfalls aus den Bits “10101010”.

Während beim Ethernet Frame ein Type-Feld das verwendete Protokoll der Schicht-3-Daten angibt (zB steht “0x0800” für IP), ist beim 802.3 Frame dort die Länge der Benutzerdaten abgelegt. Um die beiden Frames auseinanderhalten zu können, ist der Inhalt des Type-Felds beim Ethernet immer größer als die maximale Benutzerdatenlänge (definiert wurde genauergenommen 1536 dezimal, entsprechend 0x0600 hex).

Die Source Address bzw Destination Address kann beim 802.3 auch 2 Byte lang sein. Dies muß dann aber für das gesamte LAN gelten und wurde nie praktisch verwendet, da man nicht auf die eindeutigen hardkodierten MAC-Adressen des Ethernet zugreifen kann (diese sind nur für das 6 Byte Adressformat definiert).

IEEE 802.3 definiert explizites “Padding”, also das Auffüllen eines zu kurzen MAC-Frames auf 512 Bit (=64 Byte, das Collision Window des Ethernet):

6 (DA) + 6 (SA) + 2 (Type/Length) + 4 (Checksum) = 18 Byte

Das ist das minimale (und zugleich leere) Paket im IEEE 802.3. Um 64 Bytes lang zu werden, müssen in diesem Fall 46 Bytes Padding hinzugefügt werden. Im Ethernet ist es Aufgabe der Schicht 3 bzw des NIC-Treibers, das Paket auf 512 Bit zu verlängern, falls es zu kurz ist. Interessant ist die Frage: was passiert, wenn man 2 Byte Adressen einsetzen würde? Muß dann das Pad-Feld 54 Byte lang werden?

Die maximale Länge eines Ethernet-Frames beträgt 1518 Byte:

6 (DA) + 6 (SA) + 2 (Type/Length) + 1500 (Benutzerdaten) + 4 (Checksum)

Zwischen zwei MAC-Frames müssen zumindest 12 Idle-Bytes liegen, also für 96 Bitzeiträume dürfen keine Daten gesendet werden (Manchester "J" oder "K" werden verwendet).

Die "Inkompatibilitäten" des Ethernet zu 802.3 waren früher zu beachten, ist heute kaum noch relevant, da die Treiber der NICs sich darum kümmern. Der originale Ethernet Frame ist heute ebenfalls Teil der 802.3 Spezifikation.

BASE-Schreibweise und Ethernet-Varianten

Um die Vielfalt von Ethernet-Varianten (siehe unten) einfach bezeichnen zu können, hatte sich die BASE-Schreibweise etabliert.

xBASE-y

x = Datenrate in Mbps
y = Segmentlänge dividiert durch 100 in Metern (für KoAx-Ethernet)
bzw Abkürzung für das verwendete Übertragungsmedium

Jahr	**IEEE-Gruppe**	**BASE**	**Medium**
1985	802.3	10BASE-5, 10BASE-2	KoAx
1987	802.3d	FOIRL	MMF
1990	802.3I	10BASE-T	TP Cat-3
1993	802.3j	10BASE-F	MMF
1995	802.3u	100BASE-T4 100BASE-TX 100BASE-FX	TP Cat-3 TP Cat-5 MMF
1997	802.3y	100BASE-T2	TP Cat-3
1998	802.3z	1000BASE-SX 1000BASE-LX 1000BASE-CX	MMF SMF TP Cat-7 (?)
1999	802.3ab	1000BASE-T	TP Cat-7 (?)
2002 (?)	802.3ae	10GBASE-LX 10GBASE-SR 10GBASE-LR 10GBASE-ER 10GBASE-LW 10GBASE-EW	1310nm auf SMF mit WDM 850nm auf MMF 1310nm auf SMF 1550nm auf SMF 1310nm auf WAN-PHY 1550nm auf WAN-PHY

Tabelle Ethernet-Varianten

Daher steht zB 10BASE-5 für Standard Ethernet und 10BASE-2 für Cheapernet (müßte genauer eigentlich 10BASE-1,85 heißen). Mit dem Aufkommen des TP-Ethernets kam die Benennung etwas durcheinander, TP-Ethernet heißt auch 10BASE-T bzw 10BASE-TP bzw 10BASE-TX.

Mit der Ausnutzung von mehr als zwei Kabelpaaren eines TPs kamen die Varianten xBASE-T2 und xBASE-T4 in Aktion, T2 steht für zwei Kabelpaare, T4 bedeutet vier Kabelpaare.

Später kamen dann für die LWLs die Endung F hinzu (xBASE-FX).

Mit der anschließenden Unterscheidung in “long wave” und “short wave” LWL-Übertragungen bei den Gbps Ethernets teilte sich das Fx seinerseits in SX und LX auf. Ferner kam nun CX hinzu, das für “Copper” (also Kupferkabel, TP) steht.

Die 100BASE-x Varianten erhielten den Spitznamen “Fast Ethernet”.

Die 1000BASE-x Varianten erhielten den Spitznamen “Gigabit Ethernet”. Ein Firmenkonsortion, das den 1000BASE-x Standard koordiniert, ist die “Gigabit Ethernet Alliance” (GEA).

Die neue 10000BASE-x Variante wird wahrscheinlich ganz eigene BASE-Notationen erhalten, wahrscheinlich 10GBASE-x. Die treibende Kraft hier ist die “10 Gigabit Ethernet Alliance” (“10GEA”). Die oben angeführten Abkürzungen sind noch nicht festgelegt.

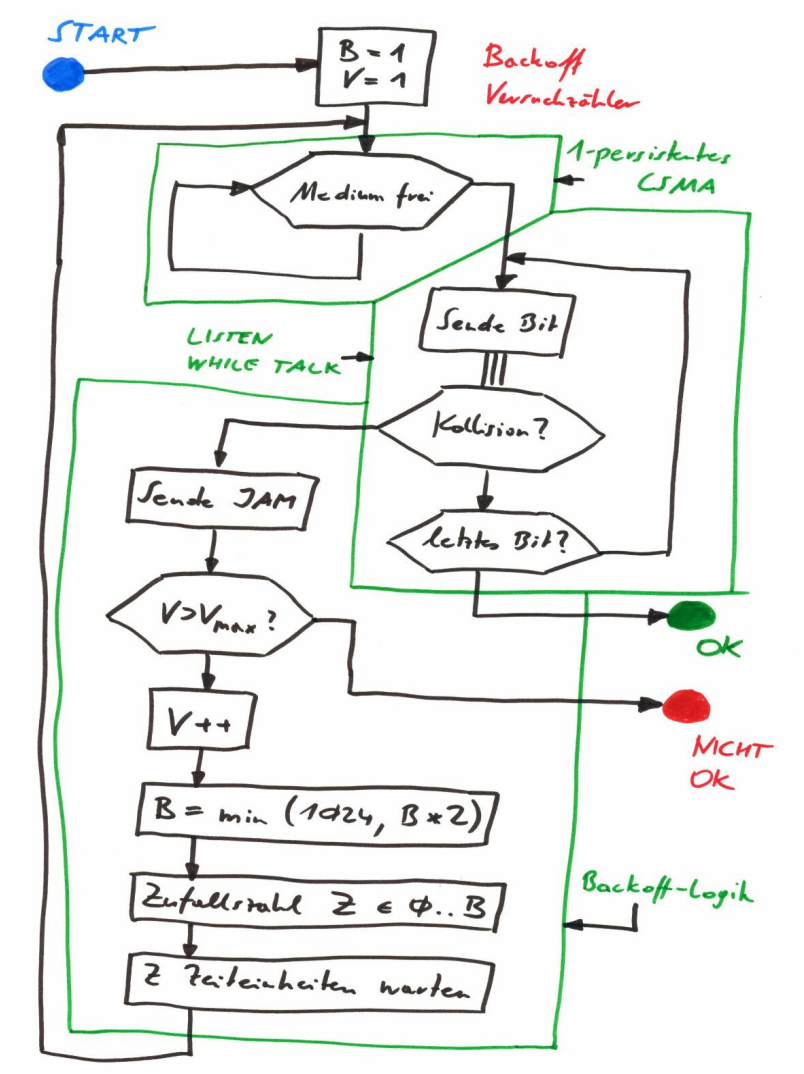
CSMA/CD MAC

Bei Ethernet wurde ursprünglich ein halbduplex “Shared Medium” MAC namens “CSMA/CD” (“Carrier Sense Multiple Access with Collision Detection”) definiert. Dieser ermöglicht es mehreren Stationen, zugleich an einem Kabel zu “hängen” und sich beim Senden zu koordinieren. CSMA/CD ist ein klassischer Contention MAC. Er kann mit den folgenden Vergleichen charakterisiert werden:

CS …	“sendet gerade jemand?”
MA …	“ich höre, was du auch hörst”
CD …	“wir senden gerade zugleich!”

Tabelle CS-MA-CD

CSMA/CD ist im Grunde genommen genau dasjenige Protokoll, das eine Gruppe von Menschen in einer Diskussionsrunde verwenden (sollten).



Skizze Flußdiagramm CSMA/CD

Das Flußdiagramm des CSMA/CD zeigt die drei großen Bestandteile des Protokolls: das sogenannte “1-persistente CSMA”, das “Listen While Talk” und die “Backoff”-Logik (Backoff bedeutet: sich zurückziehen).

Wenn also eine Station im CSMA/CD senden will, so wartet sie, bis das Medium frei wird (1-persistente CSMA).

Dann sendet sie und muß während der ersten 512 Bit (ab Gbps sind es 512 Byte) auf Kollisionen achten (“listen while talk”). Nach den ersten 512 Bit (Byte) kann eigentlich keine Kollision mehr auftreten. Falls doch, ist in den meisten Fällen die Maximallänge des Segments – oder der durch Repeater verbundenen Segmente – überschritten worden.

Wird eine Kollision erkannt, so sendet die Station einen “Jam”. Das ist ein kurzes (32 Bit langes) Signal, das einfach und eindeutig als Störung bzw Kollision erkannt wird. Dann geht die Station in den Backoff-Teil des Algorithmus über. Jede Station, die einen Jam empfängt, verwirft den bisher empfangenen Teil des Datenpakets. Jede der zugleich sendenden Stationen sendet ihr eigenes Jam-Signal, sobald sie die Kollision erkannt hat.

Der Backoff-Teil des Algorithmus dient dazu, daß die Stationen nicht sofort nach dem Freiwerden des Mediums wieder mit dem Senden beginnen. Dies hätte fatale Folgen, da alle Stationen fast

zugleich wieder zu senden beginnen würden und zusätzlich auch diejenigen Stationen senden würden, die neu hinzugekommen wären in die Runde der sendewilligen Stationen. Die nächste Kollision wäre vorprogrammiert und nach sehr kurzer Zeit würde das LAN in einer "Endlosschleife" stecken. Daher muß jede Station eine zufällige Zeitspanne warten. Diese Zeitspanne wird im statistischen Schnitt bei jedem Sendeversuch verdoppelt ("B = B * 2"). Dies gilt jedoch nur für den Durchschnitt von vielen Sendeversuchen, nicht für den einzelnen Sendeversuch! Der einzelne Sendeversuch wird von einer Zufallszahl geleitet. Der Backoff "B" wurde im Standard mit dem Wert 1024 nach oben hin gedeckelt. Die Anzahl von Stationen, die in einem gerouteten Ethernet zugleich angeschlossen sein dürfen, beträgt ebenfalls 1024. Damit hat im Idealfall in einem maximal bestückten Ethernet jede Station einen anderen Wert für die Wartezeit, falls alle Stationen senden wollen (was hoffentlich nie passiert).

Der Versuchszähler "V" wird im Ethernet-Standard von 1 bis 16 durchgezählt. Nach dem 16. Versuch gilt der Sendevorgang als mißlungen. Dies wird der oberen Schicht (LLC oder ein anderes Schicht-3-Protokoll) mitgeteilt.

CSMA/CD wurde für den Halbduplex-Betrieb konzipiert: eine Station sendet, alle anderen empfangen. Es gibt kein Senden und Empfangen zugleich, da die sendende Station genau das empfängt, was sie gerade sendet.

WICHTIG: Neuere Ethernet-Installationen, die Fullduplex-Betrieb beherrschen, verwenden kein MAC-Protokoll mehr und damit auch kein CSMA/CD mehr. Fullduplex-Betrieb bedeutet: es wird kein CSMA/CD verwendet, da zwischen sendender und empfangender Station keine anderen Stationen „mithören" und „stören" können. Full-Duplex Betrieb erfordert die ausschließliche Verwendung von Bridges oder höherwertigen Internetworking-Geräten (also keine Repeater mehr), heute kommen zumeist Switches (Layer-2 Switches) zum Einsatz.

Bei Full-Duplex-Betrieb können die beiden beteiligten Stationen zugleich Senden und Empfangen, da ab Fast Ethernet immer getrennte Leitungen für beide Datenrichtungen zur Verfügung stehen (für Ausnahmen siehe oben). Es kann also eine 100Mbps Fast Ethernet NIC mit 200Mbps Daten transferieren, 100Mbps sendend und zugleich 100Mbps empfangend.

Der Full-Duplex Modus spielt auch bei der „Point to Point" Verkabelung eine Rolle.Werden Switches kaskadiert („zusammengeschaltet") oder 2 Knoten zu einem „2 Knoten Mini-LAN" zusammengeschalten, so erfolgt dies mit „ausgekreuzten" Kabeln (siehe weiter oben) und im Full-Duplex Betrieb.

Kostenvergleich Ethernet Stand Ende 2000

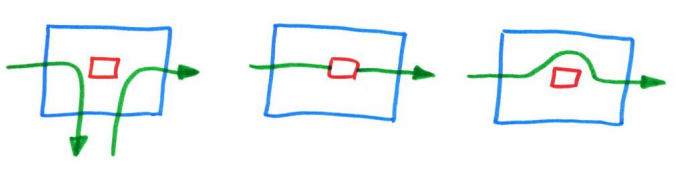
Geschwindigkeit (Mbps)	**NIC-Kosten (US$ Stand 2000)**	**Hub-Kosten für 4 Ports (US$ Stand 2000)**
10	8	18
100	8	40 (autosensing 10/100 Mbps)
1.000	260	2700
10.000	Kommt frühestens 2002	Kommt frühestens 2002

Tabelle Kostenvergleich einiger Ethernet-Varianten

IEEE 802.5 Token Ring

Der Token Ring stammt seiner Konzeption nach vom “Newhall-Ring” (ca 1969) ab. IBM hat ihn übernommen und zum „IBM Token Ring“ weiter ausgebaut. IEEE übernahm 1985 den Token Ring in den Standard IEEE 802 und nannte ihn IEEE 802.5.

Der Token Ring überträgt die Daten von Station zu Station in Form eines Ringes. Jede Station des Rings kennt drei verschiedene Zustände:



Skizze Zustände einer Token Ring Station

Im ersten Zustand ist die NIC im Sendemodus. Der Ring ist logisch geöffnet. Die Daten, die nach ihrer „Runde auf dem Ring“ wieder bei der Station eintreffen, werden dort vernichtet. Auf der anderen Seite sendet die NIC ihre Daten bitweise in den Ring hinein. Speziell bei geografisch kürzeren Ringen kommen die ersten Bits des Datenpakets bereits wieder von ihrer Runde auf dem Ring zurück, während die NIC noch immer beim Senden des Pakets ist.

Im zweiten Zustand ist der Ring in der NIC geschlossen, die Station “hört” auf dem Ring “mit”. Dieses Mithören bezieht sich auf den ein Bit großen Empfangspuffer (rotes Kästchen in der Skizze). Die Daten auf dem Ring fliegen also an einem ein Bit großen Fenster vorbei und müssen in dieser Geschwindigkeit von der Station gelesen werden.

Im dritten Zustand ist die NIC physisch aus dem Ring ausgeklinkt. Wenn die beim Token Ring übliche Phantomspannung nicht anliegt, so schließt die MSAU die Station aus dem Ring aus.

Medium

Das eigentliche Übertragungsmedium des Token Rings ist das TP. Ursprünglich war nur UTP vorgesehen, für die Überbrückung von größeren Entfernungen wurde später STP hinzugefügt.

Der original Stecker war ein sogenanntes “hermaphroditisches System”, bei dem Stecker und Buchse ident sind. IBM nennt dieses System “Datenkabel". Es hat den Vorteil, daß die Kabelenden universell aneinander gesteckt werden können. Ferner waren die Stecker intern mit einem Kurzschlußmechanismus versehen. Damit ist es möglich, daß der Stecker – wenn er abgezogen wird – den Ring sofort wieder kurzschließt. Die Verwendung des hermaphroditischen Steckers erfolgt aber nicht auf der Seite der NIC. Dort kommt ein Sub-D Stecker zum Einsatz.

Heute wird auch der RJ-45 Stecker verwendet, der die Vorteile des hermaphroditischen Steckers nicht aufweist, allerdings wesentlich kostengünstiger und universeller verbreitet ist.

Bei Verwendung von STP können bis zu 260 Stationen und bis zu 33 MSAUs in einem einzigen Ring zusammengefaßt werden. Die Entfernung der einzelnen Stationen von der MSAU kann bis zu 100 m betragen, die Entfernung der MSAUs voneinander bis zu 200m. Daraus ergibt sich eine maximale Ringlänge von beachtlichen (33*100)m + (260*100*2)m = 3300m + 52000m = ca 55km. Beim Verkabeln des Token Ring müssen allerdings in einer strengen Rechenformel die Widerstände der Stecker und Verbindungsstücke mit eingerechnet werden – diese reduzieren die mögliche Ringlänge wieder. Die 55km sind also als Idealfall zu betrachten, der von kaum einem

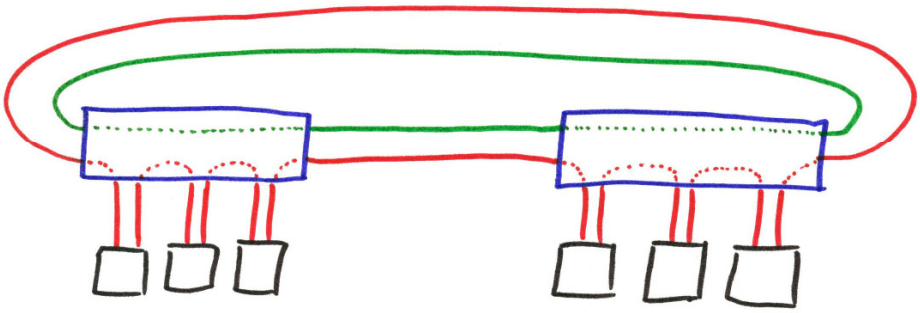
realen LAN erreicht werden kann. Wenn UTP verwendet wird, reduziert sich die Anzahl der Stationen im Ring auf 72, die Anzahl der MSAUs auf 9.

Der Grund für die Längenbeschränkung liegt nicht in der beschränkten Leistung der verwendeten Kabel oder im MAC-System oder an den elektrischen Widerständen, sondern an der Taktung des Rings. Man hatte sich beim Token Ring dazu entschlossen, daß eine spezielle Station (der aktive Ringmonitor) die Taktung des gesamten Rings übernehmen muß. Die Taktung ist der Vorgang des Bit-Sendens aufgrund des Manchester-Codes. Durch Verschiebung (Verzögerung) dieser Taktung in den einzelnen NICs kommt es zum sogenannten “Phase Jitter” und damit zum teilweisen Verlust der Synchronisationzwischen den einzelnen Stationen. Dieser elektrische Effekt verhindert die Bildung noch größerer Ringe.

An und für sich ist in einem Token Ring System jede einzelne Station ein Repeater, der unidirektional sendet. Es gibt auch kein Collision Window oder etwas Ähnliches, das die Größe des Ringes einschränkt. Man könnte daher theoretisch beliebig viele Stationen in einem Token Ring zusammenschalten und diesen Ring beliebig lang machen – wenn eben der Jitter-Effekt nicht wäre. Im FDDI, dem “logischen” Nachfolger des Token Rings, wird daher die Taktung des Rings in jeder einzelnen Station „nachjustiert“. Damit kann ein FDDI-Ring bis zu 200 km lang werden.

Topologie

Die Topologie des Token Rings ist der sternförmige Ring. Durch die Verwendung von “Ring In” und “Ring Out” Steckern in den MSAUs entsteht zusätzlich ein “innerer” Ring, der im Normalfall nicht in Verwendung ist. Er kann aber im Falle einer Rekonfiguration – ähnlich wie bei einem Doppelringsystem – aktiviert werden.



Skizze Ring mit 2 MSAUs

Übertragungsgeschwindigkeit

Der ursprüngliche Token Ring wurde mit 4 Mbps Datenrate betrieben. Später (1988) erfolgte eine Erhöhung der Datenrate auf 16 Mbps. Die Umschaltung erfolgt entweder per Software oder per Schalter auf der NIC.

Eine Urversion des Token Ring lief (vermutlich nur in den Labors der IBM) noch mit 1 Mbps. Dieser Version wurde aber aufgelassen.

Ferner gab es noch spezielle Versionen des Token Ring mit 80Mbps, zu nennen wäre hier die Firma Proteon, die die 80 Mbps Version forciert hat. Die schnell voranschreitende Entwicklung des Ethernet hat allerdings den – technisch zugegebenermaßen besseren – Token Ring überholt. Auch der sich am Horizont abzeichnende 100 Mbps FDDI Ring, der etliche der Schwächen des Token

Ring beheben sollte, ließ viele bereits ans Umsteigen denken. Zusätzlich sah die Aufstockung auf 80 Mbps im Zeitalter des (damals brandneuen) 100 Mbps „Fast Ethernet“ nicht mehr gut genug aus.

Obwohl man hier gleich anmerken muß, daß ein Token Ring mit 80 Mbps mit einem 100 Mbps Fast Ethernet leichtes Spiel hat, was die Performance - speziell im Hochlastfall - betrifft. Durch die gute Skalierung des Token Protokolls kann man fast die gesamte Brutto-Datenrate eines Token Rings in Übertragungsleistung umsetzen. Beim stochastischen Ethernet-MAC ist irgendwo bei 50% der Nettodatenrate schluß, wenn viele Stationen in einem Netz zugleich senden wollen. Der Rest geht bei Kollisionen verloren und wird damit verschwendet. Daher muß man in Ethernet-Netzwerken immer ein Auge auf die Datenrate und die Verzögerungen haben. Nimmt letztere überhand, muß man das Netzwerk entlasten. Dies erfolgt normalerweise mittels punktuell eingesetzter Bridges und “verstärken” von überlasteten Strängen durch Verdopplung oder den Einsatz höherer Datenraten. Alle diese Dinge sind bei einem Token Ring System nicht notwendig.

Kodierung

Es wird differentielle Manchester Kodierung verwendet. Eine Vorkodierung erfolgt nicht.

MAC

Der MAC des Token Rings ist das Token Protokoll des IEEE 802.5. Dieses verwendet eine Vielzahl von verschiedenen MAC-PDUs (im Gegensatz zum Ethernet, das nur eine einzige MAC-PDU kennt). Die PDUs des Token Ring werden in zwei Klassen unterteilt, in das **Token** und in den **Frame**.

Bytes	Feld
1	*SDEL (Bitfolge “JK0JK000”)*
1	*Access Control (Bitfolge “PPPTMRRR”)*
1	*EDEL (Bitfolge “JK1JK1IE”)*

Tabelle: Format des Token Ring Tokens

Im Token und im Frame werden “J” und “K”, die Sondersymbole des Manchester-Codes, verwendet. Die Bits “PPP” zeigen die aktuelle Prioritässtufe des Rings an (0 bis 7). Die Bits “RRR” zeigen den nächste gewünschte Prioritätsstufe des Rings an (Reservierung, 0 bis 7). Das Bit “T” ist das **Token Bit**, dieses ist bei einem Token immer auf “1” gesetzt. Das “M” Bit dient dem Monitoren des Rings (näheres dazu später). Die Bits “I” für “Intermediate Frame” und “E” für “Error” dienen der Anzeige besonderer Zustände. Das Token ist damit 24 Bit lang. Es muß in seiner gesamten “Länge” auf den Ring “passen”. Dazu muß der Ring gegebenenfalls verlängert werden. Dies wird durch ein Schieberegister von 24 Bit realisiert, das in jeder Station vorhanden sein muß. Dieses Schieberegister wird nur in einer einzigen Station (dem aktiven Ringmonitor, siehe unten) aktiviert und dort auch nur soviele Bits, bis das Token exakt auf den Ring paßt. Die Speicherkapazität des Rings, die sogenannte “Ringbitzahl” R, ergibt sich – analog zum Collision Window des CSMA/CD – aus der Formel:

$$R = ü * d / v$$

ü = Übertragungsgeschwindigkeit auf dem Medium in bps
d = Distanz zwischen den am weitesten auseinanderliegenden Knoten des Netzes in m

v = Mediumlichtgeschwindigkeit in m/sec, typisch 0,6-0,7c

Es wird nur der Faktor 2 nicht verwendet, da man nicht den “Round Trip” eines Signals messen muß, weil der Ring ohnehin zyklisch geschlossen ist und das Paket am Ende wieder bei der sendenden Station ankommt.

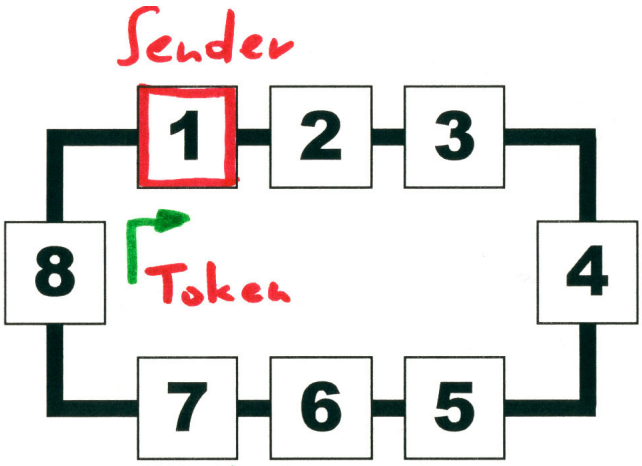
Bytes	**Feld**
1	SDEL (Bitfolge wie beim Token)
1	Access Control (Bitfolge wie beim Token)
1	Frame Control (Bitfolge “FFZZZZZZ”)
2/6	Source Address (SA)
2/6	Destination Address (DA)
n	User Data (Schicht-3-Daten), kann beim Token Ring bis zu ca 8KB groß sein
4	Checksum (CCITT CRC 32)
1	EDEL (Bitfolge wie beim Token)
1	Frame Status (Bitfolge “AcrrACrr”)

Tabelle: Format des Token Ring Frames

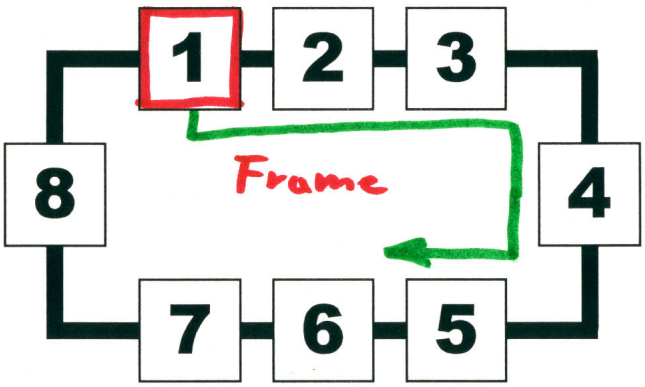
Das Frameformat des Token Rings ist dem Token in den ersten beiden Bytes ident. Das “T”-Bit ist bei einem Frame immer “0”. Das Bit “A” des Frame Status wird von jeder Station auf “1” gesetzt, die sich von der DA dieses Frames adressiert gefühlt haben (egal ob die DA Unicast, Multicast oder Broadcast war). Das Bit “C” wird zusätzlich zum Bit “A” gesetzt, wenn die empfangende (adressierte) Station das Datenpaket erfolgreich in den Empfangspuffer kopieren konnte (“Copy-Bit”). Die Bits “A” und “C” sind aus Redundanzgründen doppelt vorhanden.

Token Passing Protokoll

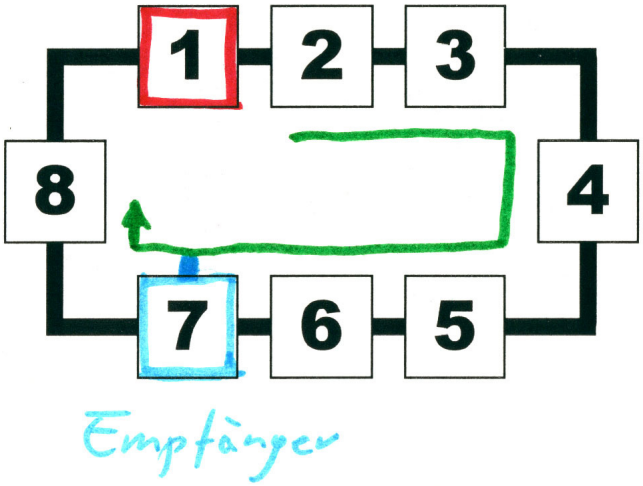
Bevor wir uns den einzelnen MAC-Frames zuwenden, müssen noch einige wichtige Konzepte des Token Rings diskutiert werden. Um einmal beim einfachsten Fall zu bleiben, sehen wir uns einen Sendevorgang im Token Ring an. Hierbei empfängt die Station 1 ein Token und setzt “on the fly” das Bit “T” auf “0” und das Bit “M” auf “0”. Anschließend fährt sie fort, den Rest des Frames zu senden. Damit hat die Station aus den beiden ersten Bytes des Tokens die ersten beiden Bytes eines Frames gemacht. Die Station 7 wird von diesem Frame adressiert.



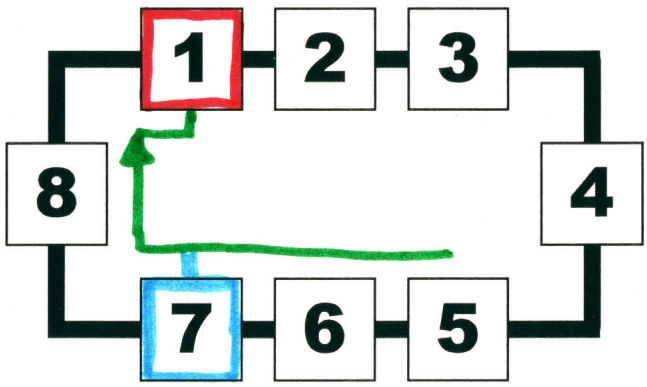
Der Die Station1 erhält das Token von Station 8, öffnet den Ring und beginnt zu senden. Bit „T“wird auf 0 gesetzt („Das ist kein Token“), Bit „M“ auch auf 0 (zeigt dem ARM einen neuen Frame an).



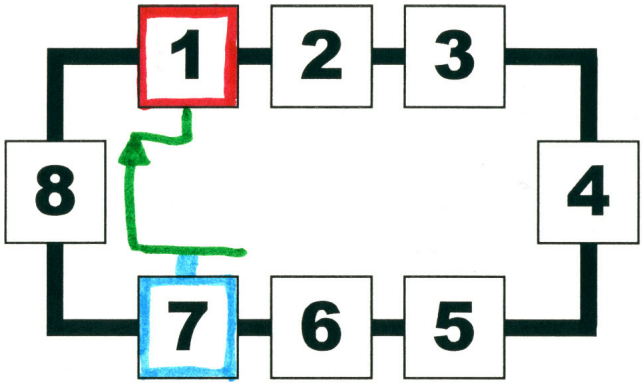
Das Datenpaket (Frame) der Station 1 ist bereits bei der Station 5 vorbei und auf dem Weg zu Station 7, dem Empfänger.



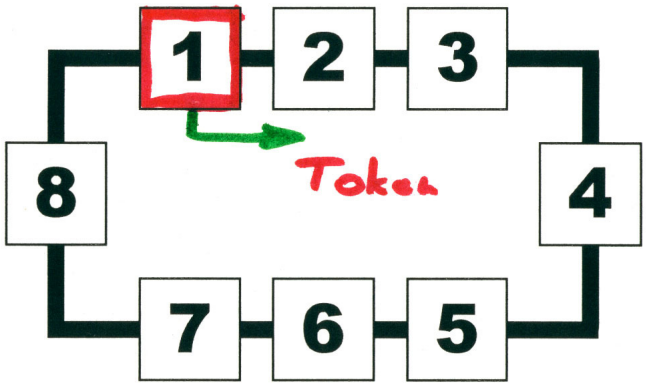
Station 7 kopiert den Inhalt des Frames, während dieser „vorbeiläuft“. Station 1 hat bereits mit dem Senden aufgehört. Der Ring ist immer noch „offen“.



Station 7 kopiert noch das Ende des Frames, Station 1 vernichtet bereits den Kopfteil des Pakets.



Station 7 kopiert immer noch, Station 1 vernichtet weiterhin. Station 7 wird dann die Bits „A“ und „C“ im Frame Status setzen. Der Ring ist immer noch offen.



Nun erst, wenn Station 1 das gesamte eigene Paket wieder vernichtet hat, erstellt sie ein neues Token (Bit „T“ = 1) und schließt den Ring.

Skizze Sendevorgang im Token Ring

Die Aufgabe der Station 7 ist es, die Bits “A” („durch dieses Datenpaket fühle ich mich adressiert“) und “C” („ich habe mich nicht nur addressiert gefühlt, sondern auch die Daten des Datenpaketes an meine oberen Schichten weitergeleitet“) zu setzen.

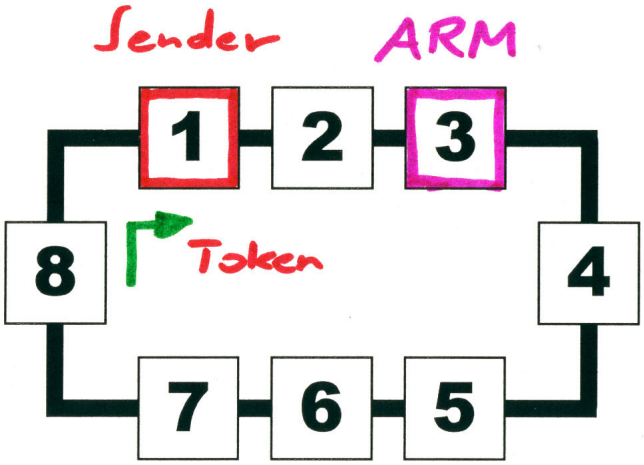
Die Station 1 muß die auf dem Ring umgelaufenen Daten des Frames wieder “empfangen” und vernichten. Wenn das letzte Bit des Frames nach einer Runde am Ring wieder bei der Station 1 eingetroffen ist, sendet die Station ein neues Token ab.

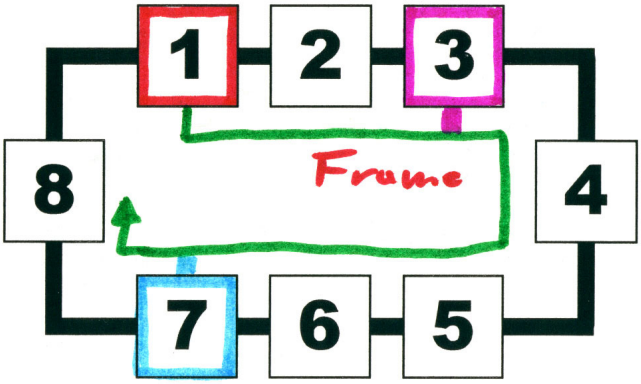
Aktiver Ringmonitor

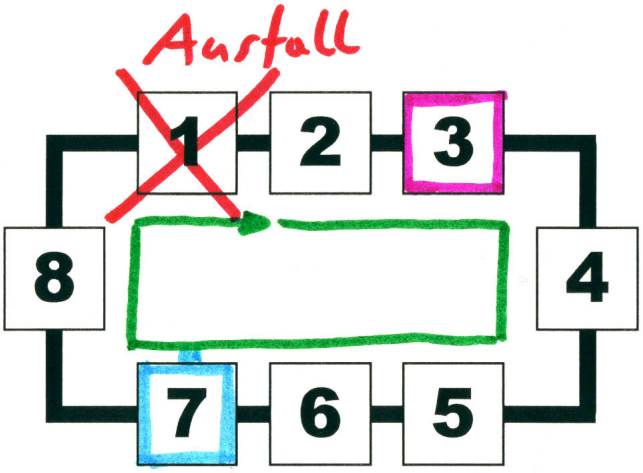
Wenn nun bei dieser Übertragung etwas schiefläuft, muß es eine Station im Ring geben, die das überwacht und gegebenenfalls korrigierend eingreift. Die Station 3 spielt hier diese Rolle, nämlich die des Aktiven Ring Monitors (“ARM”). Der ARM hat eine Menge von Aufgaben, von denen wir uns eine wichtige und einfache jetzt näher ansehen. Wenn die Station 1 ihrer Pflicht, das gesendete Datenpaket nach einer Runde am Ring wieder zu vernichten, nicht nachkommt, muß dies der ARM tun.

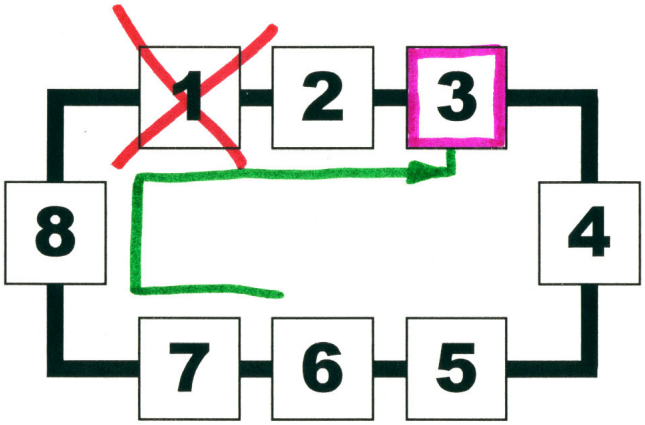
In diesem Beispiel wird die Station 1 vom Ring weggeschaltet, während sie noch Daten sendet. Damit ist der Frame erstens unvollständig und zweitens – noch schlimmer – gibt es keine Station mehr, die den unvollständigen Frame je wieder vom Ring löscht. Damit würde der Frame “ewig im Ring kreisen” – der Ring wäre effektiv unbrauchbar, da kein neues Token mehr entsteht.

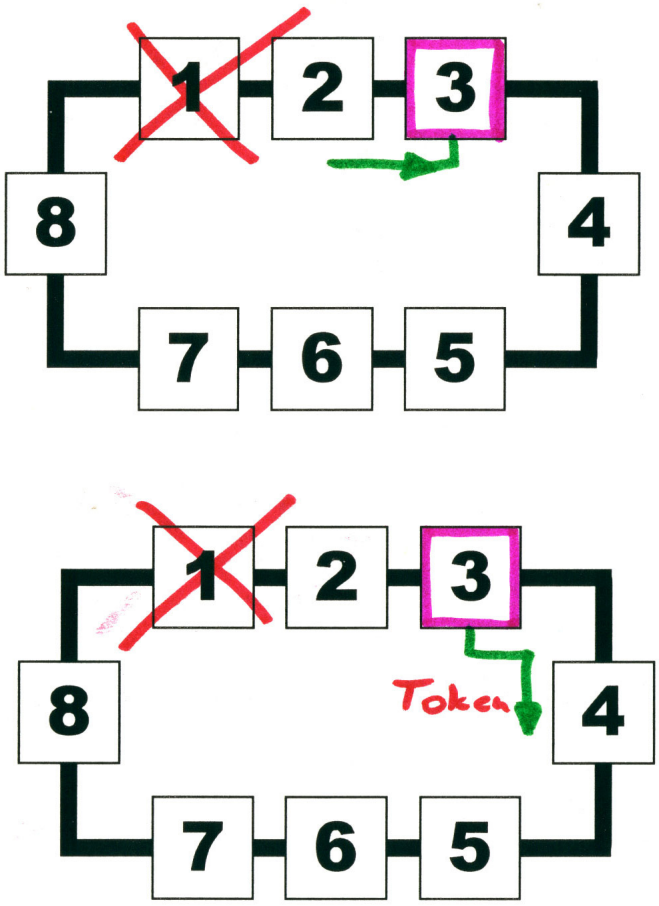
Um diesen Fehlerfall zu erkennen, muß der ARM jeden Frame markieren. Er verwendet dafür das “M” Bit im Access Control Feld des Frames und setzt es von “0” auf “1”. Läuft der Frame ein zweites Mal am ARM vorbei, so ist das “M” Bit bereits gesetzt und der ARM muß annehmen, daß der Frame von der sendenden Station nicht mehr vom Ring gelöscht wurde. In diesem Fall übernimmt der ARM die Rolle der Station und „verschluckt“ den Frame-Rest, der nach den Feldern SDEL und Access Control folgt, weg. Da die ersten 1,5 Bytes des Frames aber bereits am ARM “vorbeigelaufen” sind, muß der ARM anschließend den Ring “säubern”. Er tut dies, indem er den Purge Frame sendet.




1
2
3
Frame
8
4
7
6
5


Ausfall
1
2
3
8
4
7
6
5


1
2
3
8
4
7
6
5



Skizze Sendevorgang mit Stationsausfall im Token Ring

MAC-Frames

Die einzelnen Bits des Frame Control zeigen an, um welchen Frame-Typ es sich handelt. Die Bits werden folgend interpretiert:

FFZZZZZZ	**Bedeutung**
00 xxxxxx	Dieser Frame ist ein MAC-Frame. Die Daten im Feld “User Data” beinhalten Steuerinformationen des Token Protokolls. Die wichtigsten MAC-Frames sind:
00 000011	“Claim Token”. Dieser Frame wird gesendet, wenn sich der ARM nicht rechtzeitig gemeldet hat. In diesem Fall kämpfen die anderen Stationen des Rings darum, wer der nächsten ARM wird. Dabei gewinnt immer die Station mit der größten MAC-Adresse. Um den Wettkampf durchzuführen, werden die Claim Token Frames benötigt. Das Token Passing ist während dieser Zeit ausgesetzt. Dieser Wettkampf-Aspekt gibt auch dem Token Ring eine gewisse Contention-Note, er kommt aber selten vor.
00 000000	“Duplicate Address Test” (“DAT”). Dieser Frame wird von jeder Station gesendet, bevor sie ihren ersten “richtigen” Frame absendet. Der DAT wird von einer Station an sich selbst gesendet (SA = DA). Nach einer Runde am Ring muß das “A” Bit immer noch auf “0” stehen. Ist es auf “1”, dann hat eine andere Station dieselbe Adresse wie diejenige Station, die gerade den DAT Frame gesendet hat. In diesem Fall muß diese Station einen Fehler melden und sich wieder (elektrisch) aus dem Ring ausklinken.
00 000101	“Active Monitor Present” (“AMP”). Dieser Frame wird vom ARM in regelmäßigen Abständen als “Lebenszeichen” gesendet. Jede Station außer dem ARM ist ein “Standby Ring Monitor” (“SRM”) und wartet auf das zeitgerechte Eintreffen des AMP Frames. Bleibt dieses aus, beginnt der SRM mit dem Senden der “Claim Token Frames” und beendet damit das Token Protokoll. Ein neuer ARM muß bestimmt werden.

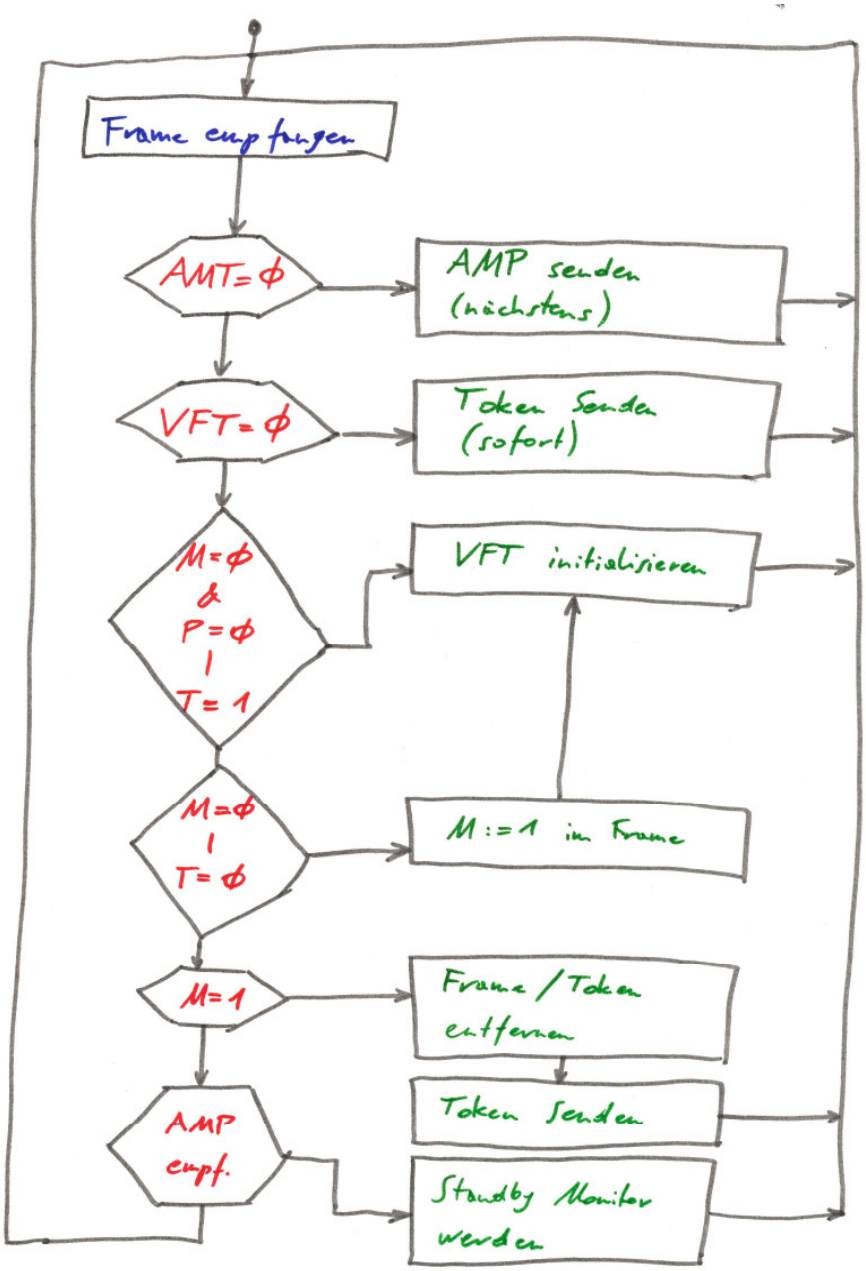
00 000110	“Standby Monitor Present” (“SMP”). Jeder SRM sendet in regelmäßigen Abständen den SMP Frame mittels Broadcast. Damit wird folgendes erreicht: jede Station im Ring kennt ihren “Vorgänger”. Dieser wird in der Token Ring Literatur als “Next Active Upstream Neighbor” (“NAUN”) bezeichnet. Jede Station ist ein SRM und sendet SMPs an die nächste Station (ihren “Nachfolger”). Diese erkennt den Frame als an sich adressiert (es ist ja ein Broadcast) und prüft das “A” Bit. Ist dieses gesetzt, so ist sie nicht der direkte Nachfolger derjenigen Station, die den Frame gesendet hat. Andernfalls ist sie der direkte Nachfolger und die SA, die in diesem SMP Frame steht, ist die Adresse des NAUN. Diese Adresse merkt sich die Station. Die AMPs des ARM werden verwendet, um den NAUN festzustellen, da der ARM keine SMP Frames sendet.
00 000010	“Beacon”. Eine Station, die keine Frames und keine Token innerhalb einer bestimmten Zeitspanne erhalten hat, nimmt an, daß die Leitung oder “die Station oberhalb” (= der NAUN) ausgefallen ist. Da jede Station ihren NAUN kennen sollte (sie kennt ihn nicht, wenn sie gerade erst in den Ring hinzukam), kann sie mittels Beacon Frame anzeigen, zwischen welchen Stationen ein Problem vermutet wird. Sie sendet daher im Beacon Frame ihre eigene Adresse und die des NAUN. Zwischen diesen beiden Stellen muß der Fehler liegen. Jede Station, die einen Beacon Frame erhält, stoppt das Token Protokoll und sendet exakt diesen Frame weiter. Damit ist in allen Stationen bekannt, wo das Problem wahrscheinlich liegt und kann in allen Stationen mittels geeigneter Monitoring-Software ausgelesen werden. Während des Beakoning ist klarerweise kein Token Passing aktiv, da der Ring nicht geschlossen ist.
00 000100	“Purge”. Der ARM sendet den Purge Frame, wenn er eine Unregelmäßigkeit im Ring entdeckt hat. Der Purge zeigt allen Stationen einen Reset-Zustand an. Alle Stationen setzen daraufhin ihre internen Timer und Variablen zurück. Nach dem Purge sende der ARM das Token neu aus.
01 000000	Dieser Frame ist ein LLC-Frame. Die Daten im Feld “User Data” beinhalten tatsächlich Benutzerdaten, also Daten, die aus der Schicht 3 kommen.

Tabelle: MAC Frame Formate des Token Ring

Wie man aus den obigen Vorgängen sieht, muß jede Station in der Lage sein, ein ARM zu werden. Daher muß in allen Stationen des Token Rings dieselbe Software installiert sein. Der ARM hat eine Menge von Zeitüberschreitungen zu überwachen und hat daher mehrere interne Timer. Auch die SRMs haben laufende Timer. Diese sind notwendig, um ganz spezielle Situationen im Ring herauszufinden.

ZB wenn sich der ARM nicht mehr meldet. Oder wenn der NAUN nichts mehr sendet. Oder wenn das Token nicht innerhalb einer bestimmten Zeitspanne eintrifft.

Diese Zustände können in einem Token Ring relativ leicht festgestellt werden, benötigen aber Koordination der Stationen untereinander und den Einsatz von Timern. Ein weiterer Grund, warum ein Token Ring auch nicht beliebig groß gemacht werden kann, da ansonsten durch die große Ringbitzahl die Timer der SRMs und des ARMs überlaufen würden, obwohl noch alles in Ordnung ist. Dem ARM fallen wichtige Aufgaben zu, ohne denen die Funktionsfähigkeit des Rings nicht gegeben wären. Die Aktionen nach dem Empfangen eines Frames im ARM sehen folgend aus:



Skizze Vereinfachtes Flußdiagramm des ARMs

Prioritäten

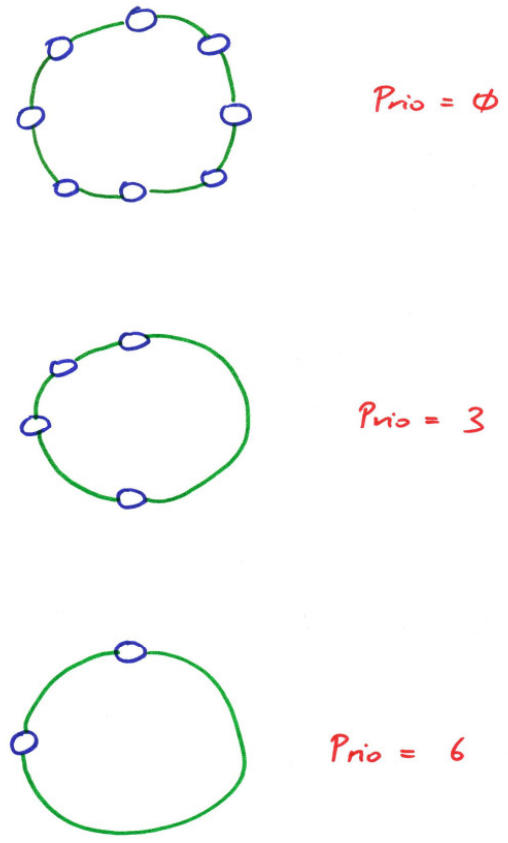
Zusätzlich wurde im Token Ring ein (optionaler) Prioritäten-Mechanismus integriert. Hierbei wird über drei Bits die Prioritäts-Stufe des Ringes vorgewählt ("RRR") und über drei weitere Bits ("PPP") die aktuelle Stufe des Tokens bzw Frames angegeben. Der Prioritäten-Mechanismus sorgt dafür, daß die Stationen eines Rings quasi "Sub-Ringe" mit höherer Priorität bilden können. Wenn eine Station die Priorität des Rings von 0 (der Basis-Priorität) auf 3 anheben will, so setzt sie – wenn nicht gleich ein Token kommt – im nächsten Frame die Reservierungsbits auf den Wert 3. Dann wartet sie auf das Token. Erhält die Station das Token, dann macht sie einen Frame daraus und sendet ihre Daten, ohne im Frame die Bits "RRR" oder "PPP" zu verändern.

Nachdem sie ihren eigenen Frame wieder vernichtet hat, sendet die Station ein neues Token aus, in dem aber jetzt die Prioritäts-Bits "PPP" auf den Wert 3 gesetzt sind (wenn die Station auch

weiterhin Daten auf Priorität 3 senden will). Eine Station, die nur Daten mit einer niedrigeren Priorität als 3 senden will, darf das jetzt nicht mehr tun, auch wenn sie das Token erhält. Der Ring befindet sich nämlich auf einer anderen Prioritäts-Ebene, in der nur noch diejenigen Stationen senden dürfen, die Daten auf dieser oder einer höheren Ebene senden wollen. Alle anderen Stationen müssen das Token sofort weiterreichen. Es bildet sich daher ein "Sub-Ring" mit denjenigen Stationen, die dieser Prioritätsebene angehören.

Diejenige Station, die die Priorität des Rings angehoben hat (das ist diejenige Station, die die "PPP" Bits auf 3 gesetzt hat), muß die Priorität auch sofort wieder auf den alten Wert (hier 0) zurücksetzen, wenn sie keine Daten mit Priorität 3 mehr senden will.

Wenn jedoch innerhalb der Prioritätsstufe 3 eine Station Daten mit Priorität 6 senden will, so kann sie ihrerseits den Ring in diese Prioritäts-Ebene heben und damit alle Stationen ausschließen, die nur mit Priorität 3 senden wollen.



Skizze Prioritätsmechanismus in einem Ring mit 8 Stationen

Es ist immer diejenige Station für das Zurücksetzen der Ringpriorität verantwortlich, die die Anhebung veranlaßt hat. Fällt die Station aus, bevor sie die Priorität rücksetzen konnte, muß dies der ARM übernehmen. Dafür muß der ARM für jede einzelne der 8 Prioritätsstufen einen eigenen Timer mitführen.

Der Prioritäts-Mechanismus des Token Ring ist daher ein ausschließlicher und außerdem fair. Die nächste Station, die senden darf, ist immer diejenge mit der höchsten Priorität.

ANSI X3T9.5 - FDDI

Der ANSI Standard X3T9.5, genannt “Fiber Distributed Data Interface” oder kurz FDDI, ca 1980 erstellt, wurde nie zu einem OSI- bzw IEEE-Standard. FDDI enstand aus der Notwendigkeit, das Token Ring System, das an den Grenzen seines Ausbaus angelangt war, abzulösen. Der Token Passing MAC ist einer der limitierenden Faktoren des Token Ring, da von einer bitweise Interpretation der PDUs “on the fly” ausgegangen wird. Dies wird aber mit steigender Datenrate immer schwerer umzusetzen. Während Firmen wie zB Proteon sich dem Ausbau des Token Rings auf 80Mbps widmeten, definierte ANSI mit dem FDDI ein komplett neues System, das mit dem Token Ring fast nichts mehr gemein hat, außer das beide Systeme topologisch Ringe sind.

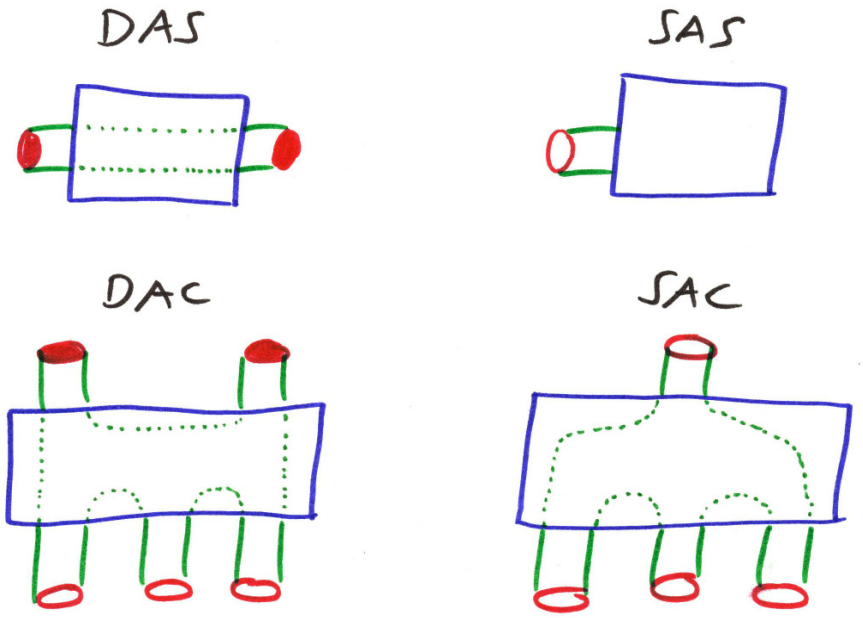
Medium

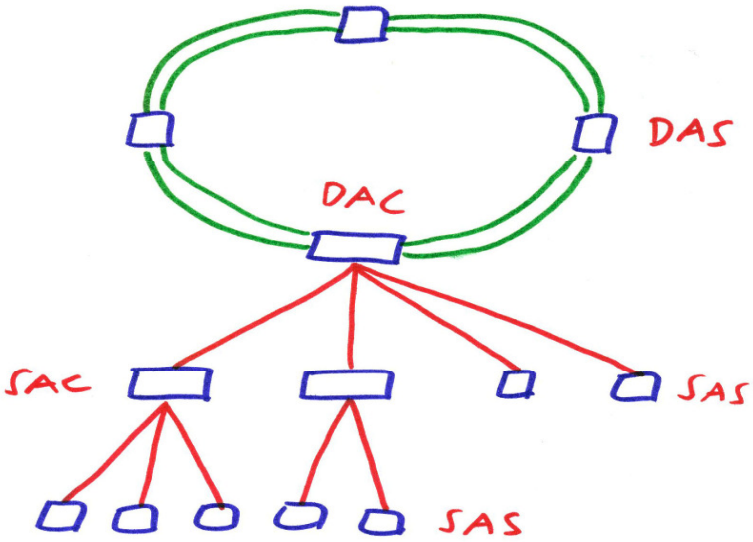
Es wird ein LWL (zumeist Multimode - 50/100 µm bzw 62,5/125 µ m) verwendet. Die Sendefrequenz beträgt 1300 nm (Infrarot-Bereich).

Als später hinzugekommene Variante ist auch das CDDI (“Copper Distributed Data Interface”) zu erwähnen. Bei dieser Form des FDDI wurde aus Kostengründen mit Kupferleitungen (TP) gearbeitet, was aber die geografische Erstreckung des Systems stark verringert.

Topologie

FDDI arbeitet als doppeltes Ringsystem. Ursprünglich war ausschließlich ein Doppelring vorgesehen, aber später wurde - aus Kostengründen - ein Einfachringsystem hinzugefügt. Die Topologie ist daher nun ein Doppelring im Backbone und ein Baum im Endbereich:





DAS = Dual Attachment Station
SAS = Single Attachment Station
DAC = Dual Attachment Concentrator
SAC = Single Attachment Concentrator

Skizze FDDI-Stationen, Konzentratoren und Topologie

In der obigen Skizze sind Stationen (oben) und Konzentratoren (unten) zu sehen. Im Doppelring-Bereich sind nur DACs und DASs zu finden. An einen DAC können dann beliebig SASs oder SACs angeschlossen werden.

Der FDDI-Ring kann sehr groß werden:

bis zu 200 km Leitungslänge
bis zu 1000 Stationen in einem Ring
bis zu 2 km Stationsabstand ohne Leitungsverstärker (Repeater) überbrückbar

Tabelle Maximale Größe eines FDDI Ringes

Übertragungsgeschwindigkeit

Im FDDI wurde die Datenübertragung mit 100 Mbps festgelegt. Intern wird mit 100 Mbps + 25% = 125 Mbps gesendet, was an der Vorkodierung mittels 4B/5B liegt.

Vorkodierung

Es wird 4B/5B verwendet.

Kodierung

Die Hauptkodierung erfolgt mittels NRZ.

MAC

Der MAC des FDDI ist ebenfalls ein Token Passing Verfahren namens “Timed Token”. Die Abweichungen zum Token Passing des IEEE 802.5 Token Ring können folgend zusammengefaßt werden:

Early Token Release: Das Token wird sofort nach dem Senden des Datenpakets auf den Ring gesetzt (also direkt hinter das Datenpaket). Damit kann in sehr großen Ringen der Durchsatz optimiert werden, da sich zu einem Zeitpunkt mehrere Datenpakete zugleich am Ring befinden können. Als Beispiel: die Ringbitzahl R = Ü*D/V beträgt bei 200 km Ringlänge und 100 Mbps Datenrate 48.000 Bit, entsprechend ca 6 KB!

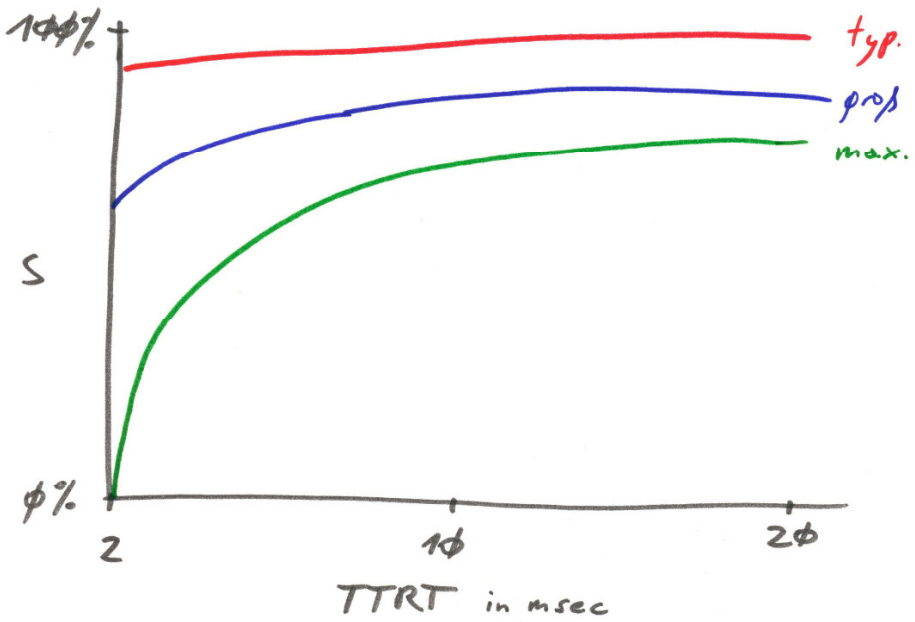
Token Absorption: Das Token wird als ganzes Empfangen und in einem Puffer abgelegt. Dann erst wird es Analysiert und ggf durch ein Datenpaket (+ abschließendes Token) ersetzt.

Alle Stationen sind aktive Ringmonitore.

Die Verwendung von mehreren sogenannten Allokationen (siehe unten).

Innerhalb der Asynchronen Allokation (AA) wird zusätzlich ein Prioritäten -Mechanismus implementiert: Ein von der Station zu sendender Frame mit Priorität X wird nur dann gesendet, wenn noch Z Zeiteinheiten in der AA übrigbleiben. Der Zusammenhang zwischen X und Z wird linear errechnet (kleines Z => große X). Dies ergibt eine ungefähr faire Verteilung der AA auf die einzelnen Stationen.

Auf die Allokationen müssen wir näher eingehen: Allokationen sind “Zuteilungen” der Ringkapazität an die einzelnen Stationen im Ring. Zu Beginn wird die Target Token Rotation Time (“TTRT”) bestimmt. Dies geschieht, in dem alle Stationen im Ring einer bestimmten TTRT zustimmen. Die TTRT definiert die “Taktrate” des Rings. Ist die TTRT klein, kommt das Token “oft vorbei” (Optimierung des Delays), es können aber jeweils nur kurze Datenpakete gesendet werden (schlecht für den Durchsatz). Umgekehrt ist bei großer TTRT der Durchsatz gut, da man lange Datenpakete senden kann, bevor man das Token wieder abgeben muß. Aber das Token kommt entsprechend “selten” bei jeder Station vorbei. Ist der Ring sehr lang, braucht das Token für seinen Umlauf bereits etliche Millisekunden. In diesem Fall muß man die TTRT zumindest größer als die Tokenumlaufzeit wählen, ansonsten darf eine Station auch bei Erhalt des Tokens nicht senden!



Skizze TTRT vs Durchsatz
typ: typischer FDDI Ring mit 20 SAS und 4km Leitungslänge, Tokenumlaufzeit = 0,04ms
groß: 100 SAS im 100km Ring, Tokenumlaufzeit = 0,6ms
max: 500 DAS im 200km Ring, Tokenumlaufzeit = 2ms

Durch die TTRT ist jeder Station eine bestimmte “Sendezeit” und damit ein bestimmter Durchsatz pro Zeit garantiert. Diese garantierte Allokation nennt man “Synchrone Allokation” (“SA”). Damit soll angedeutet werden, daß diese Form der Allokation in regelmäßigen Abständen verfügbar ist. Mit der SA ist für eine Station das Senden einer definierten Datenmenge (garantierter minimaler Durchsatz) innerhalb einer bestimmten Zeit (garantierte maximale Verzögerung) festgelegt.

Wenn eine Station innerhalb eines Tokenumlaufs ihre SA nicht aufbraucht, entsteht die sogenannte Asynchrone Allokation (“AA”). Das ist diejenige SA, die von den vorherigen Stationen nicht verbraucht wurde. Diese AA kann dann von anderen Stationen zusätzlich zu deren SA verbraucht werden.

Die AA wurde künstlich nach oben limitiert, und zwar auf den Wert der TTRT. Ansonsten könnte in einem wenig beschäftigten FDDI Netz die AA rasch anwachsen. Die nächste Station, die Daten in der AA senden will, könnte dann sehr lange Daten senden, ohne das Token weitergeben zu müssen. Die Limitierung der AA auf TTRT verhindert dies. Warum diese Limitierung eingeführt wurde, ist klar ersichtlich: die SA könnte sonst nicht immer eingehalten werden und das Token“zu spät” kommen, was der Definition und dem Sinn der SA widerspricht.

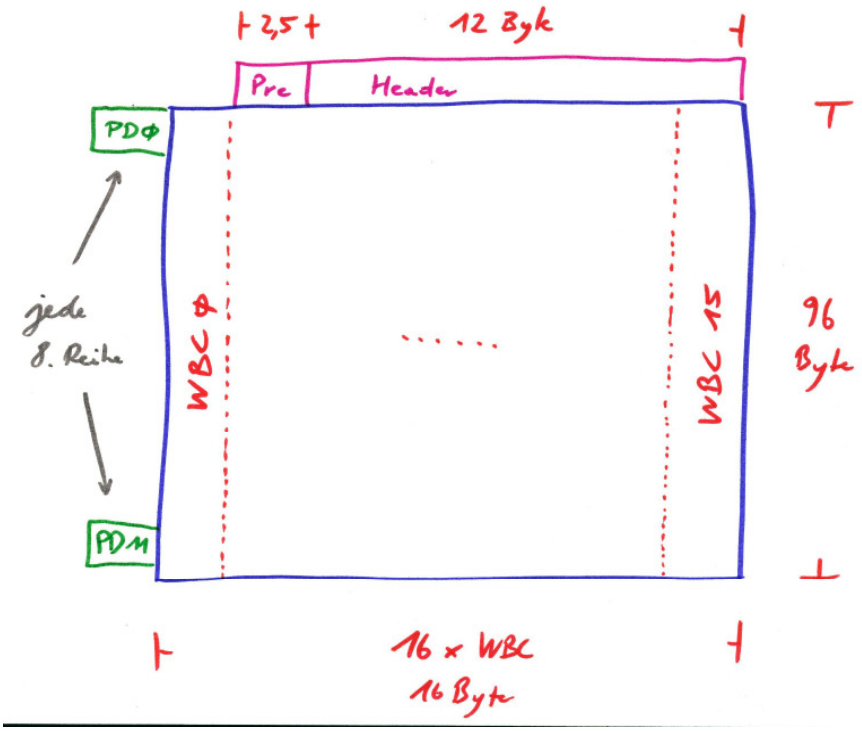
ANSI X3T9.5 - FDDI-2

Der Standard FDDI-2 entspricht in Medium, Topologie und Kodierung vollständig dem originalen FDDI (“FDDI-1”).

Es wird aber zusätzlich zur SA und zur AA eine sogenannte Isochrone Allokation (“IA”) eingeführt. Damit ist FDDI-2 echtzeitfähig (exakte Datenrate und exakte Verzögerung sind garantiert, isochrone Kommunikation möglich), während FDDI-1 nur annähernd echtzeitfähig ist.

MAC

Ein FDDI-2 System startet grundsätzlich im FDDI-1 Modus hoch (“Basic Mode”) und schaltet nur dann, wenn *alle* Stationen des Ringes FDDI-2 fähig sind, in den FDDI-2 Modus (“Hybrid Mode”) um. Hierbei wird ein sogenannter “Cycle Master” definiert (per Contention-Verfahren). Der Cycle Master ist eine besondere Station im Ring, die alle 125µs einen Cycle absendet.



Skizze Cycle

Dieser Cycle alle 125µs ist in 16 “Wide Band Channels” (“WBC”) unterteilt, wobei jeder WBC 96 Bytes umfaßt. Zusätzlich werden noch Steuerinformationen übertragen, sodaß die gesamte Information eines Cycles 12500 Bit (das entspricht 1562,5 Byte!) beträgt.

```
16 WBCs a 96 Byte = 1536 Byte = 12288 Bit
12 Byte Header
12 Byte Packet Data
2,5 Byte Preamble
```

Innerhalb des Cycles können sich nun die Stationen im Ring Kapazitäten reservieren. Die kleinste Kapazität, die reservierbar ist, beträgt ein Byte (1/96-stel eines einzelnen WBCs). Wenn man alle 125us genau 1 Byte überträgt, errechnet sich die Kapazität dieses sogenannten “Subchannels” als 125µs * 8 Bit = 64 Kbps, das entspricht exakt der Datenrate eines ISDN-B-Kanals.

Bits/Cycle	**Resultierende Datenrate (Kbps)**
12288	(alle 16 WBCs) 98,3 Mbps
768	(ein ganzer WBC) 6,144 Mbps
240	30 B-Kanäle des ISDN = H12-Kanal
193	T1-Träger (US, UK)
192	24 B-Kanäle des ISDN
48	6 B-Kanäle des ISDN = H11-Kanal
8	1 B-Kanal des ISDN = H0-Kanal

Tabelle Gängige Subchannels der FDDI-2 WBC

Die Cycle-Struktur des FDDI-2 überlagert das Token-Passing Protokoll des FDDI-1. Im Gegensatz zum Token Verfahren, bei dem nur dann Daten auf dem Ring gesendet werden, wenn auch welche gesendet werden sollen (eine Ausnahme bildet hierbei das Token) sind beim FDDI-2 immer Daten (Cycles) auf dem Ring, und zwar ohne Pause und ohne "Zwischenraum". Wenn man sich das Token Passing als Staffellauf vorstellt, bei dem immer nur eine Station das Token hält und senden darf, ist beim FDDI-2 jede Station praktisch jederzeit in der Lage, in einen Cycle Daten "einzufüllen" - vorausgesetzt, diese Station hat sich innerhalb des Cycles einen Platz reserviert. Diesen reservierten Platz nennt man die "Isochrone Allokation" ("IA") der Station.

Dies ist auch der große Gegensatz zur SA. Eine Station, die in der IA senden will, *muß* sich vorher innerhalb des Cycles den Platz fix reservieren. Minimum ist 1 Byte, Maximum sind 12 x 96 Bytes. Dieser reservierte Platz steht ab dann der Station in *jedem einzelnen* Cycle (alle 125µs) zur Verfügung und kommt *garantiert* alle 125µs "bei der Station vorbei" (außer im Fehlerfall). Während also die SA des FDDI-1 vom zeitgerechten Eintreten des Tokens abhängt (dieses kann zu früh, aber auch in beschränktem Umfang zu spät kommen), steht beim FDDI-2 alle 125µs ein vorab definierter Platz im Cycle zur Verfügung. Ob er gefüllt wird oder nicht, hängt von der reservierenden Station ab. Gegegebenfalls wird also Bandbreite verschwendet. Daher muß die IA sinnvoll gewählt werden und darf nicht überstrapaziert werden.

Der Vorteil der IA liegt darin, daß sie im Gegensatz zur SA immer und pünktlich zur Verfügung steht. Sie eignet sich daher sehr gut für alle Anwendungen, die Echtzeitdaten übertragen wollen (Telefonie, Video, Audio, etc).

Da nicht immer alle WBCs komplett "ausgebucht" mit IA-Subchannels sind, bleibt meist noch "Platz" innerhalb des Cycles übrig. Dieser wird - kompatibel mit FDDI-1 - für SA und AA verwendet. Innerhalb der nicht-reservierten Teile der Cycle wird damit das Token Passing Protokoll des FDDI-1 implementiert. Auch wenn alle WBCs komplett für IA vergeben sind, kann im Header-Bereich immer noch - in platzmäßig beschränktem Umfang - per Token Passing SA und AA abgewickelt werden.

CDDI

Copper Distributed Data Interface

Implementierung des FDDI auf Kupferkabel-Basis. Kein offizieller Standard, von einigen Firmen als Firmenstandard implementiert. Ziel war die kostengünstige Implementierung des FDDI auf billigen Kupferleitungen. Heute nicht mehr aktuell aufgrund des Preisverfalls der LWL.

Vergleich FDDI und Token Ring

	FDDI	**Token Ring**
Standard	ANSI	IEEE
Standard-Nummer	ASC X3T9.5	802.5
Medium	LWL	UTP / STP
Übertragungs-Geschw.	100 Mbps	4 / 16 Mbps
Kodierung	4B/5B + NRZ	Differentieller Manchester
Topologie	Doppelring + Einfachring (hierarchisch)	Sternförmiger Ring

Monitoring	Alle Stationen sind aktive Ringmonitore	Eine Station ist aktiver Ringmonitor
Clocking	Alle Stationen clocken den Ring	Der Aktive Ringmonitor clockt den Ring
Name des MAC Verfahrens	Timed Token	Token Passing
Token Release	Early	(normal)
Max. Frame-Größe	4500 Byte	(theoretisch) unendlich

Tabelle Vergleich Token Ring und FDDI

ATM

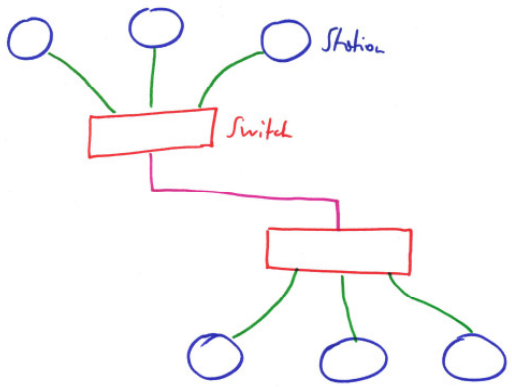
Der “Asynchronous Transfer Mode” ist eine Variante des Cell Relay (eigentlich die einzige). Er entstand aus dem Frame Relay durch festlegung auf ene fixe PDU-Größe.

Medium

Das Medium ist nicht definiert im Rahmen des ATM. Normalerweise kommt UTP (für niedrige Datenraten) oder ein LWL (für hohe Datenraten) zum Einsatz.

Topologie

Die Topologie ist ebenfalls nicht definiert im Rahmen des ATM. Im Allgemeinen wird ein Maschennetz verwendet, an dessen Knotenstellen ATM-Switches stehen.



Skizze ATM Topologie

Übertragungsgeschwindigkeit

Die Datenrate ist ebenfalls nicht definiert im Rahmen des ATM. Als Geschwindigkeitsstandards haben sich 25 Mbps (Desktop-Einsatz), 155 Mbps und 622 Mbps etabliert.

Kodierung

Ist nicht definiert im Rahmen des ATM.

MAC

Es wird ein rein verbindungsorientierter MAC verwendet. Es werden kleine Pakete, bezeichnet als "**Cells**", versendet. Diese beinhalten 48 Byte Nutzdaten und 5 Byte Headerinformation. Bevor eine Station einer anderen Station etwas senden kann, muß eine "virtuelle Verbindung" aufgebaut werden ("**Virtual Circuit**", "**VC**"). Die Identifikation des VCs (sogenannte "VCI") wird dann in allen Cells dieser Route im Header angegeben, um die einzelnen Cells zu routen.

ATM definiert "Quality of Service" ("QoS") Parameter beim Verbindungsaufbau.

ATM paßt nicht sehr gut mit LANs zusammen und ist auch keine eigentliche LAN-Technologie. Es wird in dieser Abhandlung nur wegen der Überschneidung zwischen LANs und WANs gebracht.

Um ATM als LAN zu nutzen, muß entweder mit "LAN Emulation" ("LANE") oder "IP over ATM" gearbeitet werden (näheres dazu siehe unter Windows 2000).

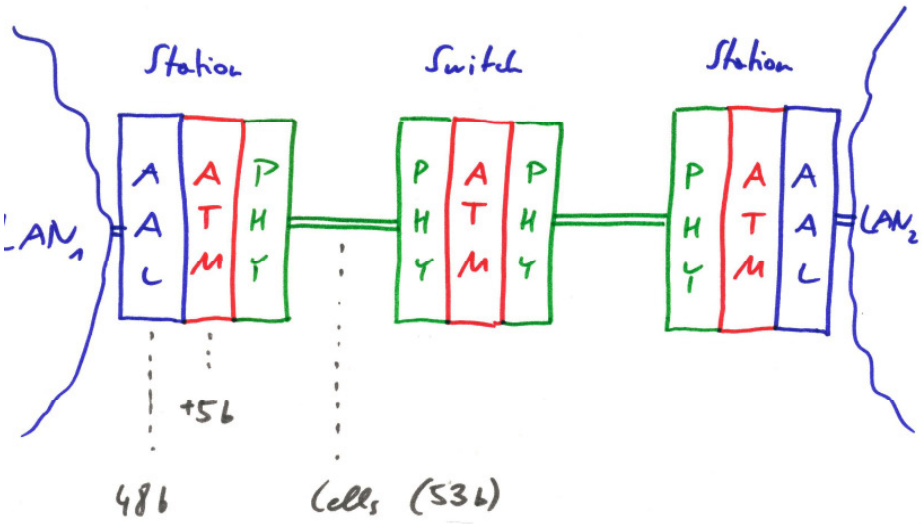
ATM hätte das universelle Netzwerksystem werden sollen (und wahrscheinlich auch können). Die Weiterentwicklung des Ethernet von derzeit 1 Gbps zu bald 10 Gbps stellt aber die Verwendung von ATM als LAN Technologie immer weiter in Frage. ATM ist bisher nicht über 622 Mbps hinaus implementiert worden. 1 Gbps Ethernet-Karten gibt es aber heute schon um wenige tausend Schilling bei Elektronik-Versandhäusern zu kaufen. Außerdem steht 2002 das 10Gbps Ethernet vor der Tür. Die prinzipiellen Vorteile des ATM (QoS, isochrone Übertragung) werden im Ethernet anders gelöst (Zitat: "throw bandwidth at QoS"), nämlich indem man einfach wesentlich mehr an Transferleistung zur Verfügung stellt als gerade gebraucht wird und hofft, daß damit die QoS ohnehin "ohne spezielles Zutun" eingehalten wird (eher optimistischer Ansatz. Anm d. Autors).

Die Anpassung des ATM verschiedene andere Technologien erfolgt via ATM Adaptation Layer ("AAL"):

AAL	**1**	**2**	**3+4**	**5**
Konstante Transferrate	Ja	Nein	Nein	Nein
Isochrone Übertragung	Ja	Ja	Ja	Nein
Anwendungs-bereiche	Telefonie via ISDN / ATM	Gepuffertes Video, Audio	X.25, UDP, TCP	LANs

Tabelle ATM Adaptation Layer

Bei der Integration von LANs in ATM Netze kommt der AAL 5 zum Einsatz.



Skizze ATM AAL Architektur

SONET

Synchronous Optical Network

SONET ist ein US-Standard für ein ringförmiges optisches Netzwerk oder Punkt-zu-Punkt Verbindung (Switched Circuit). Es werden sogenannte Optical Carrier definiert:

OC-1:	810 ISDN-B-Kanäle = 810 x 64kbps
OC-3:	155 Mbps = bei ATM oft eingesetzt
OC-12:	622 Mbps = bei ATM oft eingesetzt
OC-192:	Leistungsfähig genug, um die Daten eines 10Gbps Ethernet über weite Strecken zu transportieren
OC-768:	40 Gbps, der derzeit schnellste digitale Transfer von Daten auf weite Strecken, im Backbone-Bereich des Internets eingesetzt.

Tabelle: Optical Carrier

WLANs („WiFi“)

Folgt später…

WPANs (Bluetooth)

Folgt später…

SANs

Folgt später ...

IEEE 802.2 LLC

Der “Logical Link Control”. (“LLC”, IEEE-802.2) ist die Verbindung zwischen den IEEE-802.x MAC-Normen und den “Upper Layers” (den Protokollen der OSI-Schicht 3). Ziel ist die einheitliche Schnittstelle aller IEEE-802.x Normen “nach oben".

Type 1:	Der LLC unterstützt Datagramme.
Type 2:	Der LLC unterstützt Verbindungen. Da diese normalerweise von den IEEE-802.x Normen im MAC nicht zur Verfügung gestellt werden, erfolgt die Implementierung im LLC selbst mittels des Protokolls “HDLC” (“Highlevel Data Link Control”), einer Variante des IBM-Protokolls “SDLC” (“Synchronous Data Link Control”). HDLC realisiert einen “Sliding Window” Mechanismus und stellt damit Flußkontrolle, Wiederholung im Fehlerfall, garantierte Reihenfolge und einen kontinuierlichen Bitstrom sicher.
Type 3:	Acknowledged Datagram. Wie Type 1, nur wird jedes Datagramm mit einem Antwort-Datagramm bestätigt.

Tabelle: LLC Typen

Daraufhin wurden die folgenden LLC-Klassen definiert:

Implementiert	**Type-1**	**Type-2**	**Type-3**
Class-1	J	N	N
Class-2	J	J	N
Class-3	J	N	J
Class-4	J	J	J

Tabelle LLC Klassen

Internetworking

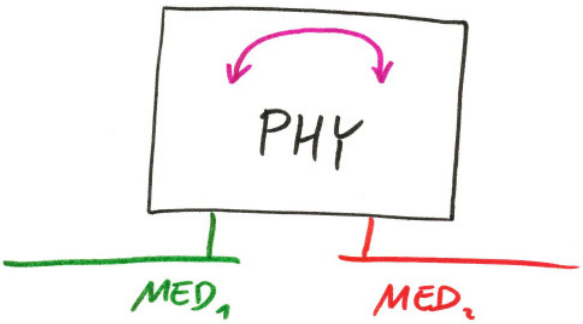
Unter Internetworking versteht man das Zusammenfügen von einzelnen Netzwerk-Segmenten zu zusammenhängenden Einheiten. Dieses Zusammenfügen kann auf unterschiedlichen OSI-Schichten erfolgen und hat damit verbunden gewisse Vorteile, aber auch zu beachtende Konsequenzen.

Repeater

Die einfachste Form, Netzwerksegmente zusammenzufügen, ist durch den Einsatz von Repeatern gegeben. Repeater sind bitweise, bidirektionale Empfänger-Signaldekoder-Sender. Sie kommen meist bei Ethernet zum Einsatz, wesentlich seltener auch bei anderen Topologien (und dort gewöhnlich nur, um außergewöhnlich große Entfernungen zu überbrücken).

Mit Repeatern verbundene Netze dürfen keine Zyklen enthalten, ansonsten kreisen Pakete “ewig”.

Im OSI-Schichtenmodell sieht ein Repeater von seiner internen Architektur her gesehen folgend aus:



Skizze Repeater

Vorteile:

Einfach in der Fertigung

Schnell (arbeitet so schnell, wie die Netze, die er verbindet, also mit “Medium Speed”)

Heutzutage sehr kostengünstig

Konfigurationsfrei und "manageable" (SNMP)

Nachteile:

Repeater eignen sich nur für Halbduplex-MACs. Sie lassen (im Falle von CSMA/CD) auch Kollisionen durch, sind also Bestandteil des “Collision Windows”.

Repeater können nur Netze verbinden, die von der Schicht 1 bis zur Schicht 7 hinauf ident sind. Nur die Schicht 0 (das ist eigentlich das Übertragungsmedium) kann auf den beiden Seiten eines Repeaters anders aussehen. Ein Repeater kann damit zB 10 Mbps Ethernet auf UTP in 10 Mbps Ethernet auf LWL “umformen”.

Bei Ethernet unterliegen Repeater zusätzlich der “5-4-3 Regel”. Diese besagt, daß auf dem Weg von einer Station im Netz zu einer anderen maximal 5 Netz-Segmente, verbunden duch maximal 4 Repeater liegen dürfen. Für den Fall des KoAx-Anschlusses im Ethernet (Standard-Ethernet 10BASE5 und Cheapernet 10BASE2) dürfen von diesen 5 Segmenten maximal 3 mit weiteren Stationen außer dem Repeater selbst belegt sein (sogenannte “populated segments”). Die beiden anderen Segmente sind dann nur sogenannte Verbindungs-Segmente, die dazu dienen, größere Strecken zu überbrücken, und dürfen nicht mit Stationen belegt werden (“unpopulated segments”). Im Falle von 10Mbps TP-Ethernet gilt nach wie vor sinngemäß die 5-4 Regel.

Für 100BASEx gibt es signifikante Einschränkungen wegen des gegenüber 10BASEx nur noch 1/10 so großen Collision Windows. Daher wurden im 100BASEx zwei Typen von Repeatern definiert: Type 1 ist langsamer, für diese Typen gilt die 2-1 Regel. Type 2 ist intern schneller, hier gilt die 3-2 Regel.

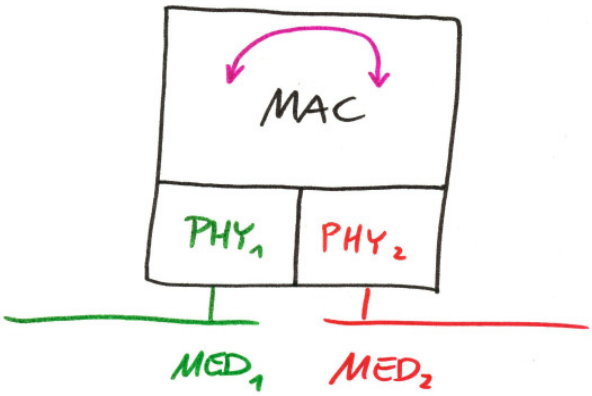
Heute kommen zumeist sogenannte Multiport-Repeater zum Einsatz, für die sich der Name “Hub” eingebürgert hat. Ein Multiport-Repeater verfügt über 4, 8, 16, 24, 32 etc sogenannte “Ports”, das sind die Ein/Ausgänge des Repeaters. An jedem solchen Port kann ein Netzwerk-Segment angeschlossen werden. Zusätzlich verfügen viele Hubs noch über einen sogenannten “Uplink-Port”.

Dieser dient der Verbindung zum nächsten übergeordneten Repeater und ist oft nur mit "ausgekreuzten" Kabeln verwendbar. Manchmal ist die Auskreuzung des Ports über einen Schalter am Repeater zu- bw abschaltbar.

Zusätzlich können spezielle Repeater mit unterschiedlichen Medien-Anschlüssen versehen sein. Typisch sind zB Repeater mit n*8 TP Ports und einem LWL-Port für den Uplink. Diese "Multimedia-Repeater" (wie sie in der deutschsprachigen Literatur bezeichnet werden) waren speziell früher interessant, als es noch viel KoAx Ethernet gab und gerade der Umstieg auf TP bzw noch später auf LWL erfolgte. Damit war ein einfacher und kostengünstiger Upgrade-Weg zu neuen Kabeltechnologien möglich.

Bridge

Eine Bridge arbeitet auf der OSI-Schicht 2. Sie empfängt, interpretiert und versendet ganze Pakete – im Gegensatz zum Repeater, der mit den Paketen der OSI-Schicht 1 – den einzelnen Bits – arbeitet. Mit Bridges verbundene Netze dürfen ebenfalls keine Zyklen (=redundante Leitungen) enthalten, ansonsten kreisen Pakete "ewig". Ausgenommen davon sind Spanning Tree Bridges und Source Routing Bridges. Bei diesen - und nur bei diesen - Sonderformen dürfen Zyklen physisch realisiert werden.



Skizze Bridge intern

Vorteile:

Eine Bridge arbeitet auf dem MAC Niveau (Schicht 2), sie ist daher für alle höheren Protokolle transparent (unsichtbar).

Die Bridge puffert Pakete. Diese werden daher auch zumeist auf Fehler im Frame (falsche CRC, unvollständige Pakete) untersucht und gegebenenfalls ausgesondert. Der Puffer hat auch eine gewisse Speicherwirkung, um Spitzen auf den Eingangsports aufzufangen.

Bridges wirken durch die Pufferung der Pakete auch als Filter gegenüber Kollisionen beim Ethernet. Kollisionen werden von den Bridges aufgefangen und nicht weitertransportiert (im Allgemeinen, siehe jedoch auch unter "Switches"). Damit können Bridges zur Performance-Steigerung von überlasteten Netzwerk-Segmenten verwendet werden.

Auch um ein LANs geografisch "größer" zu machen als es ein mit Repeatern verbundenes Netz sein könnte, ist eine Fähigkeit der Bridge. Da für eine Bridge die Repeater-Regeln nicht gelten, können Bridges fast beliebig verwendet werden und man kann damit das LAN

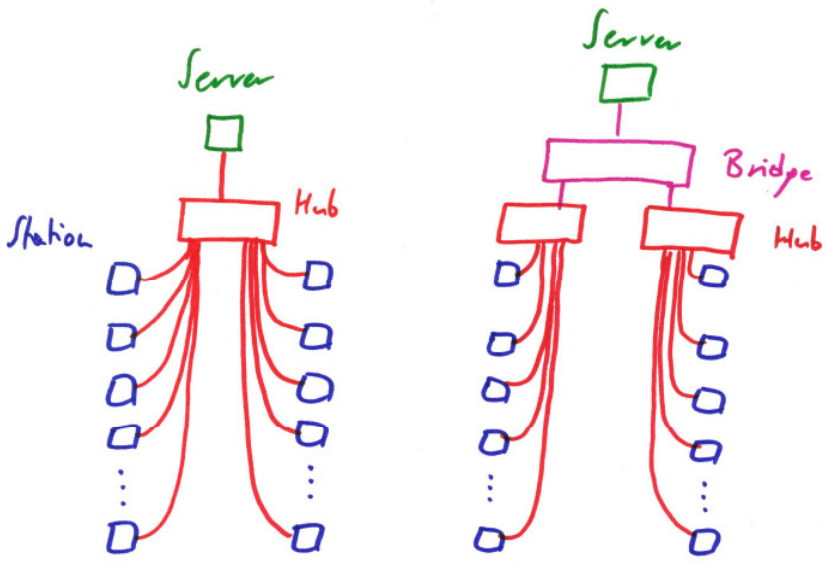
wesentlich vergrößern. Dennoch sind große rein gebridgte Netze nicht sehr parktikabel, da es ihnen an Redundanz und Lastverteilungs-Fähigkeiten fehlt. Dazu muß dann aber ein Router verwendet werden.

Bridges sind auf der OSI-Schicht 2 angesiedelt und können daher Netze, deren Schicht 1 unterschiedlich sind, miteinander verbinden. Damit sind mit Bridges auch Netzwerke unterschiedlicher Geschwindigkeit verbindbar. ZB 10 Mbps Ethernet mit 100 Mbps Ethernet. Die Konvertierung von zB Token Ring auf FDDI ist mit einer Bridge nach unserer Definition der Bridge aber nicht möglich. Dies wird aber von einigen Herstellern unter dem Namen “Translational Bridge” verkauft. Translationale Bridges sind nach unserer Definition Router.

Bridges lassen Broadcasts ungehindert durch, so wie jedes andere Paket auch. Sie können daher nicht zum Einschränken der Broadcast auf eine bestimmte Anzahl von Stationen verwendet werden. Broadcasts können aber in einem großen Netz einen bedeutenden Anteil der Pakete darstellen.

Bridges entlasten Netzwerksegmente. Wenn auf einem Segment zuviele Stationen angeschaltet sind, kann man per Bridge dieses Segment in zwei einzelne Teil-Segmente aufsplitten. Wenn man diese beiden Segmente stattdessen per Repeater verbinden würde, wäre nichts gewonnen. Wenn man allerdings per Bridge trennt, dann hat man zwei Segmente mit jeweils halber Stationsanzahl und damit pro Segment das halbe Datenaufkommen. Dies ist speziell bei Ethernet und dann von entscheidender Bedeutung, wenn das Segment vorher bereits im Bereich der Sättigung gearbeitet hat.

Dieses Verhalten der Bridge gilt allerdings nur dann, wenn man von einem Client/Server Modell im Netz ausgeht. Hierbei fließen die Daten primär zwischen den Clients und einem einzigen Server. Wenn man nun das Netz zwischen Bridge und Server verstärkt und die beiden Subnetze durch die Bridge trennt, so wird die gesamte gewünschte Last auf die beiden Netze verteilt. Bei Ethernet hat eine Entlastung aber eine deutliche (überproportionale) Reduktion der Kollisionen und damit der Verzögerung zur Folge, was eben der gewünschte Effekt ist.



Skizze Lasttrennung durch eine Bridge in einem Client/Server Netz
links: ein einzelnes Netz
rechts: zwei Teilnetze, per Bridge verbunden

Diverse Sonderformen von Bridges sind ebenfalls verfügbar (siehe weiter unten).

Nachteile:

Bridges benötigen CPU und Memory. Sie puffern Datenpakete im Speicher. Sie sind „kleine Computer“ und daher teurer als Repeater.

Die Geschwindigkeit einer Bridge wird in “Frames per Second” (“**fps**”) gemessen. Hierbei muß zwischen Eingangsgeschwindigkeit (“in” oder “filtering”) und Ausgangsgeschwindigkeit (“out” oder “forwarding”) unterschieden werden. Diese können unterschiedlich sein und sind es normalerweise auch. Bridges können maximal so schnell sein wie Repeater, sind normalerweise aber langsamer.

Bridges dürfen auch vollständige und korrekte Pakete verwerfen. Sie tun dies speziell dann, wenn zuviele Pakete zugleich hereinkommen bzw die Ausgänge nicht rasch genug Daten wegschicken können.

Sonderformen von Bridges

Firewall

Hierbei wird die Bridge intern mit einer Tabelle von MAC-Adressen + Ports versehen. Diese Tabelle wird beim Weiterreichen von Paketen konsultiert. Ist eine MAC-Adresse nicht eingetragen (bzw eingetragen im Falle einer Sperrliste) wird das Datenpake von / an diese MAC-Adresse nicht weitergeleitet.

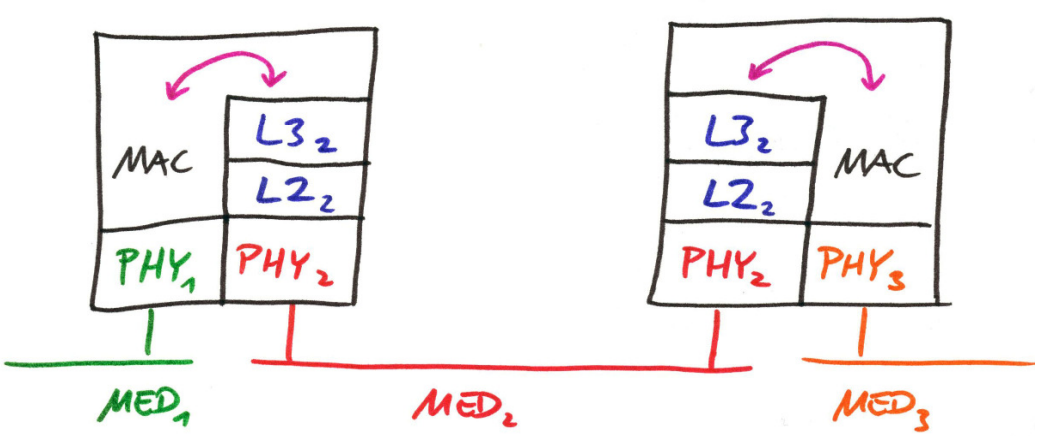
Damit ist eine – allerdings sehr triviale und unbequem zu administrierende – Firewall-Funktionalität realisierbar. Von Nachteil ist speziell, daß die MAC-Adressen in die Bridge konfiguriert werden müssen. Ändert sich diese, muß in der Bridge die Änderung entsprechend nachgezogen werden. Da sich MAC-Adressen aber recht oft ändern können, ist eine Firewall auf Bridge-Ebene einfach unhandlich.

Bridge mit Paketpriorisierung

Hier wird wie bei der Firewall vorgegangen, aber der Datenstrom von oder an bestimmte MAC-Adressen bevorzugt behandelt. Dies ist bei einer Bridge insofern sinnvoll, als im Falle hoher Last die Datenpakete im Puffer der Bridge priorisiert versendet werden können. Im Falle geringer Last werden ohnehin alle Datenpakete sofort weitergeleitet.

Remote Bridge

Wenn eine Bridge vertikal “in zwei Hälften” zerteilt wird, und man die entstandene “Innenseite” mit einer anderen Netzwerktechnologie versieht, entsteht eine Remote Bridge. Diese fungiert mit ihren “Außenseiten” wie eine Bridge, “innen” jedoch wird eine völlig andere Form der Datenübertragung verwendet.



Skizze Remote Bridge

Remote Bridges werden verwendet, um ein LAN über ein MAN/WAN mit einem zweiten LAN zu verbinden. ZB Ethernet über "leased lines", Frame Relay oder über ISDN. Aus Gründen der Redundanz ist aber oft ein Router die bessere Wahl.

Filterbridge

Hierbei wird ein Filteralgorithmus in der Bridge implementiert. Die Bridge verfügt intern über eine Tabelle T mit 3 Spalten:

MAC-Adresse Tm, Port Tp, Timeout Tt

Der Algorithmus innerhalb der Bridge läuft folgend ab:

Filter-Phase:	Wenn die Bridge ein Datenpaket p über einen Port q erhält, so schaut sie in ihrer internen Tabelle T nach, ob die Destination-MAC-Adresse des Paketes (Pm) in der Tabelle (Spalte Tm) enthalten ist. Ist dies der Fall, so wird das Paket gezielt an denjenigen Port weitergesendet, der in der Spalte Tp in der gefundenen Zeile in der Tabelle angegeben ist. Wird in der Tabelle nichts gefunden, wird das Paket auf alle Ports außer auf q ausgegeben (Defaultfall). Auf den Port q wird das Paket nie ausgegeben, da es ja von diesem Segment stammt.
Learn-Phase:	Jedesmal, wenn die Bridge ein Paket p erhält, schaut sie in der Tabelle T nach, ob es einen Eintrag mit dieser Source-MAC-Adresse Tm bereits gibt. Gibt es ihn bereits, so wird der Eintrag in der Tabelle überschrieben mit der Portnummer q, der MAC-Adresse p und einem Timeout von 300 Sekunden. Ist er noch nicht in der Tabelle enthalten, so wird er mit diesen Werten eingefügt.

Tabelle Filter and Forward

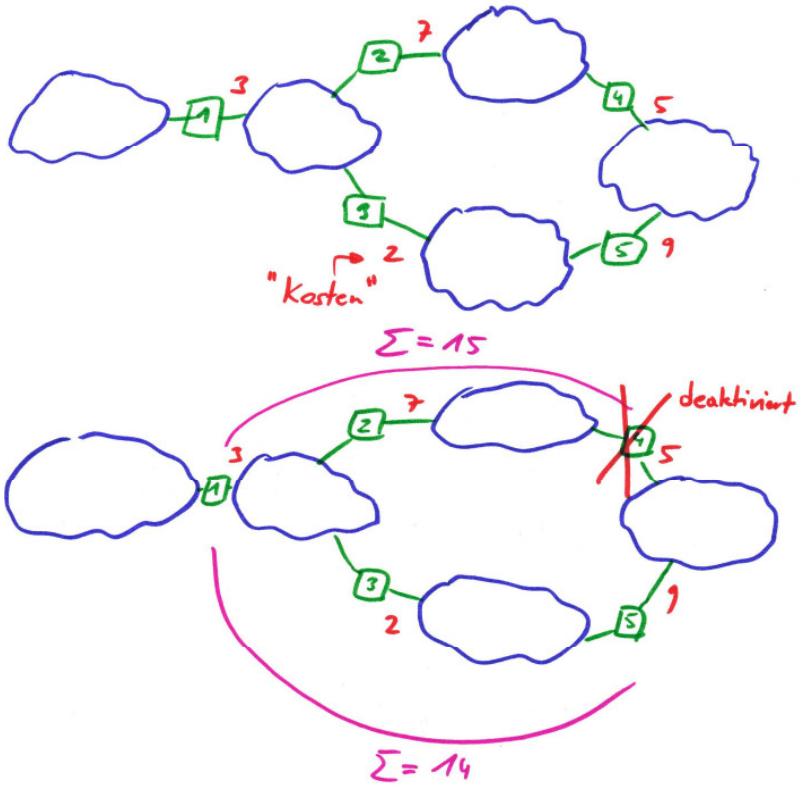
Dieser Algorithmus führt dazu, daß die Bridge "lernt", an welchen Ports sich Stationen mit welchen MAC-Adressen befinden. Sie nutzt dieses Wissen, um Frames gezielt nur an bestimmte Ports weiterzuleiten und damit die Netzlast zu senken. Wenn die Bridge von einer Station noch nichts "gehört" hat, so werden Frames, die an diese Station adressiert sind, auf allen Ports weitergeleitet. In dem Moment, in dem diese Station das erste Mal ein Paket abgesendet hat, wissen alle Bridges im Netz, wohin ein Frame an diese Station gezielt weitergeleitet werden muß. Das Timeout dient dazu, daß eine Station, die das Segment wechselt (zB ein Notebook), nach einigen Minuten korrekt mit Daten beliefert wird, auch wenn sie (wieder Erwarten) keine Frames in dieser Zeit abschickt.

Spanning Tree Bridge

Diese Sonderform implementiert einen eigenen Algorithmus, den Spanning Tree Algorithmus. Dieser stellt sicher, daß in einem Netz keine logischen Zyklen entstehen, physische Zyklen sind dagegen erlaubt. Der Einsatzzweck dieser Sonderform liegt darin, daß man physische Redundanz einbaut, um den Ausfall von Segmenten oder Bridges zu korrigieren. Die definierende Norm ist IEEE 802.1d. Spanning Tree Bridges kommen hauptsächlich bei Ethernet zum Einsatz.

In einem Spanning Tree Netz darf logisch auch immer nur exakt ein Weg von jeder Station im Netz zu jeder anderen führen. Redundante Wege (die damit auch zu Zyklen im Netz führen) werden logisch deaktiviert, bleiben aber physisch verbunden. Um diese Deaktivierung (genannt “Standby”) von Bridges zu erreichen, müssen die Bridges untereinander kommunizieren. Daher hat jede Spanning Tree Bridge eine MAC-Adresse und kann damit von einer anderen Spanning Tree Bridge kontaktiert werden. Der Aufbau des Spanning Trees wird mittels Datenpaketen (sogenannte “Bridge-PDUs” oder “BPDUs”) erreicht. Diese Datenpakete werden zwischen den Bridges ausgetauscht (MAC-Multicast auf Adresse 01:80:C2:00:00:10), ohne daß eine Station etwas davon merkt. Sie kosten daher auch Netzleistung (allerdings sehr wenig). Die BPDUs müssen in regelmäßigen Abständen (30 Sekunden bis einige Minuten) ausgetauscht werden, damit die Bridges den Ausfall anderer Bridges oder von Kabelsegmenten rasch erkennen und beheben können.

Die Zeit für die Rekonfiguration im Falle eines Bridge-Ausfalls beträgt daher ca 30 bis 60 Sekunden. Dies ist für heutige Anwendungen oft bereits zu viel. In IEEE 802.1w ist eine schnellere Version des Spanning Tree Algorithmus (RSTP „Rapid Spanning Tree Protocol“) definiert worden, deren Umschaltdauer ca 1 Sekunde beträgt und das abwärtskompatibel zu IEEE 802.1d ist.



Skizze Konfiguration eines Spanning Trees

Daher kommt auch die alternative Bezeichnung “Transparent Bridging”. Die einzelnen Stationen

brauchen weder eine eigene Software installiert noch eine spezielle Hardware. Die gesamte Arbeit wird von den Bridges erledigt. Es ist daher sehr einfach möglich, in bestehende Ethernet-Netzwerke Spanning Tree Bridges im nachhinein zu integrieren.

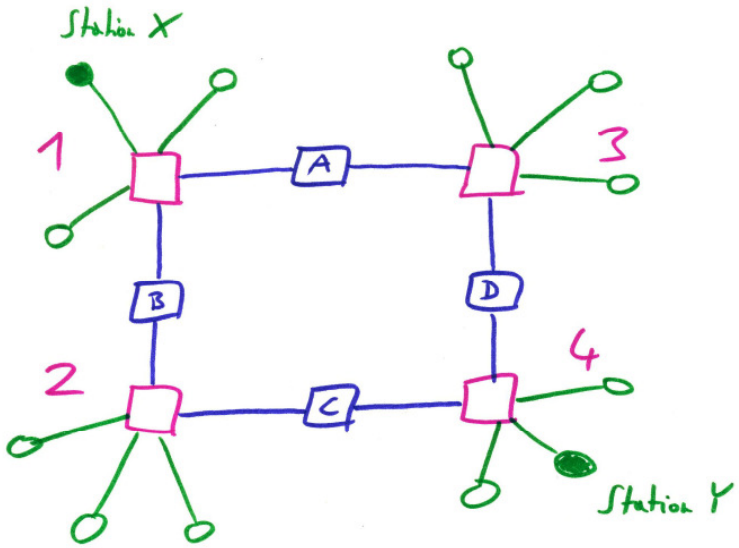
Source Routing Bridge

Diese Sonderform erlaubt ebenfalls Zyklen im Netz, nutzt aber neben der Redundanz auch die Möglichkeit, in limitiertem Umfang Lastverteilung zu implementieren. Source Routing Bridges kommen im Bereich des Token Rings vor und sind in der Token Ring Norm IEEE 802.5d definiert.

Der Source Routing Algorithmus ist insofern nicht transparent, als man in den einzelnen Stationen eigene Software installieren muß, die das Source Routing implementiert. Source Routing ist damit ein Vorgang, an dem nicht nur die Bridges beteiligt sind, sondern in dem auch die Stationen eine aktive Rolle spielen.

Wenn eine Station im Token Ring einen Frame an eine andere Station sendet, so addressiert sie diese zuerst einmal in den lokalen Ring hinein. Dieses Paket wird von einer Bridge nicht weitergeleitet und bleibt daher auf den lokalen Ring beschränkt. Ein solches Paket hat den Routing-Typ “null”.

Wenn der Empfänger nicht antwortet, so ist er entweder nicht aktiv oder “jenseits” einer Bridge. In diesem Falle wird ein Paket mit dem Routing-Typ “all routes broadcast” versendet, eine Art “Pfadfinderpaket”. Genauer gesagt wird eine XID (“Exchange ID”) PDU versendet. Diese wird von allen Bridges auf allen Ports weitergeleitet. Vorher trägt aber die Bridge in jedem von ihr weggehenden Datenpaket ihre eigene Kennung (die sogenannte Bridge-ID) und die Kennung des Rings (Ring-ID), auf dem das Paket weitergesendet wird, ein. Diese Routing-Information wird im Info-Teil des Frames gesammelt und wird bei jeder Bridge, die ein Frame “überquert”, um einen Eintrag (Bridge-ID+Ring-ID) länger.

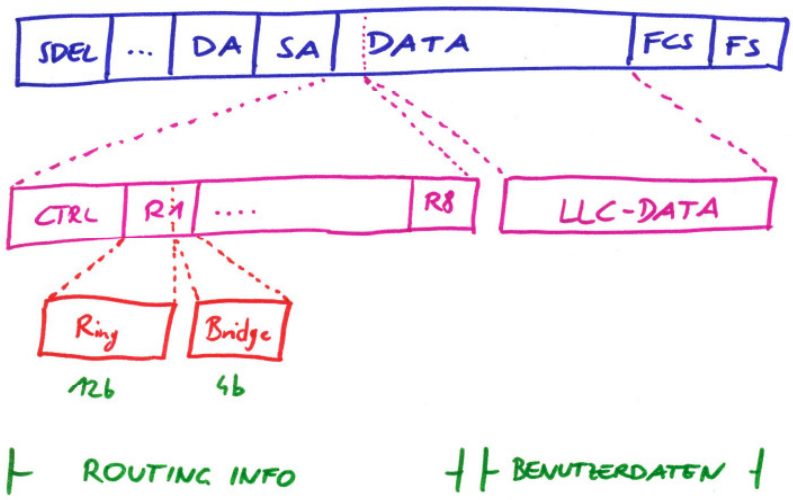


Skizze SR Netz

Kommt der Frame bei einer Bridge an und findet diese Bridge ihre eigene Bridge-ID im Routing-Teil des Frames, so wird das Paket vernichtet und nicht weitergesendet, da es sich um ein kreisendes Paket handelt.

Kommt der Frame bei der Zielstation vorbei, so dreht diese die Routing-Information um, d.h. die letzten Einträge in der Routing-Information werden nun die ersten und umgekehrt. Dies erfolgt nicht durch wirkliches Sortieren des Routing-Inforrmations-Feldes im Frame, sondern es gibt ein eigenes Bit, das die Auslese-Richtung der Routing-Information bestimmt. Die Zielstation setzt dieses "Direction-Bit" um, kopiert die Routing-Information ein zweites Mal in den Frame (warum, sehen wir gleich) und sendet das Paket nun mit dem Routing-Typ "non-broadcast" an die Quellstation zurück.

Auf dem Weg durch einen Ring wird dieser Frame nur dann von einer Bridge weitergeleitet, wenn diese Bridge ihre Bridge-ID in der Routing-Information des Frames vorfindet, und zwar an der "ersten" Stelle. In diesem Fall wird dieser Routing-Eintrag aus dem Frame entfernt, und die Bridge sendet den Frame in denjenigen Ring weiter, der in der Routing-Information als Ring-ID angegeben ist. So findet das "non-broadcast" Paket auf exakt einem Weg zurück zum Sender. Wenn es bei diesem ankommt, ist die Routing-Information des Paketes dann leer. Alle Pakete, die bei der Zielstation eintreffen, werden wieder zur Quellstation zurückgesendet. Die Quellstation entscheidet sich dann für eines der angekommenen Pakete und liest dessen (zweite, kopierte) Routing-Information aus. Diese Route bleibt dann als "fixe" Route zu dieser Zielstation gespeichert.



Skizze Source Routing: Routing Informationen
bis zu 8 "Hops" können in der Routing Info eines 802.5d Pakets gespeichert werden
das SA-Feld zeigt das Vorhandensein der Routing-Info an
das CTRL-Feld steuert das Routing, es enthält unter anderem die Broadcast-Type und das „Direction"-Bit

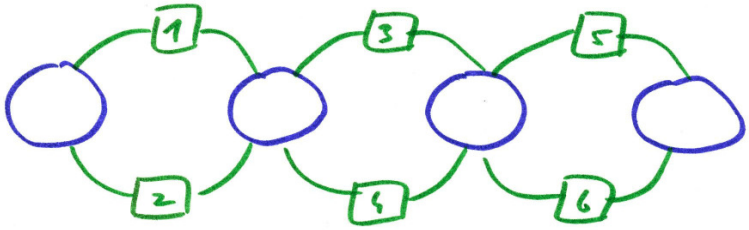
Bleibt noch die Frage, welche Route die Quellstation nehmen soll. Dafür gibt es zwei naheliegende Kriterien: Entweder das erste "non-broadcast" XID-Paket, das zurückkommt, wird genommen. Dieses hat offensichtlich die "schnellste" Route gefunden. Oder dasjenige "non-broadcast" XID-Paket wird verwendet, das die kürzeste Routing-Information angesammelt hat. Dieses Paket hat die "kürzeste" Route gefunden.

Der Vorteil des Source Route Bridging liegt sicherlich in seiner Fähigkeit, eine gewisse Lastverteilung zu erzielen und immer alle möglichen Wege zum Ziel in Betracht zu ziehen. Ist einmal eine Route festgelegt, so wird diese gespeichert und immer wieder verwendet. Die Lastverteilung erfolgt also nur ein einziges mal, nämlich bei der ersten "Wegsuche". Fällt eine Leitung oder Bridge aus, so wird eine neue Route gesucht.

Wenn nun die Quellstation zu einer Zielstation ein Datenpaket senden will, und die Route zur

Zielstation bekannt ist, so kann ein “non-broadcast”-Frame verwendet werden. In diesen wird nun vor den eigentlichen Nutzdaten die Routing-Information hineingestellt. Zusätzlich wird in der Source-Adresse (“SA”) des Frames (das ist die 48 Bit MAC Adresse der Token Ring Karte) das erste Bit gesetzt. Dieses Bit ist bei Source-Adressen immer Null, da die Definition der Adresse besagt, daß eine Eins als erstes Bit einer Adresse eine Gruppe adressiert und keine Station. Dieses Bit macht aber in der Source-Adresse keinen Sinn (die Quelle eines MAC Frames kann nie eine Gruppe sein, sondern nur eine einzelne Station) und wird daher hier mißbraucht, um den Bridges zu sagen, daß der erste Teil der Nutzdaten des Frames Routing-Informationen trägt.

Vorsicht ist bei ungünstigen Topologien gegeben, da die Anzahl der XID-Pakete rasch anwachsen kann. Wenn man zB die Bridges folgend anordnet



Skizze Der "2 hoch n"-Effekt ungünstiger Source-Routing-Topologien

dann beträgt die Anzahl der XID-Pakete, die beim Ziel ankommen, 2 hoch (Anzahl der Bridges).

Switch

Als weitere Sonderform von Bridges muß auch der Switch angeführt werden. Ein Switch ist eine Bridge, die intern über besondere Mechanismen verfügt, um effizient und mit möglichst geringer Latenz Pakete weiterzuleiten. So gut wie alle heute produzierten Bridges sind intern als Switches ausgelegt und werden auch unter dem Namen „Switch“ vermarktet.

Zumeist wird dafür eine Art von “**Crossbar Switch**” (Kreuzschalter) verwendet, der dieser Sonderform auch den Namen gegeben hat. Dieser Mechanismus hat das Ziel, mehrere Pakete zugleich (natürlich von verschiedenen Ports) empfangen und auch zugleich (auf verschiedenen Ports) weiterleiten zu können (setzt damit die Funktionalität der Filterbridge voraus). Der Unterschied zur Bridge ist also primär die Parallelität der Verarbeitung der Frames.

Anstelle des teuren Crossbar Switches wird oft auch ein hoch **performantes Bussystem** verwendet. Hierbei wird ein interner Pufferspeicher (im Bereich einiger Mbit) verwendet, in dem die Datenpakete vor dem Weiterreichen ganz oder teilweise zwischengelagert werden. Wenn die Bandbreite des im Switch verwendeten Bussystems größer als die Summe der maximalen Bandbreiten aller Ports des Switches beträgt, so sollte es (theoretisch) ebenfalls zu keinen Engpässen kommen.

Wenn die interne Bandbreite des Switches größer als die Summe aller Bandbreiten der Eingangsports ist, so arbeitet der Switch auf „wire speed“. Er muß dann keine Datenpakete mehr eingangsseitig verwerfen. Ausgangsseitig kann und darf der sehr wohl Datenpakete verwerfen, wenn zB ein Server sich nicht schnell genung die Daten aus dem Switch abholen kann, so ist das nicht die Schuld des Switches. Ein 8 Port 100Mbps Fast Ethernet Switch sollte also 800Mbps an Daten eingangsseitig bearbeiten und puffern können, damit er non-blocking ist. Die fps hängt dagegen von der Bandbreite des Switches und von der minimalen Paketgröße ab, bei Ethernet sind

das 64 Byte (512 bit) plus die minimal einzuhaltende Ruhepause zwischen zwei Frames (der sogenannte „Inter Frame Gap“, 96 Bits). So erhalten wir 100.000.000 / (512 + Overhead) = ca 150.000 fps theoretisch maximal. Bei acht Ports macht das ca 1,2 Mio fps. Schafft der Switch diesen Durchsatz, so ist er praktisch so schnell wie ein Repeater, schafft also auch „wire speed“. Besonders beachten muß man hierbei modular aufgebaute Switches („stackable switches“), da bei diesen fast immer ein Bussystem als „Backbone“ oder „Backplane“ zum Einsatz kommt und dieses dann die Summe der Eingangsdaten aller angeschlossenen Switches zugleich verkraften können muß, um „wire speed“ fähig zu sein. Da bei 100Mbps Ethernet oder noch schnelleren Varianten die theoretischen Datenraten bald sehr hoch werden, sind nur teure Switches in der Lage, tatsächlich mit „wire speed“ zu arbeiten. Generell gilt es jedenfalls herauszufinden, ob durch geschickte Verschaltung nicht doch eine geringere Belastung der Switches möglich ist und man so mit einem kostengünstigeren Modell nicht auch das Auslangen finden kann.

Im Extremfall – bei den sogenannten Switches mit “cut through” – wird das Datenpaket nur soweit in den Puffer der Bridge eingelesen, bis die Source Address (SA) und die Destination Address (DA) im Puffer stehen. Dann bereits kann die Bridge entscheiden, auf welchen Ports das Paket weiterzuleiten ist (siehe dafür Filterbridge, Spanning Tree Bridge, etc), sie muß nicht warten, bis das ganze Paket eingetroffen ist. Die “cut through” Technik bringt primär eine Reduktion in der Bearbeitungszeit („Latenz“) des Paketes, da es bereits bei seinem Ausgangsport herausgesendet wird während noch am Eingangsport die letzten Bits des Paketes hereinströmen. Hier wird nicht mit der Methode “zuerst mal alles puffern, dann weitersenden” (engl: “store and forward”) gearbeitet, sondern es fließen - wie bei einer geschalteten Leitung - die Bits quasi hintereinander in einer Röhre von der Quelle zum Ziel.

Dem prinzipiellen Vorteil des “cut through” Switches, daß die Latenz auf einige wenige Bytes des Paketes plus der Verarbeitungszeit im Switch (bei gegenwärtiger Technologie beträgt diese ca 35µs) reduziert wird, steht ein gravierender Nachteil entgegen: bei dieser Methode kann es passieren, daß eine Kollision innerhalb des Collision Windows (512 bit) auftritt, aber vom Switch zu spät erkannt wird. Wenn nämlich der Switch bereits mit der Weiterleitung der Daten begonnen hat und erst dann die Kollision auf tritt, kann er klarerweise den bereits gesendeten Paketteil nicht mehr “zurücknehmen”. Damit wird die Kollision von der Bridge nicht mehr abgeblockt. Dasselbe gilt für Prüfsummenfehler und Pakete, die plötzlich “mittendrin” aufhören. Auch diese werden - so wie sie sind - weitergeleitet. Als Kompromiß kann der Switch die Latenz so wählen, daß er zumindest 512 Bit puffert und damit alle “korrekten” Kollisionen (also die, die innerhalb des Collision Windows auftreten) auch als solche erkennt und wegfiltert.

Die neueste Generation von „cut through“ Switches arbeitet pro Port initial mit der „cut through“ Technik, es wird aber die Korrektheit aller Frames über diesen Eingangsport mitprotokolliert (via Auswertung des CRC). Werden zuviele zerstörte Pakete erkannt, schaltet der Switch diesen Port auf „store and forward“ zurück.

Flow Control

Was kann ein Switch, der nicht „wire speed“ fähig ist, tun, um den eingangsseitigen Datenfluß zu verringern und sich damit Entlastung zu verschaffen? Hierfür bestehen die folgenden Möglichkeiten:

1. IEEE 802.3 Flow Control

Durch senden von Datenpaketen („Pause Frames“) an die Multicast-MAC-Adresse 01:80:C2:00:00:01 mit dem Type-Field auf „8808“ gesetzr teilt ein Switch (der hoffentlich darauf achtenden Netzwerkumgebung) seinen Überlastungszustand mit. Funktioniert nur im Fullduplex-

Modus, also nur wenn CSMA/CD außer Betrieb ist. Alle Gigabit Ethernet Hersteller und einige Fast Ethernet Hersteller unterstützen Flow Control.

2. Back Pressure (Jamming)

Ein Switch sendet einen Jam mitten in ein gerade empfangenes Paket hinein. Dadurch wird das Paket zerstört, der Sender hört sofort mit dem Senden auf, wenn das Jam innerhalb des Collision Windows angekommen ist. Damit kann ein Switch einen Datentransfer durch Abbruch verkürzen. Funktioniert auch bei Halbduplex-Betrieb (also bei CSMA/CD).

3. Head-of-Line Blocking

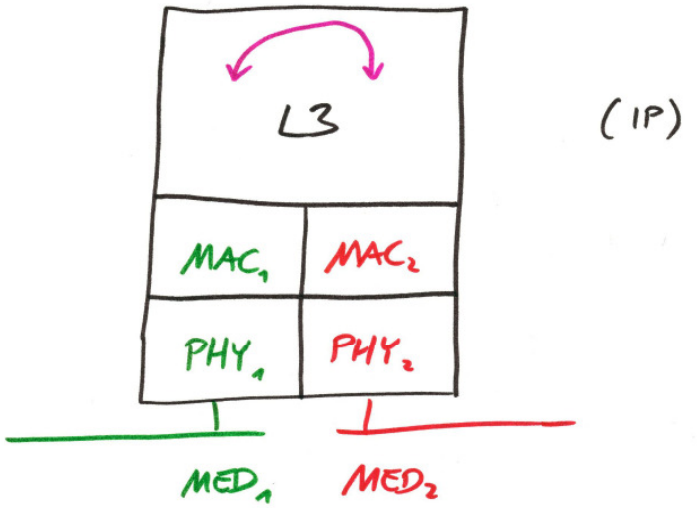
Der Switch verwirft Pakete, die an einen überlasteten Empfänger gerichtet sind. Dies erkennt er an den Paketen, die im Ausgangspuffer für einen bestimmten Port stehen. Staut es sich dort, so werden weitere Pakete an diesen Port verworfen. Dies entlastet den Eingangsport. Damit wird eine etwas rauhe Art der Flußkontrolle implementiert, ferner wird dem Überlaufen des internen Puffers im Switch entgegengewirkt. Das Verfahren funktioniert bei Voll- und auch bei Halbduplex-Betrieb.

4. Broadcast Storms

Wenn der Anteil der Broadcasts an der gesamten Netzlast einen bestimmten Wert übersteigt (meist 20%, einstellbar), verweigert der Switch das Weiterleiten der Broadcasts auf eine bestimmte Zeitspanne (meist 10 Sekunden, einstellbar).

Router

Router sind auf der OSI-Schicht 3 angesiedelt und erfordern daher, daß alle Schichten “nach oben” ab der Schicht 3 (inklusive) ident sind.



Skizze Router

Die charakteristische Eigenschaft eines Routers ist seine Fähigkeit, Datenpakete zu routen (daher auch sein Name), sie also durch Interpretation der Schicht-3-Adresse (im Beispiele TCP/IP also der IP-Adresse) einen Weg für das Paket zum Ziel zu finden. Zu diesem Zwecke bedienen sich alle Router einer **Routing-Tabelle**. In dieser steht der Weg des Datenpaketes zum Ziel under zumindest

zum nächsten Router, der dann diese Aufgabe übernimmt. Im Zusammenhang mit dem Routing tritt ein neuer Sachverhalt auf, der bisher bei den Internetworking-Geräten der Schichten 1 und 2 nicht aufgetreten ist, nämlich daß es gleichzeitig mehrere aktive Wege zum Ziel gibt. Dies ist bei allen Bridges (Ausnahme Source Routing Bridge) und bei den Repeatern nicht der Fall. Daher benötigen mit Repeatern bzw Bridges verbundene Netze auch kein Routing.

Wie gesagt mit der Ausnahme der Source Routing Bridge. Diese ist aber – wie ihr Name schon sagt – ein Zwitter zwischen Bridge und Router. Genaugenommen ist sie mehr ein Router. Aber die Tatsache, daß sie über keine Routing-Tabellen in den Bridges verfügt, macht sie auch irgendwie wieder Bridge-ähnlich.

Mit der Möglichkeit, den Datenpaketen mehrere Wege zum Ziel alternativ anzubieten, ergibt sich für den Router auch die Möglichkeit, diese Alternativen zugleich zu verwenden und damit eine **Lastaufteilung** (**load balancing**) durchzuführen. Zugleich geben mehrere Wege zum Ziel auch automatisch die **Redundanz** die man benötigt, um fehlertolerante Netze aufzubauen.

Durch die Notwendigkeit, Routingtabellen dynamisch aufzubauen, müssen Router auch untereinander kommunizieren und tauschen daher Routing-PDUs aus.

Die Geschwindigkeit von Routern wird in "**Packets per Second**" ("**pps**") gemessen. Sie sind aufgrund des Routing-Aufwands im Normalfall langsamer als Bridges. Durch die Reduktion der Netzwerk-Welt auf den Schichten 3 und 4 auf ein einziges Protokoll namens TCP/IP ist der Router heutzutage ein spezialisierter IP Router geworden. In früheren Tagen gab es wesentlich mehr routbare Protokolle (zB Novell IPX, Xerox Courier, Appletalk, etc).

Für sehr einfache Netzwerke bzw aus Sicherheitsgründen ist auch statisches Routing möglich. Dabei wird die Routingtabelle fix in die einzelnen Router geladen. Lastverteilung und Redundanz sind in diesem Falle nicht möglich. Unter Unix erfolgt die statische Routenwahl zB mit dem Kommando "route add …".

Für konkrete Beispiele zum IP-Routing siehe auch weiter unten im Kapitel TCP/IP.

B-Router

Dieser Zwitter zwischen einem Router und einer Bridge ist heute kaum noch im Einsatz. Er kann bestimmte Protokolle routen, andere dagegen nur bridgen. Durch die Reduktion der Protokollwelt der Schicht 3 auf IP spielt der BRouter keine wichtige Rolle mehr.

Routing Switch – Layer-n Switch – Content Switching

Die Bezeichnung Layer-n-Switch ist in unserer Nomenklatur generell falsch, außer bei n=2! Sie hat sich aber bereits fest eingebürgert.

Für n=3: Diese neuen Sonderform sind genaugenommen Router, die „so schnell wie ein Switch" arbeiten und ähnlich wie der Switch über viele Ports verfügt, sodaß man das lokale Netz statt mit Switches gleich mit Routern segmentieren kann. Eine andere gängige Bezeichnung für Routing Switches lautet „Layer-3 Switches".

Ebenfalls für n=3: Ein solcher Switch schaut sich die Inhalte des L3-Paketes an (zB IP, NetBEUI, IPX) und bildet automatisch eigene VLANs für alle die Hosts, die diese Protokolle jeweils verwenden. Multicasts und Broadcasts werden dann nur noch innerhalb dieser automatisch gebildeten VLANs weitergeleitet, was zu einer deutlichen Entlastung des Gesamtnetzes führt. Der Switch interpretiert also Schicht 3 Inhalte der Datenpakete und bildet die internen Tabellen für den

Filter and Forward Algorithmus anhand dieser Informationen.

Der Hintergrund dieser Technologie liegt in der Verwendung von „Application Specific Integrated Circuits“ (ASICs, also speziell gefertigten Chips), die mittels Hardware das erledigen, was bei einem konventionellen Router heute in Software abläuft: das Routing.

Es gilt daher, die internen Routing-Abläufe des IP-Protokolls und der dem IP-Protokoll zugeordneten Routing-Protokolle anstatt in Software in einem Chip in Hardware ablaufen zu lassen. Der damit verbundene Geschwindigkeitsvorteil ist enorm und man könnte das Routing in Hardware ablaufen lassen.

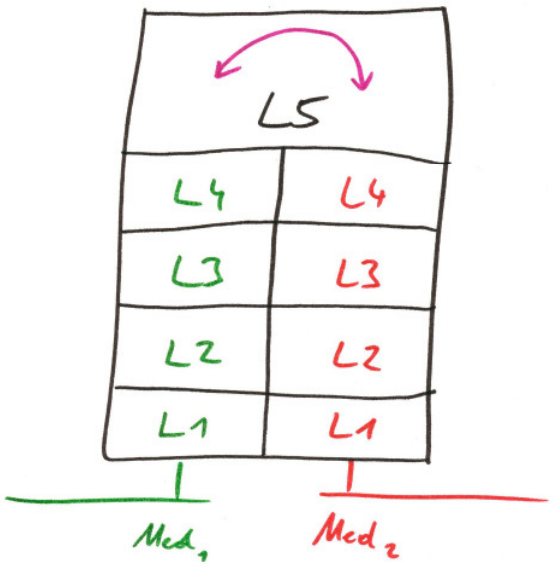
Diese Technologie wird auch gerade in den Storage Area Networks (parallel) unter der Bezeichnung „iSCSI“ („IP SCSI“) entwickelt. Hierbei wird der SCSI Datenstrom der Festplatte (oder eher des Storage Systems) in IP-Pakete umgewandelt und in Gbps Ethernet Pakete verpackt („gewrapt“). Dann erfolgt die Weiterverarbeitung der Pakete über normale Ethernet-Router und Switches bis zum Host.

Derzeit werden die Switching-Technologien in immer höhere OSI-Levels angehoben, man spricht daher von Layer-n-Switches, wobei n zwischen 2 und 7 liegen kann. Der Vorteil des Switchings auf höheren Layern ist die Effizienz, mit der das Bearbeiten der Pakete erfolgen kann. Der Nachteil liegt heute zumeist darin, daß die Bearbeitung der Informationen auf höheren Layern meist durch immer komplexere Software erfolgt und damit immer langsamer wird. Von einem Switch erwartet man sich aber gerade besondere Effizienz (bis hin zu „wire speed“), was also teilweise widersprüchliche Aufgabenstellungen sind. Erst die Konvertierung von Software in Hardware schafft hier Abhilfe.

Beispielsweise könnte ein Switch auch auf Layer-4 arbeiten. Beim Einsatz von TCP/IP switcht er also Datenpakete auf Basis der Informationen der Layer-4 Protokolle im TCP/IP-Stack, also des TCP oder des UDP. Hierbei können also Informationen mit in das Switching/Routing aufgenommen werden, die Layer-4 spezifisch sind, wie zB die Port-Nummer. Alle TCP-Datenpakete, die einen bestimmten Port beinhalten (zB Port 80 http), könnten an einen bestimmten Server oder gleich an eine bestimmte Serverfarm weitergeleitet werden. Damit funktioniert das Layer-4-Switching wie eine Art Virtuelles Netz: Daten werden aufgrund ihres Inhaltes (Content) an bestimmte virtuelle Adressen weitergeleitet. Dies kann bis zu Layer-7 (URLs, Cookies, etc) weitergehen.

Gateway

Als Gateways werden alle anderen Internetworking-Geräte bezeichnet, die weder Repeater, noch Bridges oder Router sind. Bei einem Gateway dürfen alle Schichten von 1 bis 7 unterschiedlich sein. Gateways werden zumeist dann eingesetzt, wenn eine Anwendungsprotokoll in ein anderes übersetzt werden soll.



Skizze Gateway (hier am Beispiel eines Schicht-5-Gateways)

Anwendungsbeispiele sind zB Mail-Protokolle. Konkret wird in einer Microsoft-Umgebung MS Exchange als Mailsystem verwendet. MS Exchange kann direkt (nativ) mit anderen MS Exchange Servern Mails austauschen, aber nicht direkt zB Internet-Mails (nach X.400 SMTP-Standard) versenden. Dazu wird ein Exchange-SMTP-Gateway verwendet. Hierbei sind alle Schichten *außer* der Schicht 7 auf beiden Seiten ident. Ein anderer Anwendungsfall wäre die Konvertierung von TCP/IP in ein anderes Protokoll, zB IBM SNA (System Network Architecture). SNA wird nach wie vor sehr oft im Mainframe-Bereich (S/390) als Netzwerkprotokoll verwendet. Um nun einem TCP-System Zugriff ins SNA zu geben, ist ebenfalls ein Gateway erforderlich (zB MS SNA Server).

Gateways sind im Normalfall teurer, komplizierter und auch langsamer als alle anderen Internetworking-Geräte. Mit ihnen kann man dafür auch komplett unterschiedliche Netzwerke verbinden.

LAN Management / Monitoring

Folgt später…

Windows NT 4.x / Windows 2000 (W2K)

Der Nachfolger des Windows NT 4.x wird Windows 2000 (kurz “W2K”) genannt. Windows 2000 ist gegenüber dem NT 4.x in wesentlichen Punkten überarbeitet worden. Speziell im Bereich Netzwerk sind die folgenden Punkte erwähnenswert:

Das Active Directory (AD) ersetzt das NT v4 Domain-Konzept
Integration von Kerberos Security
Driver Certification
Dynamisches DNS

Tabelle "New Features" des Windows 2000

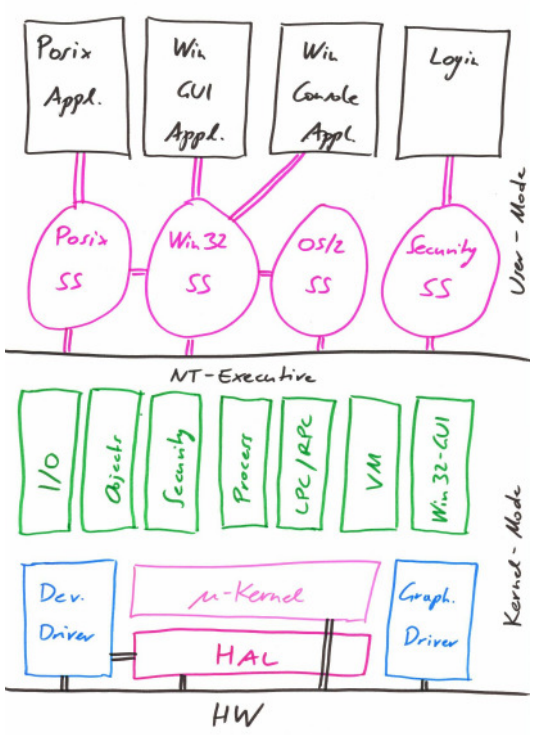
W2K gibt es in mehreren Ausführungen.

Windows 2000 Professional:
Für den Desktop-Einsatz
Nachfolger des Windows9x / ME / NT3.x / NT4.x
Bis zu 4GB RAM und 2 CPUs
Optimiert für „Mobile Usage", Synchroniserung und Hot Docking
USB, IrDA, IEEE 1394 („Fire Wire") Support
VPN-Support
Internet Connection Sharing, IE, IIS
Windows 2000 Server:
Application-, Web-, File- & Printserver
Nachfolger des NT3.x / NT4.x Servers bzw Enterprise Edition Servers
Bis zu 4GB RAM und 4 CPUs
Hat alle Features des W2K Professional plus:
Kernel Mode Schreibschutz
Anwendungs-Zertifizierung + DLL-Schutz
Active Directory Server
Router
RAS-Server
Dynamic DNS Server
Terminal Server
Distributed Filesystem
Disk Quotas
Hierarchisches Speicher-Management (HSM)
Windows 2000 Advanced Server:
Application-, Web-, File- & Printserver für "Enterprise Applications" und Datenbanken
Nachfolger des NT3.x / NT4.x Servers bzw Enterprise Edition Servers
Hat alle Features des W2K Server plus:

Bis zu 8GB RAM und 8 CPUs
Clustering bis 2 Knoten
Network Load Balancing
Windows 2000 Datacenter Server:
Application-, Web-, File- & Printserver für “Enterprise Applications” und Datenbanken
Nachfolger des NT3.x / NT4.x Servers bzw Enterprise Edition Servers
Hat alle Features des W2K Advanced Server plus:
Bis zu 64GB RAM und 32 CPUs
Clustering bis 4 Knoten
Wird nur zusammen mit Hardware und Professional Services vertrieben

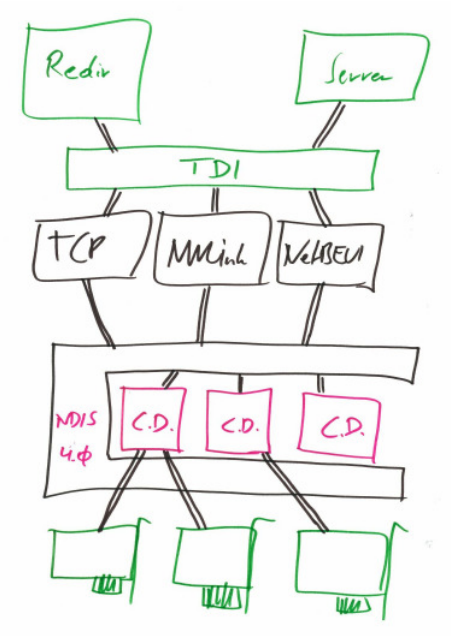
Tabelle W2K Ausprägungen

Architektur des Betriebssystems NT



Skizze Architektur NT4.0 / W2K

Die Architektur des I/O Systems, speziell der Netzwerk-Elemente, sieht folgend aus:



Skizze NT I/O

NDIS

NDIS: Network Driver Interface Specification; in NT v4 kommt NDIS v4 zum Einsatz, in W2K NDIS v5

NDIS ist eine Sezifikation, wie ein Treiber für ein Telekommunikationsgerät aufgebaut sein muß, damit er in das Betriebssystem Windows NT integriert werden kann. NDIS definiert die obere Schnittstelle des Treibers in das Betriebssystem. NDIS v4 behandelt alle Geräte als LANs, geht also davon aus, daß ein Broadcast möglich ist und daß einzelne Datenpakete (Datagramme) versendet werden können. NDIS v4 stellt also keine sehr hohen Anforderungen an das zugrundeliegende System, nutzt aber auch dessen spezifische Funktionen nicht wirklich aus.

Die Zuordnung der Treiber zu den Protokollen nennt man “binden”. Diese Zuordnung ist änderbar. Speziell auf die Reihenfolge, mit der die “unteren” Protokolle an die “oberen” Protokolle gebunden sind, ist manchmal acht zu geben. So ist es beispielsweise günstig, die am meisten verwendeten “unteren” (Schicht 2) Protokolle an die vorderen Stellen der Bindungslisten zu bewegen. Dies ist in NT in den Netzwerk-Eigenschaften möglich.

NDIS-Treiber greifen auf einen gemeinsamten NDIS-Kern zurück, der in der DLL (Dynamic Link Library) NDIS.SYS zusammengefaßt ist. Die einzelnen NDIS-Treiber werden von den Herstellern der NICs mitgeliefert bzw sind in der Installations-CD des NT/W2K bereits enthalten.

Die NDIS-Treiber, die den NIC-spezifischen Teil des Treibers ausmachen (also alles außer der NDIS.SYS), werden auch als Card Driver (“CD” oder im NDIS v5 als "Miniports") bezeichnet. CDs sind nicht nur für LAN-Protokolle wie zB die IEEE 802.x verfügbar, sondern auch für ATM, DecNet, SNA, APPC und auch Point-to-Point-Protokolle wie bz Datex-P (X.25).

Ein NDIS-Treiber pro NIC-Typ ist ausreichend. Es können also mehrere NICs desselben Typs von einem NDIS-Treiber serviciert werden.

Mit Windows 2000 wurde die ursprüngliche Spezifikation erweitert, um verbindungsorientierte untere Schichten (der spezifische Grund war ATM) besser zu nutzen. Dieser NDIS v5 genannte

Standard bietet neben dem verbindungslosen Datagramm-Diensten auch verbindungsorientierte Dienste an. Er wird daher auch als "CONDIS" („Connection Oriented NDIS“) bezeichnet. NDIS v5 ist zu NDIS v4 abwärtskompatibel.

Eine Verbindung im NDIS v5 muß explizit angefordert und abschließend wieder getrennt werden. Dabei können QoS-Charakteristika vereinbart werden, die direkt an die Schicht 2 weitergereicht werden. Ferner dient ein komplexes Multicast-System dazu, die Broadcast-Lastigkeit des NDIS v4 zu korrigieren. Der Vorteil des NDIS v5 liegt im der wesentlich besseren Abbildung des verbindungsorientierten TCP als dem wichtigsten L4 Protokoll auf die verbindungsorientierten Dienste des ATM.

TDI

TDI: Transport Driver Interface

Als Designmechanismus des “alten” NT 4.x wurde die Abstraktion eines TDIs vorgenommen. Das TDI ist sowohl eine Interface-Definition, als auch ein API. Damit ist es für alle “oberen” Schichten des Netzwerks möglich, über eine einheitliche Schnittstelle auf die einzelnen Schicht 4 Protokolle (Transport-Protokolle laut OSI) zuzugreifen.

Damit kann man unter NT beispielsweise die File- und Print-Services über TCP/IP laufen lassen, aber auch über NWLink (Novell Netware IPX/SPX-Protokoll) oder NetBIOS (NetBEUI). Die Abstraktion des TDI ist für diese Anwendungsfälle sehr mächtig, aber mit der Reduktion der Netzwerkprotokolle auf der Transportschicht auf ein einziges - nämlich TCP/IP - ist die Bedeutung des TDI etwas in den Hintergrund getreten. Mit dem Nachfolger von Windows 2000 (derzeitiger Codename "Whistler", wahrscheinlicher Produktname "Windows XP") gibt es überhaupt nur noch ein einziges Protokollsystem im Windows, nämlich eben TCP/IP (Windows XP soll noch 2001 erscheinen).

WinSock

WinSock: Windows Socket Interface

WinSock ist ebenso wie TDI eine Abstraktion für eine Transportschicht, aber in diesem Fall ganz konkret und ausschließlich nur für das TCP/IP Protokollsystem. WinSock baut auf der bereits legendären Socket-Abstraktion des TCP/IP im Unix auf. Bei dieser Methode wurde der “Socket” erfunden. Ein Socket ist eine Kombination aus Protokoll, Quell-Hostname, Quell-Port, Ziel-Hostname und Ziel-Port (zb `TCP, cdemuth2.bmc.com, 10103, hpsrv01.infosys.tuwien.ac.at, 1521`). Wenn ein Socket einmal geöffnet ist, verhält er sich wie ein File-Deskriptor. Man kann also mit konventionellen File-Funktionsaufrufen Daten über den Socket senden bzw Daten von diesem empfangen. Dies ist speziell im Unix-Umfeld eine vorteilhafte Eigenschaft, da man alle Utilities, die mit (bereits vorgeöffneten) Files (sog. Handles) arbeiten können, auch mit Sockets verwenden kann.

WinSock v2 (Windows-Datei "WS2_32.dll") ist die Erweiterung des alten, nun als WinSock v1.1 bezeichneten APIs, sie erfolgte speziell in Richtung der Unterstützung von ATM und anderen MAC-Protokollen. Im WinSock v2 können zB spezielle ATM-Eigenschaften direkt aus der Anwendung heraus angesprochen werden. WinSock v2 bildet also einen Weg für die Anwendung, "durch das TCP/IP hindurch" direkt mit der Schicht 2 zu kommunizieren und dabei dennoch innerhalb eines "genormten" APIs zu bleiben.

Während WinSock v1.1 noch sehr nahe am Berkeley-Standard war, was die Namen und Parameter der 24 Funktionen betrifft, so ist WinSock v2 mit 37 Funktionen (die aber die v1 Funktionen

größtenteils ersetzen) doch um etliches komplexer. Spezielle Erweiterungen sind vorhanden für:

1. ATM als MAC-Protokoll: Direkter Durchgriff auf ATM-Funktionen aus dem Socket Interface heraus (hat mit der ursprünglichen Idee der Socket Abstraktion nichts mehr zu tun)

2. QoS-Vereinbarungen: in der sogenannten "Flow Specification" können die gewünschten QoS-Werte "Latency" (in µsec), "Peak Bandwidth" (in Bytes/sec) und "Delay Variation" (µsec) von der Anwendung eingestellt werden. Praktisch alle QoS-Vereinbarungen werden vom Protokoll "RSVP" ("Resource Reservation Protocoll") übertragen. Das RSVP steht in der Architektur des TCP/IP neben dem TCP und dem UDP auf der Schicht 4.

   Dies entspricht nicht ganz den ToS-Parametern des TCP/IP, da dort nur ein Bit pro QoS vorhanden ist. Ferner erlaubt WinSock v2 auch die Auswahl eines der folgenden Servicetyps:

   "SERVICETYPE_BESTEFFORT" (so gut das System kann),
   "SERVICETYPE_CONTROLLEDLOAD" (unter "normaler Last" werden die QoS-Werte eingehalten),
   "SERVICETYPE_GUARANTEED" (garantierte Einhaltung der QoS-Werte) und
   "SERVICETYPE_NETWORKCONTROL" (maximale QoS-Priorität, nur für interne Protokolle wie RSVP verwendbar) eingestellt werden.

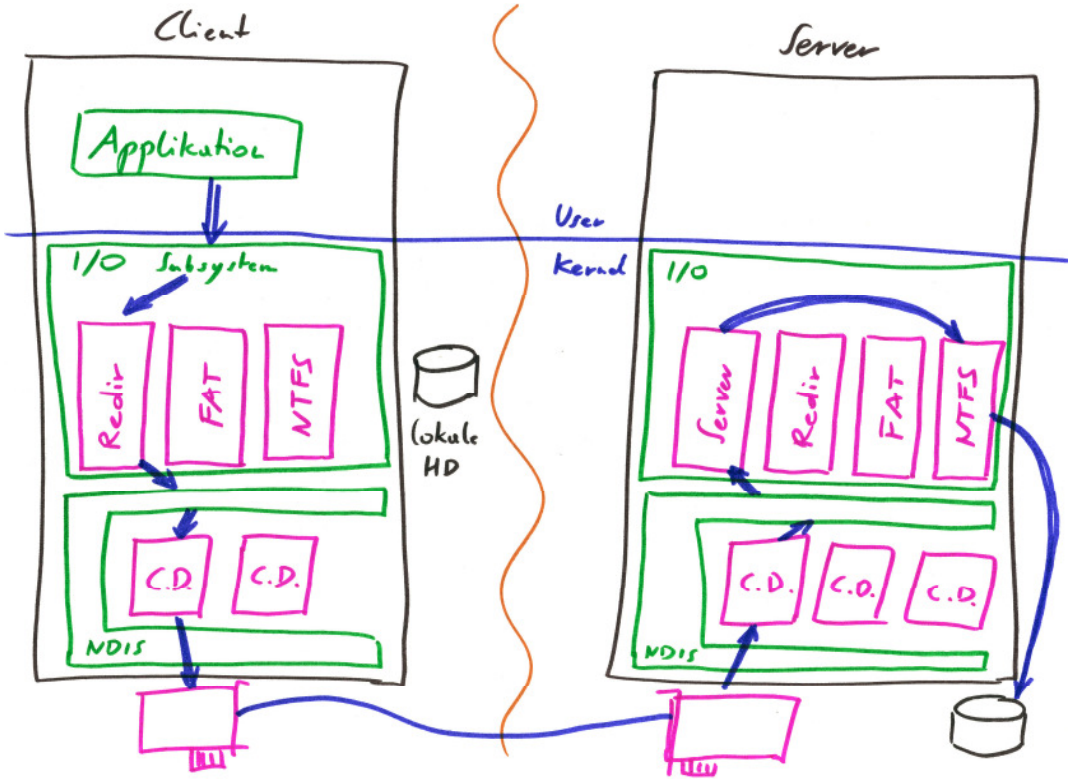
3. Scatter/Gather IO: hierbei werden die Benutzerdaten für ein Datenpaket (UDP) oder den Datenstrom (TCP) nicht aus einem zusammenhängenden Stück Speicher geholt bzw in ein zusammengehöriges Stück abgelegt, sondern der Funktion wird ein Array mit Pointern auf einzelne Speicherbereiche samt deren Längen übergeben. Die Funktionen sorgen dann dafür, daß die Daten für die Datenpakete aus den einzelnen Speicherbereichen zusammengesetzt ("gather") oder in einzelne Speicherbereiche aufgeteilt ("scatter") werden.

4. Direkte Manipulation von Multicast-Funktionalität untenliegender Protokolle (zB LAN-MACs, ATM, etc).

5. Asynchrone Versionen vieler WinSock v1.1 Funktionen.

Redirector / Server

Der “Redirector Service” (“Redirector-Dienst”) des NT stellt den Client-Teil des File- und Print-Sharing dar. Der Redirector leitet Zugriffe auf Shares, also auf entfernte (remote) Resourcen, an den zugehörigen Rechner und dort an den zugehörigen Dienst namens “Server Servcice” (“Server-Dienst”) weiter. Dieser führt dann die Aufgabe lokal durch und gibt das Ergebnis wieder an den Redirector zurück.



Skizze Redirector und Server Service

Network Services

Der sogenannte “Computer Browser” ist eine der angenehmen Dinge des NT. Man kann mittels des NT-Exploreres in der “Netzwerkumgebung” durch die einzelnen Computer der Domains und Arbeitsgruppen blättern. Als Alternative kann man sich den „Uniform Naming Convention Name“ („UNC-Name“) der Resource merken. Ein UNC sieht folgend aus:

\\computername\share für nicht-hierarchische Resourcen wie zB Netzwerk-Drucker

\\computername\share\subdir\subdir\… für hierarchische Resourcen wie zB Netzwerk-Filesysteme

Anstatt des Computernamens kann auch der DNS-Namen des Computers verwendet werden bzw die IP-Adresse (wenn TCP/IP als Transport-Protokoll verwendet wird).

Der Nachteil des Computer Browsings ist die relativ hohe Netzwerklast, die entsteht, da sich die Computer untereinander oft Updates zuschicken. Der PDC ist der Master Browser im Falle daß Domains verwendet werden. Ansonsten wird der Master Browser durch ein Auswahlverfahren festgelegt. Einige Computer werden zu Backup-Browsern. Ein Rechner, der den Browser-Service aktiviert hat, kann einerseits andere Computer per Browser finden, andererseits aber auch zum Browser Server ernannt werden und damit eine Tabelle der Computernamen aufbauen.

ATM im NT / W2K

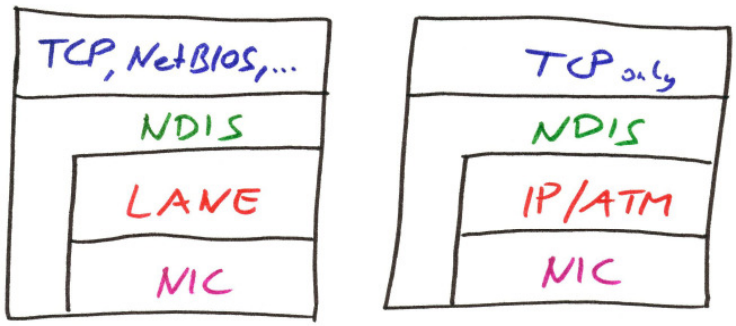
Die Einbindung des ATM in ein „von unten weg“ auf LANs aufgebautes System ist nicht wirklich einfach. ATM als verbindungsorientiertes und broadcastloses Protokoll paßt nicht so recht in das NDIS-Konzept, das sich mit dem Versenden von einzelnen Datenpaketen (Datagrammen) befaßt.

Es muß daher entweder das NDIS an das ATM angepaßt werden oder das ATM an das NDIS. Der

zweite Ansatz ist der einfachere (zumindest für das Betriebssystem). Es wird das ATM so "umgebaut", daß es ein LAN emuliert. Daher der Name LANE ("LAN Emulation"). Die speziellen Fähigkeiten des ATM (zB die möglichen "Quality of Service" Vereinbarungen) werden deaktiviert. Die für ATM notwendigen Verbindungen werden durch ein komplexes System an Software vor dem NDIS "versteckt". Broadcasts werden durch ein komplexes System an Multicasts emuliert. ATM verhält sich nun fast wie ein LAN und kann mittels NDIS genauso in ein NT System integriert werden wie Ethernet, Tokem Ring oder FDDI.

Alternativ kann man auch anstelle eines allgemeinen NDIS-Treibers einen speziell auf ein einziges darüberliegendes Protokoll "zugeschnittenen" Treiber verwenden. Dieser heißt IP/ATM. Bei IP/ATM wird als einziges Transportprotokoll TCP bzw UDP zugelassen. Dafür kann aber besser die QoS des TCP bzw UDP auf die QoS des ATM abgebildet werden.

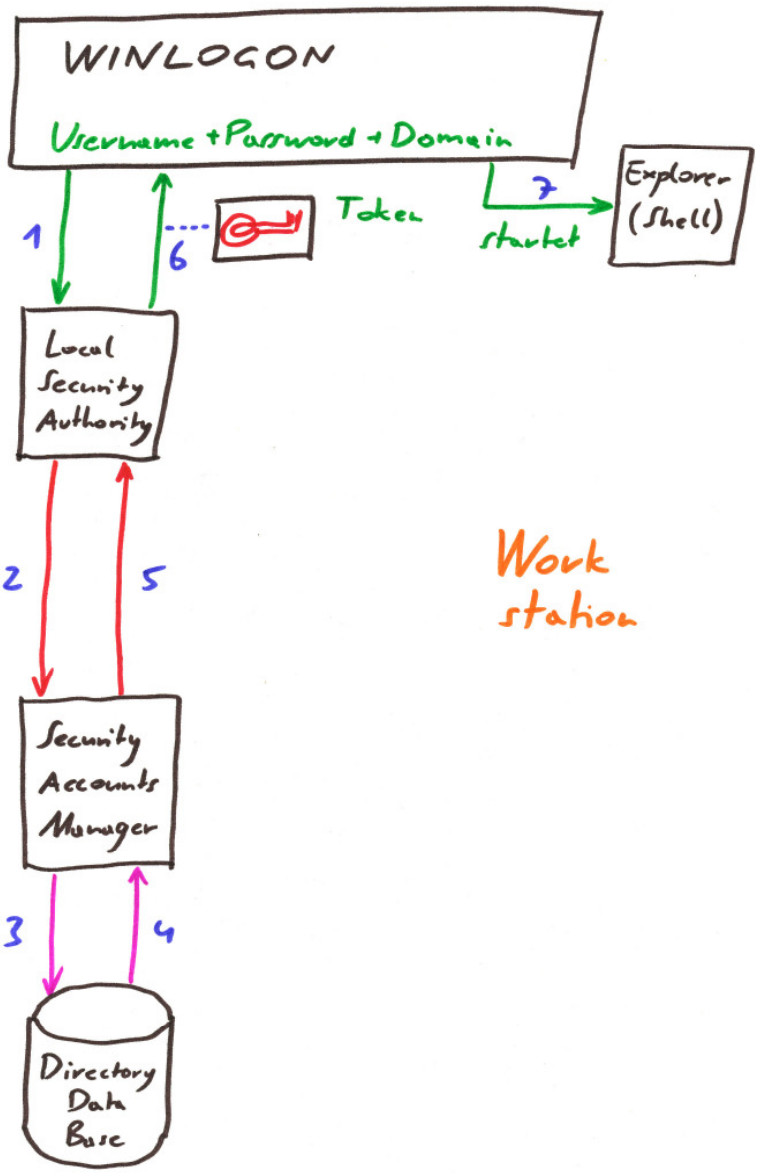
Die erstgenannte Variante bedeutet, daß man ATM-Funktionalität verliert. Man kann aber auch das NDIS erweitern. Dies geschah mit W2K, hier wurde NDIS v5 erstmals verwendet. NDIS v5 kennt einen erweiterten Modus, in dem ATM und ATM-ähnliche Systeme direkt unterstützt werden. NDIS v5 kennt neben den aus v4 bekannten verbindungslosen Treibern nun auch verbindungsorientierte Treiber. Mit dieser Art von CDs ist es einfach, ATM in W2K zu integrieren und man kann zusätzlich die volle Leistungsfähigkeit von ATM, die teilweise weit über das bei LANs übliche hinausgeht, nutzen. Zusätzlich wurde WinSock v2 in W2K integriert. Damit ist es nun auch auf dem Socket-Interface möglich, die speziellen Features des ATM mittels Socket-Interface anzusteuern.



Skizze Architektur von LANE und IP/ATM

NT v4 Security

NT v4 verwendet als Berechtigungskonzept Login und Passwort. Aus diesen beiden Werten wird ein Security-Token generiert, das anschließend für die Authentifizierung verwendet wird. Man kann eine lokale Security-Datenbank verwenden oder eine zentralisierte. Im ersten Fall erfolgt der Login gegen die lokale Datenbank:

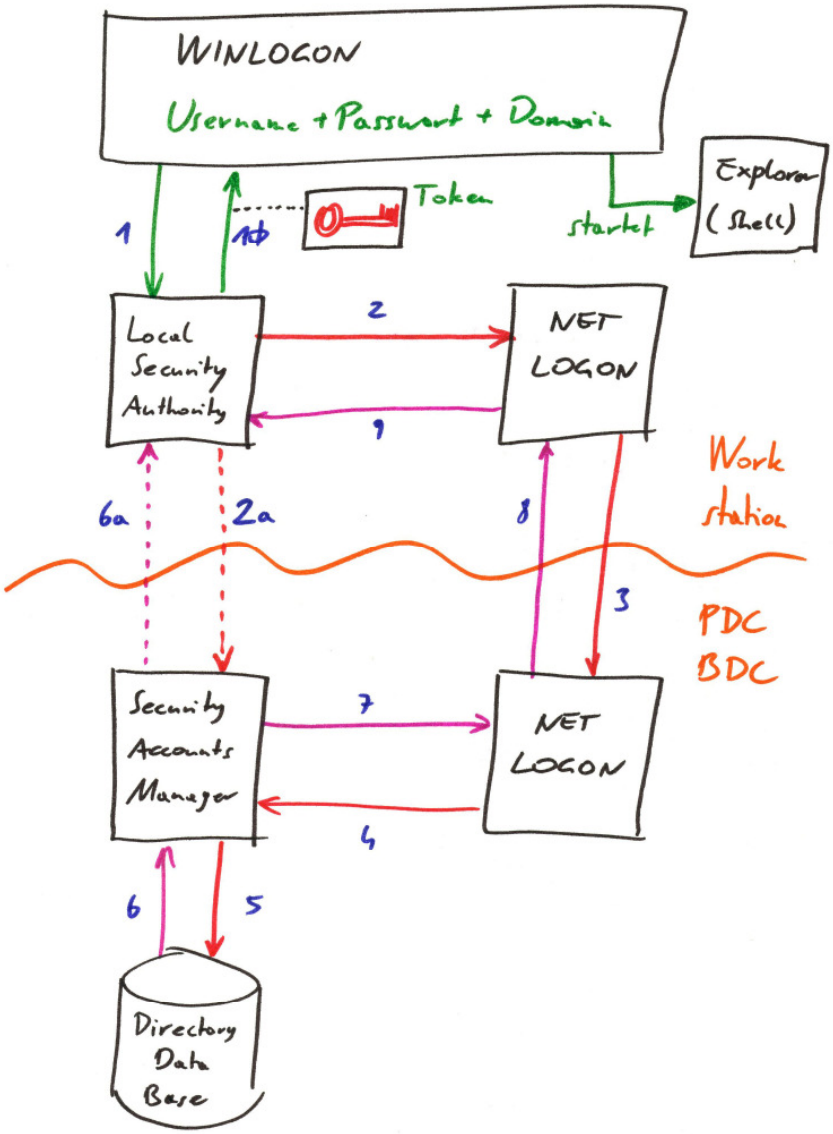


Skizze Lokaler NT Login

Man verwendet den lokalen Rechnernamen (unter NT hat jeder Computer einen lokalen Namen, dieser wird auch als Hostname des Computers im DNS verwendet) als sogenannten “Domain-Namen”. Damit wird eine Authentifizierung gegen die lokale Security-Datenbank durchgeführt.

Domain

Die sogenannte “Domain” (Deutsch: Domäne) ist eine Sammlung von Resourcen und Benutzern und Gruppen samt zugehörigen Berechtigungen. Dabei erhält jeder Benutzer bzw jede Gruppe für eine Resource eine bestimmte Berechtigung, zB diese Resource lesend zu verwenden. Die Sammlung dieser Berechtigungen kann zentralisiert in einer einzigen Datenbank erfolgen. In disem Fall spricht man von einer Domain. Die einzelnen Teilnehmer der Domain (die sogenannten “Workstations”) authentifizieren ihre Benutzer gegen diese zentralisierte Datenbank.



Skizze Domain-Login

Damit ist die Verwaltung der Benutzer, Gruppen und Resourcen zentral auf einem Rechner möglich. Dieser Rechner ist der sogenannte “Primary Domain Controller” (“PDC”). Ihm zur Seite stehen ein oder mehrere “Backup Domain Controller” (“BDC”), die Redundanz und Lastverteilung bewirken. Die BDCs sind Eins-zu-Eins Replikate des PDC. Schreibzugriffe werden auf dem PDC durchgeführt und sofort repliziert. Lesezugriffe (das ist die überwiegene Mehrzahl der Zugriffe) werden allgemein nur von den BDCs beantwortet (sofern welche existieren).

Der Login in eine Domain ermöglicht die zentralisierte Vergabe von Rechten und auch die Ausführung von Scripts beim Login (zB für das automatische Mappen von Shares, das Einbinden von Netzwerkdruckern, das Installieren von Softeware, etc), die so geschützt werden können, daß sie von der Workstation nicht mehr beeinflußt werden können. Es ist damit eine weitgehende Steuerung der Workstations von der Domain aus möglich.

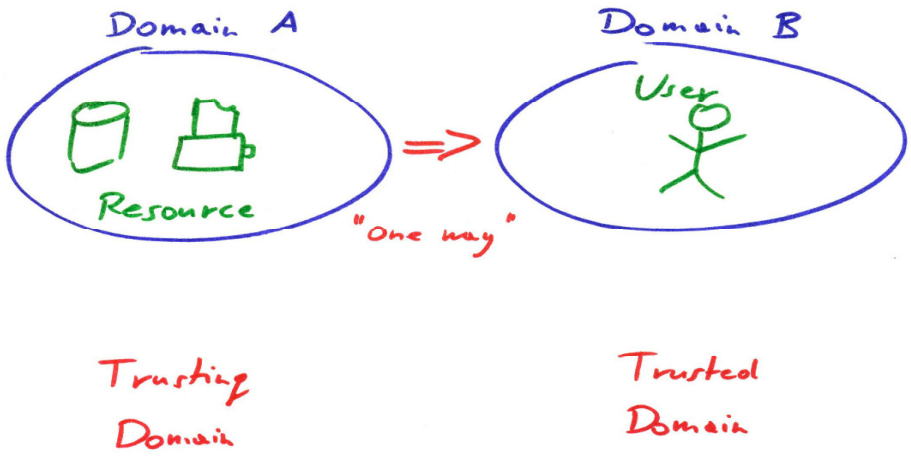
Workgroups

Ist ein NT-Rechner in keiner Domain eingeloggt, sondern wird lokal betrieben (der Domainname ist ident mit dem Hostnamen), kann man noch weiters “Workgroups” (“Arbeitsgruppen”) bilden. Diese werden nicht zentralisiert verwaltet und bilden damit nur eine lose Gruppe von Workstations. Workgroups und Domains schließen sich gegenseitig aus, eine Workstation kann zu einem Zeitpunkt entweder in einer Domain oder in einer Workgroup eingeloggt sein. Der eigentlich einzige Vorteil von Workgroups ist, daß sich Workstations, die in derselben Workgroup befinden, im Computer Browser als “zusammengehörig” angezeigt werden. Die Zugehörigkeit zu einer Workstation zu einer Arbeitsgruppe hat sicherheitsmäßig keinerlei Konsequenzen. Jede Station kann sich jederzeit in einer Workgroup anmelden und wieder daraus entfernen.

Domain-Trusts

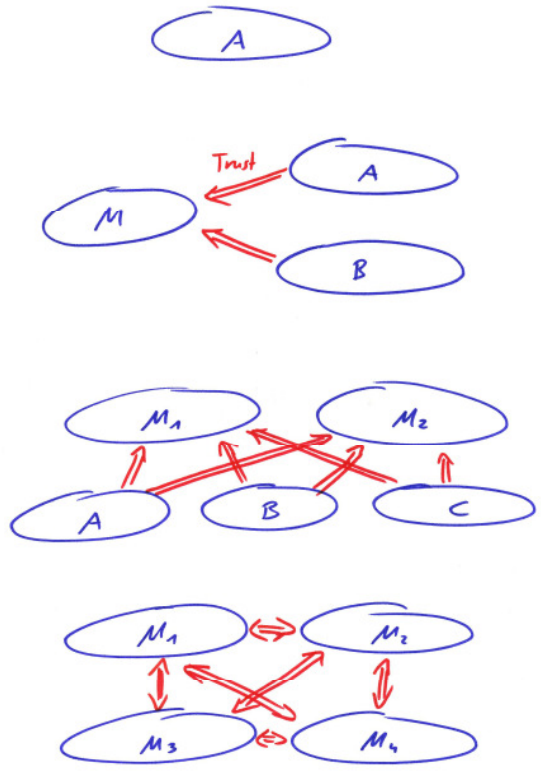
Die Domain unter Windows NT 4.x ist eine “flache” – also nicht hierarchische – Sammlung von Benutzern, Gruppen und Rechten. Alle Namen in dieser Domain sind gleichwertig. Die Anzahl der Namen, die in einer Domain verwaltbar sind, ist auf einige zig-Tausend beschränkt. Ferner wird die Verwaltung dieses Namensraums bei einigen Tausend Namen bereits unangenehm, da man immer mit Drop-Down-Listen zu tun hat, die mit tausenden Werten befüllt sind. Und daraus etwas auszuwählen wird allein schon vom Handling her mühsam.

Daher gibt es bei NT 4.x die Möglichkeit, mehrere Domains zubilden und diese miteinander zu verbinden. Diese sogenannten “Domain-Trusts” sind unidirektional, also nur in eine Richtung gültig, und außerdem nicht transitiv (also wenn A dem B vertraut und B dem C, dann ist nicht automatisch das Vertrauen von A nach C gegeben, was eigentlich logisch wäre). Dieses Verhalten bewirkt, daß bei mehr als einigen wenigen Domains die wechselseitigen Trusts kompliziert zu administrieren werden (Complete Trust).



Skizze Domain vertraut ("trusts") einer anderen Domain

Es wird daher oft die Aufteilung in sogenannte Resource-Domains und User-Domains durchgeführt. Dabei werden in der ersten Domain-Art nur Resourcen (Shares, Drucker, sonstige Geräte) verwaltet und in der zweiten Art nur Benutzer und Gruppen. Diese Aufteilung vermindert den Administrationsaufwand.



Skizze Master Domain / Multi-Master-Domain / Complete Trust

W2K Security: Active Directory

Die bei weitem wichtigste Neuerung des W2K gegenüber dem NT v4 ist sicherlich das „Active Directory“ („AD”) des W2K. AD ist der Nachfolger des Domain-Systems im NT 4.x und damit das neue Security-System. Die wesentlichen Schwachpunkte des Domain-Systems wurden behoben und dabei auch gleich im Gegenzug Erweiterungen eingeführt, die das W2K für das nächste “Windows-Jahrzehnt” vorbereiten.

AD behebt den “flachen” Namensraum (eigentlich nur eine Liste von Namen) des alten Domain-Konzepts und ersetzt ihn durch einen hierarchischen Namens-Baum (“Directory Information Tree”, kurz “DIT”). Dieser ist konzeptionell an LDAP (RFC-1777 “Lightweight Directory Access Protocol” und RFC-2251 “Lightweight Directory Access Protocol v3”, eine vereinfachte Form des X.500-Systems) angelehnt. Auch das Netzwerk-Protokoll basiert in wesentlichen Teilen auf LDAP. Ferner ist auch das DNS ein integraler Bestandteil des AD. AD ist also – so wie das alte Domain-Security-System – sowohl eine Datenbank als auch ein Protokoll. Im Gegensatz zum alten Domain-System ist die Datenbank des AD nun aber erweiterbar (auch in ihrer Struktur, dem sogenannten Schema), replizierbar und eben hierarchisch aufgebaut. Ferner wurde die Obergrenze der verwaltbaren Objekte auf einige Millionen angehoben, während bei der NT v4 Domain bei einigen zig-tausend Objekten pro Domain Schluß war. AD kennt keinen Unterschied zwischen PDC und BDS mehr, alle beteiligten AD-Server sind PDCs.

User und Gruppen (zB Rechte, Profiles, Policies)
Client-Daten (zB Mgmt Profile, Netzwerk-Infos, Policy)

Server-Daten (zB Services, Drucker, Shares, Policies)
Netzwerk-Daten (zB QoS-Policy, Security, Konfigurationsdaten)
Anwendungs-Daten (Anwendungs-Konfigurationsdaten)
E-Mail-Daten (zB Mailbox-Infos, Addressbücher)
Andere Directory-Daten (zB von externen Directory-Diensten)

Tabelle Was wird alles im AD gespeichert?

Das zugrundeliegende Protokoll LDAP ist ein stark auf den lesenden Zugriff ausgelegtes System (90% lesen und 10% schreiben). Es ist nicht transaktionsorientiert. Im Gegensatz zum allumfassenden X.500 (“Directory Services”) ist man hier von der bekannten „90/10 Regel“ ausgegangen und hat diejenigen 90% der Features implementiert, die einfach zu realisieren waren und insgesamt 10% der Kosten ausmachen und den Rest dann einfach weggelassen. Es wurden also keine sogenannten “low use functions” implementiert und das gesamte Protokoll auf Strings und andere einfache Datenstrukturen und ein einfaches Encoding (unter Encoding versteht man die Umsatzung der Datenstrukturen im Rechner in Datenpakete auf dem Netz) aufgebaut. LDAP verwendet den Port 389 für normale Abfragen und den Port 636 bei Verwendung des "Secure Socket Layers" ("SSL"). Es wird sowohl TCP als auch UDP unterstützt.

Ferner können die Daten des AD repliziert werden. Im Gegensatz zum NT 4.x muß dafür aber keine Online-Verbindung zwischen den Domain-Controllern bestehen, sondern eine Replikation per SMTP (Mail) ist ebenfalls möglich. Die dabei entstehenden Kollisionen werden behandelt.

Trusts zwischen einzelnen ADs sind bidirektional und transitiv.

Dem LDAP liegen 4 Modelle zugrunde:

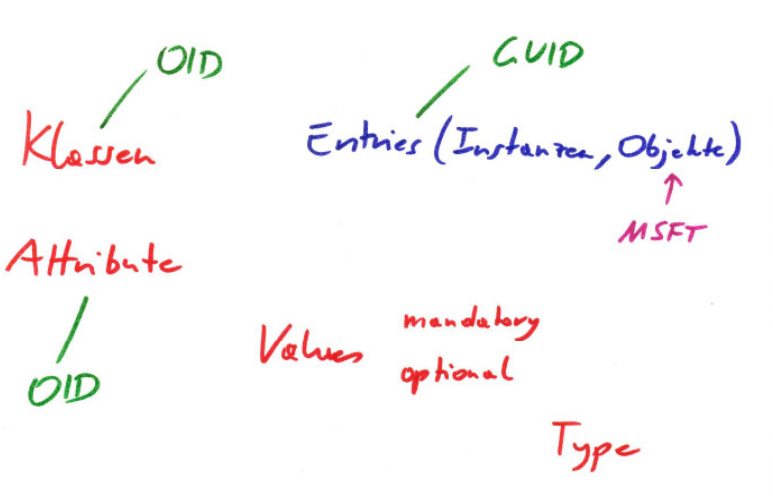
Information (Struktur)
Naming (Namensgebung)
Funktion (Zugriff)
Security (Sicherheitsaspekte)

Tabelle LDAP Modelle

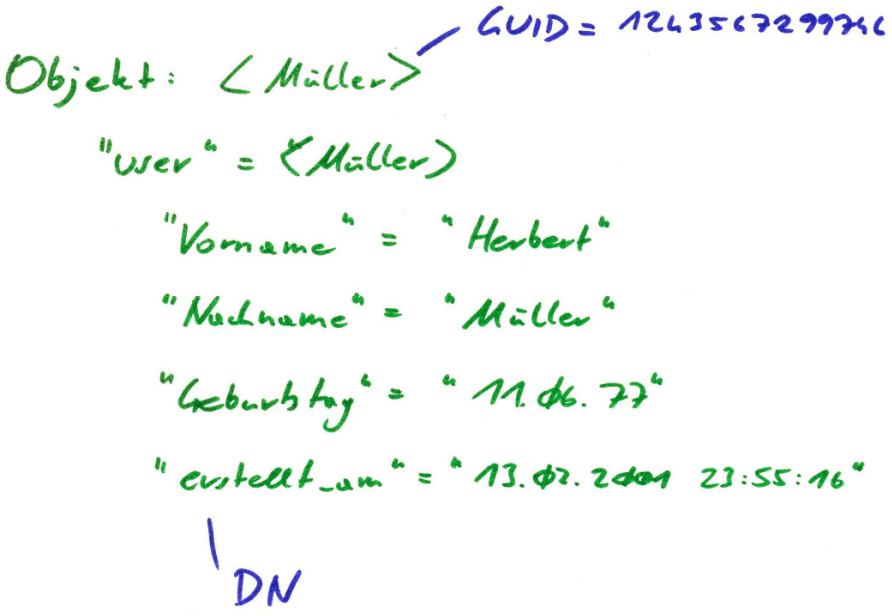
Information

Nach streng objektorientierter Sitte werden Klassen mit Attributen gebildet. Dann werden Instanzen (im LDAP-Standard „Entries“ oder bei W2K „Objects“ genannt). Objects haben Global-Unique-Ids (GUIDs), das sind 128 Bit lange eindeutige Schlüssel. Klassen und deren Attribute haben Object-Ids (OIDs), die - genauso wie die Klassen und Attribute hierachisch angeordnet sind - auch hierarchischen Aufbau haben. Attribute haben einen Typ und einen Value, ferner gibt es “mandatory” und “optional” Attribute. Außerdem können Attribute mehrere Werte (gleichen Typs) haben.

Die OID für den Attribut-Type "Zeit" ("UTC time") lautet 1.3.6.1.4.1.1466.115.121.1.53 und die für einen "Common Name" lautet 2.5.4.3. Der Zugriff auf Objekte oder Attribute erfolgt aber in der Regel nicht nach den OIDs, sondern nach deren textueller Form, also nach den Texten "UTC" bzw "CN".



Skizze Klasse – Attribute – Entries – Values - Types
links die „statischen" Informationen: Klasse, Attribute der Klasse und deren Typen – Klassen, Attribute und Typen haben OIDs
rechts die „dynamischen" Informationen: Objekte (Instanzen der Klasse) mit ihren Values (Instanzen der Attribute) – Objekte haben GUIDs



Skizze Objekt mit Attributen und Werten
ein Objekt „Müller" mit seiner GUID
links die Attribute (statisch), rechts die Values (dynamisch)
die Typen der Attribute sind unterschiedlich (Vorname = String, Geburtstag = Datum, erstellt_am = Timestamp)

Naming

Um nun ein Objekt im AD ansprechen zu können, muß man entweder dessen GUID kennen, oder man kann es „benennen". Dafür gibt es die "Distinguished Names" ("DN"). Sie werden unterschieden in relative DNs ("RDNs") und absolute DNs (kurz nur "DNs" genannt). RDNs sind nur in einem vorgegebenen Kontext des DIT (der AD Datenbank) sinnvoll. Ein RDN muß relativ zu etwas Absolutem sein, in diesem Fall ist es ein Zweig des Baumes, von dem aus gestartet wird. RDNs haben ihre Berechtigung, da sie kürzer sind als volle DNs und in sehr vielen Fällen

ausreichen (ausreichend eindeutig sind). Die Attribut-Namen und Typen von DNs sind teilweise vorgegeben, zB bei den folgenden:

cn=cdemuth	cn = “Common Name"
dc=bmc	dc = “Domain Component"
ou=sales	ou = “Organizational Unit"

Absolute DNs (RFC 1779) bestimmen ein Objekt im DIT des AD “von der Wurzel weg”. Sie sind also eine Kette von RDNs, getrennt durch “,” (oder auch “;”) die im “root” des AD beginnen. Die Objekte des AD sind im DIT nach ihrer Lage im Baum angeordnet, wobei die Wurzel am Ende steht - im Gegensatz zur Schreibweise von zB Directories.

```
cn=cdemuth,ou=sales,dc=bmc,dc=com
```

Die OUs sind aus mehreren Gründen wichtig. Einerseits ermöglichen sie eine Strukturierung der Objekte in einer Art „Verzeichnisstruktur“, andererseits dienen sie als Träger für die Berechtigungsstruktur. Mithilfe von OUs kann ein Systemadministrator spezifische Teilrechte an „Subadministratoren“ delegieren. OUs sind beliebig schachtelbar. Am Ende einer OU-Kette liegen die CNs. Vor der OU-Kette liegen die DCs. Die Kette der DCs beschreibt ein Active Directory mithilfe seines DNS-Namens. Ein AD hat also einen DNS-Namen, zB „sales.bmc.com“. Der Namen des AD ist auch zugleich der DNS-Namen des „ersten“ DCs dieses ADs.

Die Objekte eines AD (samt OUs etc) bilden eine sogenannte Domain.

```
bmc.com   (Domain)
          ou1
                    ou1.1
                    ou1.2
                              ou1.2.1
                                        cn=cdemuth
                                        cn=müller
                              ou1.2.2
                              …
                    …
          ou2
          …
```

Mehrere Domains können hierarchisch angeordnet werden (Subdomains, genau wie im DNS). OUs, CNs etc können mehrfach vorkommen und durchaus lokal gleich heißen, sie werden durch die unterschiedlichen DCs (unterschiedliche Domains) auseinandergehalten.

```
bmc.com   (Domain)
          sales.bmc.com     (Sub-Domain, dc=sales)
                    ou1
                              ou1.1
                                        cn=cdemuth
                              ou1.2
                              …
                    ou2
                    …
          tech.bmc.com      (Sub-Domain, dc=tech)
                    ou1
                              ou1.1
                                        cn=cdemuth
                              ou1.2
                              …
                    ou2
                    …
          ou1
                    ou1.1
                    ou1.2
                              ou1.2.1
                                        cn=cdemuth
                                        cn=müller
```

ou1.2.2
…
…
ou2
…

Die hierarchische Gruppierung mehrerer Domains nach obigem Muster nennt man einen „Tree“. „bmc.com“ wäre also ein Tree mit drei Domains („bmc.com“, Subdomains „sales.bmc.com“ und „tech.bmc.com“).

Zusätzlich ist es möglich, mehrere Trees, die parallel nebeneinander existieren, zu einem sogenannten „Forest“ zusammenzubinden.

Die logische Struktur der AD-Objekte lautet also

Objekte
OUs
Domains
Trees
Forests

Sites

Parallel zur „logischen“ Anordnung von Objekten innerhalb von AD-Systemen gibt es eine „physische“ Anordnung. Da man sich für die Replikation zwischen den einzelnen Domain Controllern nach Eigenheiten wie zB Leitungsgeschwindigkeit etc richten muß, werden ADs in sogenannte „Sites“ unterteilt. Eine Site ist eine Replikations-Einheit mit zumindest einem Domain Controller. Sind mehrere DCs in der Site, so werden diese in einen logischen bidirektionalen Ring aufgenommen und replizieren die Datenänderungen kreisförmig weiter.

Zwischen Sites werden „Site Links“ explizit definiert. Diese Links sind vorgegebene Schnittstellen zwischen einzelnen Sites. Es werden mittels „Least Cost Routing“ die geänderten Daten zwischen Sites bidirektional ausgetauscht.

Schreibweisen

Ferner wird eine “kanonische” Form definiert. Diese gibt eine alternative, allgemeingültige Schreibweise an. Im Falle des AD ist diese Schreibweise der URL des WWW sehr ähnlich. Es werden die Informationen teilweise umgedreht (alle RDNs außer den “dc”), teilweise in ihrer Reihenfolge gelassen (“dc"-RDNs). Die RFS822 (Mail)-Schreibweise lautet:

cdemuth@sales.bmc.com

Auch URL-Schreibweisen sind möglich, zB HTTP:

http://sales.bmc.com/cdemuth

oder auch als UNC:

\\sales.bmc.com\cdemuth

oder als LDAP URL:

ldap://sales.bmc.com/cn=cdemuth

Funktion

Die wesentlichen Funktionen im AD sind das Abfragen von Informationen, der Vorgang der Authentifizierung und das Updaten von Informationen. Das AD ist auf die Abfrage von

Informationen hin optimiert. Die Abfrage muß nicht an der Wurzel des DIT starten, man kann auch in einem Teilast beginnen (“scoped search”). Dazu ist ein “base object” zu definieren (das ist ein DN des DIT), an dem die Suche beginnt, zB suche nach “cn=cdemuth” beginnend bei “dc=bmc,dc=com” (das ist das “base object”). Das Suchergebnis sind alle Attribute desjenigen DN, der dem Suchkriterium genügt.

Bei der Suche kann man entweder nur die durch den Start-DN (das “base object”) aktuelle Ebene im DIT absuchen, oder nur die direkt unter dem “base object” liegende Ebene des DIT oder alle darunterliegenden Ebenen. Dies wird durch den sogenannten “scope” eingestellt.

Gesucht werden kann nach:

exaktem Treffer	(cn="cdemuth”)
Substring	(cn="cde*”)
Pattern Matching	(cn=~"cde”)
Größer	(cn >= “cdemuth”)
Kleiner	(cn <= “cdemuth”)
Existenz	(cn =*)

Zusätzlich ist die Boolsche Verknüpfung ("and" &, "or" | und "not" !) von Suchkriterien möglich. Das Modifizieren von Einträgen des DIT erfolgt ebenfalls über LDAP. Dabei kann man Einträge hinzufügen (“add”, benötigt den DN und alle Attribute), Löschen (“delete”, benötigt den DN), Attribute ändern (“modify”, benötigt DN und die zu modifizierenden Attribute plus ggf deren Werte beim Hinzufügen) oder den RDN ändern (“modify RDN”, benötigt den DN, neuen RDN und ein Flag, ob der alte RDN aufgehoben wird).

Security

Es wird Kerberos (RFC 1510 "The Kerberos Network Authentication Service v5") für die Verschlüsselung der LDAP-Datenströme verwendet. Dieser bisher nur in der Unix-Welt weitverbreitete Standard ermöglicht Single-Sign-On.

Während des Login-Vorgangs mittels Benutzername, Passwort und Domainname wird vom Kerberos “Key Distribution Center” (“KDC”) ein sogenanntes “Ticket” (exakter ein “Kerberos Session Key”) erstellt. Dieses Ticket wird immer dann "vorgezeigt", wenn eine Authentifizierung vorgenommen werden soll.

Kerberos ist allerdings nicht das einzige unterstützte Security System des W2K. Auch das alte "NT Lan Manager" Security System (stammt ursprünglich von der IBM aus den späten 80er Jahren) ist noch unterstützt genauso wie das im MSN und im CompuServe verwendete "DPA" ("Distributed Password Authentication").

Driver Certification

Im Gegensatz zum NT 4.x, bei dem die von den Herstellern der diversen Geräte beigestellten Treiber von Microsoft nicht auf Funktionsfähigkeit geprüft wurden, gibt es bei W2K ein Treiber-Zertifizierungs-Programm. Dabei werden die Treiber nach bestimmten Kriterien untersucht und treten “gegeneinander” in einer Arena an, um sie auf schädliche Wechselwirkungen zu testen. Wie wohl bekannt ist, kann man durch Testen nicht die Fehlerfreiheit eines Programmes beweisen, man kann nur mehr Vertrauen in das Programm gewinnen. Dennoch ist es gegen die bei Treiberfehlern im NT 4.x und auch im W2K unvermeidlichen Blue Screens of Death (“BSoD”) eine Hilfe.

DDNS

Dynamic DNS

Hinter diesem Begriff versteckt sich ein einfacher Mechanismus. Ein DHCP-Server vergibt einem Client beim Booten eine IP-Adresse und einen Namen und fügt dieses Paar sofort auch dem DNS-Server hinzu. Damit hat der Client immer sofort gültige IP-Adressen und DNS-Namen. Dieser scheinbar harmlos klingende Mechanismus war die Ursache vieler kleiner Probleme im NT. DDNS gibt es nur bei W2K.

TCP / IP

Überblick

TCP / IP (Transport Control Protocol / Internet Protocol) ist ein "Protocol Stack", also ein ganzer Satz von zusammengehörigen Protokollen auf verschiedenen OSI-Schichten. TCP/IP ist das einheitliche Protokoll-System, das dem Internet und all seinen Diensten zugrundeliegt. Seine geschichtliche Entwicklung beginnt bereits in der Frühzeit der EDV.

Die IETF ("Internet Engineering Task Force") leitet derzeit die Normung der Internet-Technologien.

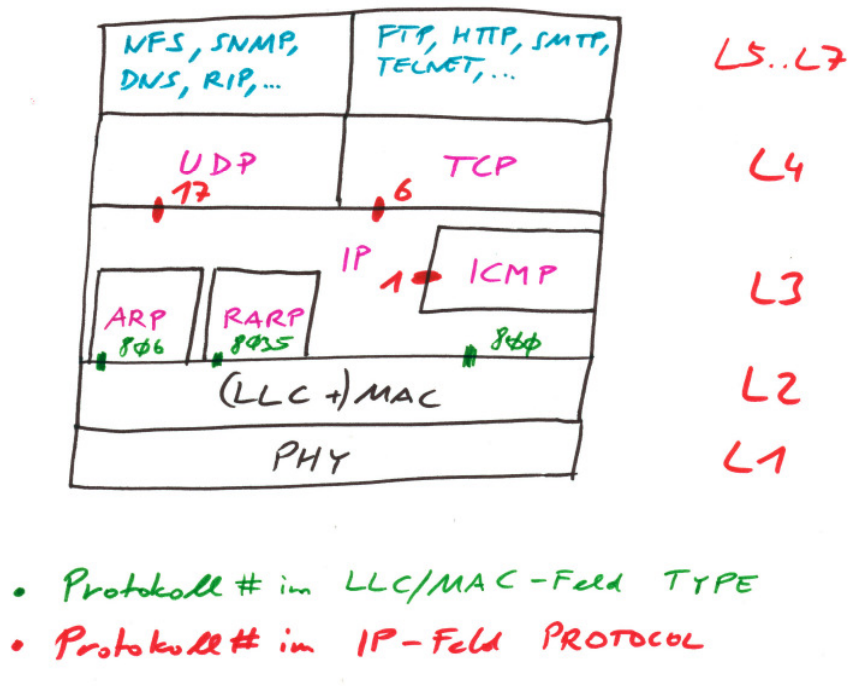
1957	Der Satellit Sputnik wird von der UDSSR ins All befördert und löst in den USA den sogenannten "Sputnik-Schock" aus. Die USA gründen daraufhin die "ARPA" ("Advanced Research Project Agency").
1962	Erste Ideen zur Paketschaltung reifen.
1969	Das Amerikanische Departmanet of Defense ("DoD") gründet eine eigene "Netzwerk-Abteilung" in der ARPA, diese beginnt mit dem Aufbau des ARPANET. Initial waren vier Super-Computer in den USA im ARPANET zusammengebunden.
1971	15 Knoten sind im ARPANET verbunden, als Protokoll wird das "Network Control Protocol" ("NCP") verwendet.
1973	EMail kommt zum ersten Mal zum Einsatz. Ursprünglich wurde EMail für die Kommunikation der Leute, die mit dem Netz zusammenarbeiteten, konzipiert. Robert Kahn und Vint Cerf (heute die "Väter des Internet" genannt) beginnen mit dem Umbau von NCP auf einen neuen, offenen Standard, der später TCP/IP genannt werden wird.
1974	Ethernet wird erfunden. Ein Forschungsprogramm, „Internetting Project" genannt, wird ins Leben gerufen und das Netzwerksystem, das sich daraus entwickelte wurde später als das „Internet" bekannt. Das Protokollsystem, das zugrunde liegt, ist TCP/IP.
1976	"Unix to Unix Copy" ("UUCP") ist das Transportmedium in den internationalen Netzen. Es ist ein Weitergabedienst für Dateien und Mails, kann aber auf Remote Rechnern auch Batch-Programme ausführen.
1982	TCP/IP wird im ARPANET zum Standard, nachdem es ab 1980 Standard-Protokoll im militärischen Bereich wurde. In Europa wird das EUnet gegründet, das Europäische Unix Netzwerk. Damit kommt EMail auch nach Europa.
1984	245 Knoten im ARPANET. Umstieg im ARPANET von NCP auf TCP/IP. Aufgrund der ständig wachsenden Domains Einführung von DNS. Damit wurde die Verteilung der Last auf mehrere Rechenr möglich (hatte bislang ein einziger Rechner übernommen).

1985	Symbolics.com wurde die erste registrierte Domain und wurde am 15. März zugewiesen.
1986	Start des NSFNET ("National Science Foundation Network") – Beginn der Entmilitarisierung des ARPANET. Die NSFNET-Infrastruktur ist auch heute noch ein wichtiger Teil des Internet-Backbones. Das Exponentielles Wachstum des Internets beginnt.
1988	Der berühmte Internet-Wurm greift um sich und bringt das Netz in weiten Teilen zum Stillstand. Entwicklung des Internet Relay Chat (IRC). Der damalige US-Senator Al Gore liest den NSF-Bericht "Towards a National Research Network" und beginnt mit dem Aufbau von Hochgeschwindigkeitsnetzwerken, welche den Grundstein für einen späteren Daten-Superhighway legten.
1990	ARPANET wird vollständig durch das NSFNET ersetzt, das Internet in seiner heutigen kommerziellen Form beginnt. Es wird nach wie vor nur von den Universitäten und Forschungseinrichtungen und einigen Enthusiasten verwendet. Ca 100.000 Knoten sind im Netz. Österreich wird an das NSFNET angeschlossen. Tim Berners-Lee - er gilt allgemein als Erfinder des "World Wide Web" oder kurz "Web" – schreibt den ersten Web-Browser. Dieser hat mit den heutigen allerdings nur die Funktionsprinzipien gemeinsam.
1991	Gopher, der Vorläufer des WWW, wird erfunden. Ca 600.000 Knoten im Netz. NSFNET gibt das Internet, das bisher nur für Universitäten und Forschungseinrichtungen gedacht war, für den kommerziellen Bereich "frei".
1992	Jean Armour Polly prägte den Begriff „Surfen im Internet".
1993	Das InterNIC wird gegründet und übernimmt die administrativen Aufgaben im Internet zentralisiert. "Mosaic X", der erste Web Browser, wird von Marc Andreesen (am National Center for Supercomputing Applications) erfunden. Mosaic kann Grafiken direkt in die Texte einbetten. Andreesen ist später Mitbegründer von Netscape, das die nicht-kommerziellen Mosaic-Systeme erweitert und auf viele Plattformen portiert. Ende 1993 gibt es ca 600 WebSites.
1994	Eigentlicher Beginn des WWW wie man es heute kennt (HTTP/HTML). Ca 3 Mio Knoten im Netz. Netscape ist als eine der ersten Softwareprodukte im Internet gratis downloadbar.
1995	Das Federal Networking Council (FNC) beschließt, dass der folgende Text unserer Definition des Wortes "Internet" entspricht. „Internet" bezieht sich auf das globale Informationssystem, das – (I.) logisch durch einen weltweit einzigartigen Adressraum basierend auf das Internet Protokoll (IP) oder dessen folgenden Erweiterungen/Follow-Ons vernetzt ist; (II.) es ermöglicht Datenübertragungen durch die Verwendung des Transmission Control Protocol/Internet Protocol (TCP/IP) oder dessen folgenden Erweiterungen/Follow-Ons und/oder anderer IP-kompatibler Protokolle zu unterstützen; und (III.) entweder für öffentliche oder private Zwecke auf hoher Ebene Dienste basierend auf hier beschriebener Datenübertragung und diesbezüglicher Infrastruktur zur Verfügung stellt, verwendet oder zugänglich macht. Wichtige eigene Netzbetreiber steigen als "ISPs" in das Internet ein: AOL, CompuServe, Prodigy.
1996	Mehr als 100.000 WebSites online.
1997	Mehr als 1 Mio. WebSites online. Mehr als 100 Mio Internet-Benutzer.
1998	Mehr als 3,7 Mio WebSites online. Mehr als 150 Mio Internet-Benutzer.
1999	Mehr als 10 Mio. WebSites online. 250-300 Mio Internet-Benutzer – die Zahl ist nicht mehr exakt feststellbar.
1999	Dezentralisierung der Domainname-Vergabe
Heute	Kommerzielles WWW. Das WWW ist das Aushängeschild des Internet. Aber das eMail ist die eigentliche "Killer-Application", die jedes kommerziell erfolgreiche System benötigt.

	Einige statistische Daten: Muttersprachen der Internet-Benutzer: Englisch 51%, Deutsch 7%, Japanisch 7%, dann folgen Spanisch, Chinesisch und Französisch. Sprachen in den WebSites: 87% Englisch Geschätzte 1 Billion(!) Dokumente im Internet verfügbar. 18 Mio registrierte Domains (DNS), 9.5 Mio davon sind ".com"s. Die USA haben die größte Anzahl an registrierten Domains, Deutschland ist auf Platz drei mit über einer Mio, Österreich unter den "Top Ten" mit 100.000. Das Internet Software Consortium nimmt zweimal im Jahr eine Zählung der Host-Computer vor, und im Januar 2000 schätzte man 72,4 Millionen in der DNS eingetragene Hosts. Messaging Online schätzt, dass es Ende 1999 569 Millionen EMail-Accounts gab, um 83% mehr als im Jahr zuvor. Jedoch benutzen nur 5 % der Beschäftigten und 6 % der Haushalte EMail. Ca 200.000 aktuelle aktive NewsGroups.

Tabelle Entwicklung von TCP/IP

Im TCP/IP Protokoll Stack sind heute hunderte Protokolle definiert. Für einen kurzen Einblick empfiehlt sich die Datei `/etc/services` eines Unix-Systems. Sie listet die wichtigsten (die sogenannten "well known") TCP/IP Schicht 7 Protokolle im System auf. Die wichtigsten Protokolle des Stacks sind folgend aufgebaut:



Skizze TCP/IP Architektiur ("Stack")

Der Kern des TCP/IP Systems ist ein Datagramm-Dienst, das IP ("Internet Protocol"). Dies läßt sich historisch damit erklären, daß so ein Dienst relativ einfach zu realisieren ist und daher eine Menge an Stabilität und Fehlertoleranz gewinnt. Ein IP-Datagramm kann verloren gehen, es kann dupliziert werden und einzelne Datagramme können in einer anderen Reihenfolge eintreffen, als sie abgesendet wurden. All dies ist zulässig und daher werden keine hohen Anforderungen an das IP gestellt. Damit erklärt sich auch die Historie des IP als Protokoll, das aus dem militär-nahen Bereich kommt.

Andererseits wurde es von Grund auf als routendes Netzwerk gebaut. Routing ist einer der wesentlichsten Teile des IP und ohne dem Routing ist IP fast wertlos. Das ist jetzt übertrieben, da auch zB das Fragmentieren Teil der IP-Aufgabe ist und die Fehler-Meldungen über das ICMP Teil des IP sind. Dennoch ist IP genaugenommen eine “geroutete Verlängerung” des Datagramm-Dienstes von Ethernet in die Schicht 3 hinein.

Wir besprechen erstmal die aktuelle Version des IP, IPv4. Auf das neue IPv6 gehen wir erst später explizit ein.

IP-Adressen

Internet-Adressen - oder korrekter IPv4-Adressen - sind 32 Bit Zahlen, die eindeutig im Netz sein müssen. Als Schreibweise wird die Form `A.B.C.D` verwendet, wobei jeder Buchstabe für ein Byte in dezimaler Schreibweise steht.

Jeder Knoten in einem TCP/IP Netzwerk muß eine eindeutige IP-Adresse haben.

IP-Adressen bestehen aus einem Netz-Anteil - auch Netz-Adresse genannt - und einem Host-Anteil (im TCP/IP werden die Knoten als Hosts bezeichnet) - auch Host-Adresse genannt. Alle Knoten eines Netzes, die dieselbe Netz-Adresse haben, können sich direkt – also ohne daß ein Router verwendet werden muß oder kann (!) – miteinander verständigen. In einem LAN ist ein mittels Repeatern oder Bridges zusammengefügtes Netz nach wie vor ein einzelnes Netz aus IP-Sicht.

TCP/IP ist aber nicht nur auf LANs als Netze beschränkt, es können auch zB Punkt-zu-Punkt-Verindungen über TCP/IP geführt werden. In diesem Fall besteht das Netz aus genau zwei Knoten. Damit ist auch klar, daß man für Punkt-zu-Punkt-Verbindungen besser eine IP-Adresse mit einem kleinen Host-Anteil verwendet und für ein großes LAN, das man als zusammengehöriges einzelnes Netz einbinden will, einen IP-Adressbereich benötigt, der viele verschiedene Hosts zuläßt. Die IP-Adressen werden daher in 5 Klassen unterteilt:

Klasse	**erste Bits der IP Adresse**	**Anzahl der Netzbits**	**Anzahl der Hostbits**	**Erste IP-Netz-Adresse** **Letzte IP-Netz-Adresse**
A	0	7	24	001 126
B	10	14	16	128.001 191.254
C	110	21	8	192.000.001 223.255.254
D	1110	---	---	224.000.000.000 239.255.255.254
E	1111	---	---	240.000.000.000 255.255.255.254

Tabelle IP Adressklassen

Die Klassen A bis C bilden “normale” IP-Adressen. Lauter binäre Einsen im Host-Teil werden als “Broadcast in diesem Subnetz” interpretiert. Lauter binäre Nullen werden als “dieses Subnetz” interpretiert. Eine Quell-Adresse der Form 0.0.0.0 wird verwendet, wenn ein Gerät seine eigene IP-

Adresse nicht kennt. Die Ziel-Adresse 255.255.255.255 dient als allgemeine Broadcast Adresse, wird aber im IP-Netz nur im eigenen Subnetz als Broadcast versendet. Ein alter Standard erlaubt auch die Verwendung von 0.0.0.0 als Broadcasts-Adresse.

Netz-Adresse	Host-Adresse	Als Quelle	Als Ziel	Bedeutung
0er	0er	OK	Nie	Dieser Host auf diesem (dem lokalen) Netz. Wird bei der Initialisierung von Maschinen verwendet, die die eigene IP-Adresse noch nicht kennen. Tritt auch bei einigen Statistik-Tools (zB “netstat”) auf.
0er	Host-ID	OK	Nie	Ein bestimmter Host auf diesem (dem lokalen) Netz. Ähnlich wie oben.
127	Egal	OK	OK	Loopback IP-Adresse.
1er	1er	Nie	OK	Nicht-gerouteter (auf das lokale Netz beschränkter) Broadcast
Netz-ID	1er	Nie	OK	Weitergerouteter Broadcast (geht über das eigene Netz hinaus)

Tabelle Spezielle IP-Adressen

Die Klasse D beinhaltet die IP-Multicast-Adressen und die Klasse E die “Spezialadressen für Sonderzwecke". Beide Klassen sind vorreserviert und stehen nicht für individuelle IP-Adressen der Knoten zur Verfügung.

Da es in der Klasse A nur 126 Subnetze (2 ^ 7 - 2) gibt, waren diese Adressen auch bald vergeben. In jedem Klasse-A-Netz kann man zwar 2 hoch 24 minus 2 Knoten betreiben, aber so viele Knoten hat wohl kaum einer der Besitzer einer Klasse-A-Adresse je in Betrieb genommen. Auch die Klasse-B-Adressen sind bereits vergeben, heute gibt es nur noch Klasse-C-Adressen, da es von diesen 2 ^ 21 - 2 verschiedene Netze gibt. Damit ist aber die Administration dieser Adressen wieder komplex geworden, da größere Unternehmen gleich -zig verschiedener Klasse-C-Netze erwerben müssen. Und um zwei Netze miteinander zu verbinden muß man zwingend einen Router dazwischenschalten, was die Sache nicht nur komplizierter, sondern auch noch uneffizienter macht. Als Workaround wurden daher die sogenannten Subnetze eingeführt (siehe weiter unten).

IP

IP ist das Kernprotokoll im TCP/IP System. Es wird derzeit hauptsächlich (fast ausschließlich) in der Version 4 verwendet.

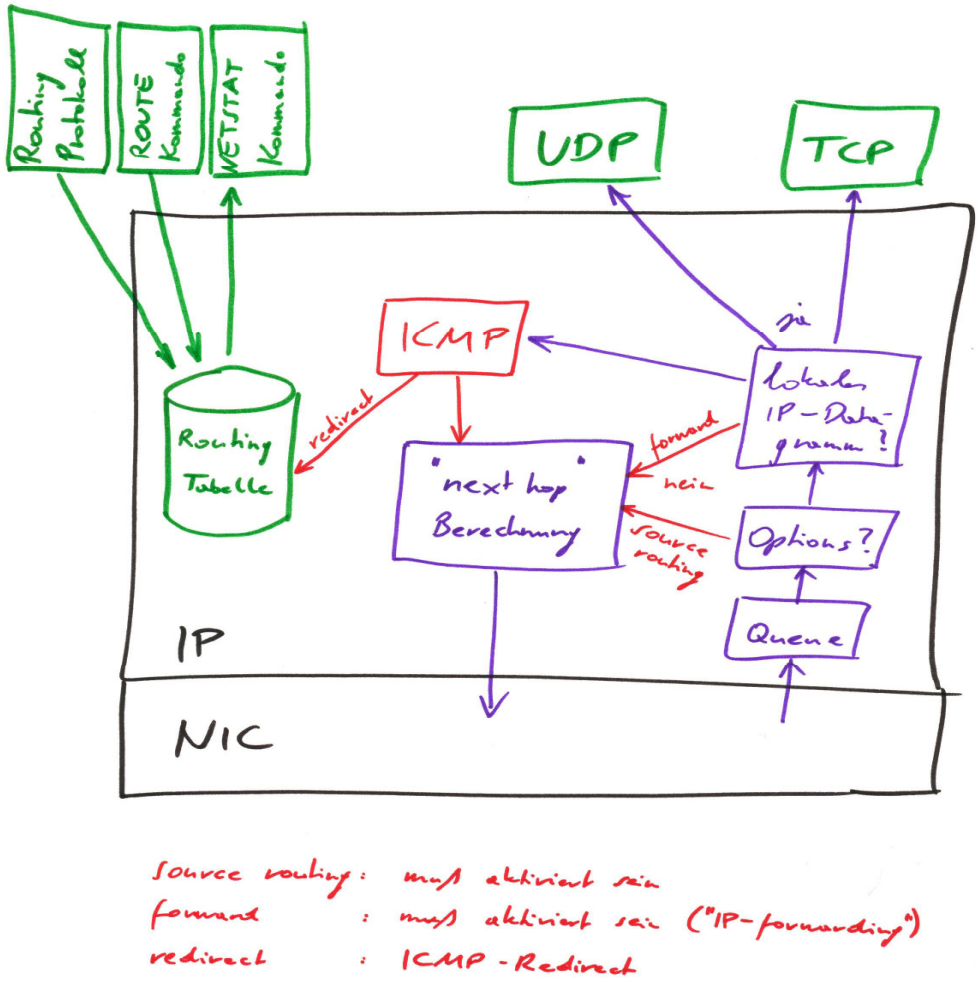
Eine der wichtigsten Aufgaben des Schicht 3 Protokolles IP ist das Routing. Andere Aufgaben sind die Verwaltung der “Quality of Service” Parameter (im IP “Type of Service”, “TOS” genannt), das Fragmentieren und das Adressieren.

Beim Senden eines Paketes wird die Ziel-IP-Adresse mit der Subnetzmaske ge-Und-et und das Resultat mit der eigenen Netz-Adresse (ebenfalls mit der Subnetzmaske ge-Und-et) verglichen. Ist das Ergebnis in beiden Fällen gleich, so befindet sich der Zielknoten im selben Netz wie der sendende Knoten und das Paket kann direkt gesendet werden. Wenn ein LAN das zugrundeliegende Netz bildet, wird per Hilfsprotokoll ARP die MAC-Adresse des Zielknotens festgestellt und das Paket direkt an diese LAN-Adresse gesendet.

Ansonsten wird das Paket direkt an das konfigurierte Default Gateway gesendet, welches das weitere Routing übernimmt.

Innerhalb des IP-Protokolls sind weitere Sub-Protokolle eingebettet: ARP und RARP als Schnittstellenprotokoll zur Schicht 2 und ICMP als Steuerungsprotokoll.

IP ist ein Datagramm-Protokoll, es werden also ausschließlich Datagramme gesendet und im IP werden nie Verbindungen aufgebaut. Das Protokoll wird daher auch als “zustandslos” bezeichnet. Damit ist gemeint, daß es sich in keinem bestimmten Protokoll-Zustand befinden kann, da es nur einen einzigen Zustand (“Paket senden” auf der Senderseite bzw “Paket empfangen” auf der Empfängerseite) kennt. Diese Einfachheit ist zugleich auch der Schlüssel zum Erfolg des IP.



Skizze innerer Aufbau des IP-Layers

IP-Datagramme dürfen auf ihrem Weg zum Ziel verloren gehen, dupliziert werden und auch in der zeitlichen Reihenfolge des Eintreffens vertauscht werden. Dies war ein Designziel des ursprünglichen IP und hat sich bis heute bewährt, da es sehr geringe Ansprüche an das darunterliegende System der Schichten 2 und 1 stellt. Allerdings harmoniert es nicht so gut mit modernen, komplexen Schicht 2 Systemen wie zB dem ATM.

Das IP-Datagramm beinhaltet die folgenden Informationen:

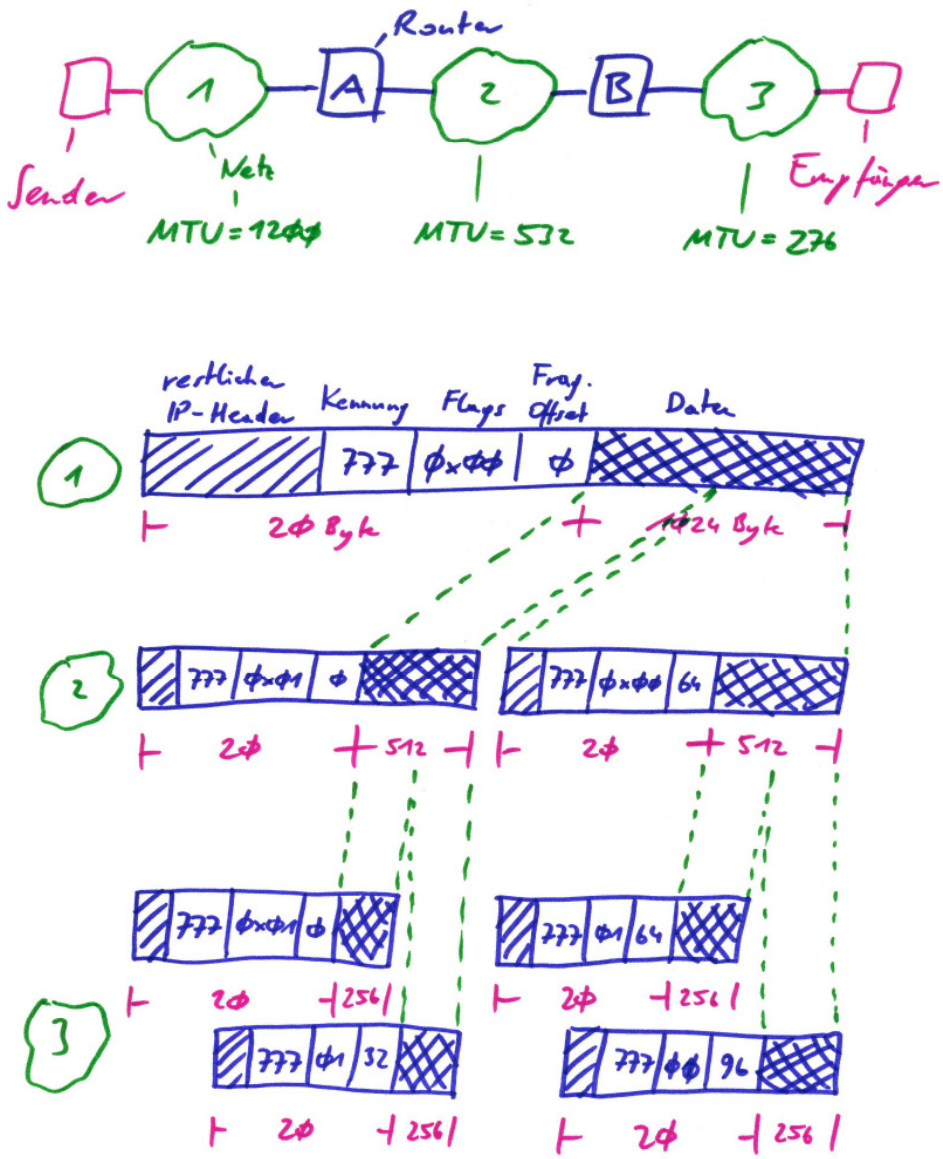
Feld	**Bits**	**Bedeutung**
Version	4	Versionsnummer des IP Protokolls, derzeit “4”

IHL - Inter Header Length	4	Länge des IP Headers in 32-Bit-Einheiten. Daraus folgt, daß ein IP-Header immer ein Vielfaches von 32 Bit sein muß
TOS – Type of Service	8 (4)	Von diesen 8 Bits werden nur 4 verwendet, nämlich (D)elay, (T)hroughput, (R)eliability (zusammen "DTR" genannt) und (C)ost. Das TOS Feld fordert vom IP-Transportsystem eine Behandlung des Datagrammes nach den Kriterien Delay (kürzestmögliche Verzögerumg = raschestmögliche Zustellung), Throughput (bester Durchsatz für hohe Datenvolumen), Reliability (Zuverlässigkeit = Datagramm sollte möglichst nicht vernichtet werden) oder Cost (geringste Kosten der Übertragung, gemessen in den jeweiligen Routing-Metriken). Es darf immer nur ein einziges TOS-Bit gesetzt sein! Moderne Routing-Protokolle wie OSPF machen von den TOS-Feldern Gebrauch bei der Routenbestimmung.
Länge	16	Gesamtlänge des IP-Datagrammes in Bytes
Kennung	16	Kennung der einzelnen Fragmente des Datenpakets der Schicht 4, das in mehrere Datenpakete im IP zerlegt werden mußte. Alle Fragmente eines Datenpakets haben dieselbe Kennung im IP-Header, selbst wenn von einem anderen Router das Fragment nochmals fragmentiert werden muß.
Flags	3 (2)	Bit "do not fragment" (DF) und Bit "More fragments" (MF). Drittes Bit nicht verwendet. DF sagt, daß das Datagramm unfragmentiert weitergesendet werden muß. Falls das nicht geht, wird per ICMP eine Fehlermeldung an den ursprünglichen Sender retourniert und das Datagramm wird verworfen. MF zeigt das Ende (das letzte Fragment) eines fragmentierten Datenpaketes der Schicht 4 an.
Fragment-Offset	13	Startposition in 8-Bytes des in diesem IP-Datagramms transportierten Fragments im Datenpaket der Schicht 4. Ein Fragment-Offset von 10 bedeutet daher, daß dieses Paket Daten trägt, die im endgültigen reassemblierten Datenpaket an der Position 8*10 stehen würden.
TTL – Time to Live	8	Maximale Anzahl der Router auf dem Weg zum Ziel. Jeder Router dekrementiert TTL um eins. Ist TTL gleich Null in einem Router, verwirft dieser das Datagramm. Alternativ wird TTL auch als Anzahl von Sekunden "Lebenszeit" für ein IP-Datagramm interpretiert. Nach TTL Sekunden wird es verworfen (im Router) und ein ICMP-Paket (Type 11, "time exceeded") an den Sender geschickt.
Protokoll	8	Welches Schicht 4 Protokoll sendet Daten in diesem Datagramm? Ein Byte: UDP=17, TCP=6, ICMP=1, EGP=8, OSPF=89, …
Kopf-Prüfsumme	16	Sehr einfach errechnete Prüfsumme über den IP-Header (bitinvertierte Summe der bitinvertierten Bytes im Header).
SA	32	Quell-IP-Adresse
DA	32	Ziel-IP-Adresse
Optionen	Var	Optionale Felder im IP-Header. Werden zB vom ICMP verwendet.
Daten	Var	Variable Länge an Dateninhalten der Schicht 4. Maximale Größe aufgrund des Feldes "Länge" im IP-Header 2 hoch 16 Bytes, in fragmentierten Datagrammen aufgrund des Feldes Fragment-Offset auch kleiner.

Tabelle IP-Header. Der fixe Header (also ohne Optionen) ist exakt 20 Bytes lang.

Die "Maximum Transferrable Unit" ("MTU") bezeichnet die maximale Größe eines IP-Datagrammes.

Alle Fragmente eines Datenpakets der Schicht 4 werden erst vom Zielknoten wieder zusammengebaut (reassembliert). Jedes Fragment kann seinerseits wieder in einem Router fragmentiert werden, wenn die MTU des empfangenden Subnetzes kleiner ist als die MTU des sendenden Subnetzes. In allen Fragmenten wird dieselbe Kennung im Header gesetzt, aber das MF-Bit wird nur im letzten Fragment gesetzt. Auch die Fragment-Offsets werden jeweils angepaßt.



Skizze Fragmentierung
Vorgegeben ist ein Internet mit 3 Netzen (1, 2 und 3), durch 2 Router (A und B) verbunden. Der Sender will ein Datagramm mit 1024 Byte Nutzdaten und 20 Byte IP-Header übertragen. Die MTU des Netzes 1 beträgt 1200, das Paket wird daher als ein einziges Datagramm der Länge 1044 versendet.

Im Netz 2, das nur eine MTU von 532 Byte aufweist, wird das Datagramm in zwei Datagramme fragmentiert. Jedes Datagramm trägt 512 Bytes der Benutzerdaten und den IP-Header von 20 Byte, macht zusammen 512+20=532 Byte. Jedes der beiden Datagramme trägt dieselbe Kennung, das zweite und derzeit letzte Datagramm hat im Flag das Bit "MF" ("More Flag") gelöscht und im Fragment-Offset den Wert 512/8=64 eingetragen.

Im Netz 3, das nur noch über eine MTU von 276 Bytes verfügt, werden beide Datagramme nochmals in je zwei Datagramme fragmentiert. Beim Empfänger kommen also 4 Datagramme an. Alle haben die Kennung 777, das letzte hat das MF-Bit gelöscht, die anderen drei haben das MF-Bit gesetzt. Jedes Datagramm zeigt den Fragment-Offset der in seinem Feld Daten transportierten Benutzerdaten an (0, 32, 64, 96 – entsprechend den Byte-Offsets 0, 256, 512, 768). Der Empfänger reassemliert das ursprüngliche Datagramm Nummer 777 aus den vier Teildatagrammen und reicht es an die Schicht 4 weiter. Kommen die Teildatagramme nicht zeitgerecht beim Empfänger an, wird das Datagramm verworfen.

IP Subnetze

Da es teilweise notwendig ist, ein IP-Netz in mehrere Teilnetze zu zerteilen, muß eine Möglichkeit geschaffen werden, die Adressbereiche aufzutrennen und Router einzufügen. Normalerweise müßte dafür eine zweite, neue Netzadresse verwendet werden und alle Knoten, die im neuen Subnetz liegen sollten, müssten neue IP-Adressen (nämlich die des zweiten Netzes) erhalten.

Da es aber oft notwendig ist, Netze zu zerteilen und den zugeteilten IP-Adressraum bestmöglich auszunützen, sollte eine Möglichkeit existieren, dies einfach zu bewerkstelligen. Diese Möglichkeit nennt sich “Subnetzmaske” und wurde schon sehr bald Teil des TCP/IP Standards. Die Subnetzmaske ist eine Folge von 32 Bits, die mit den Bits der IP-Adresse ge-Und-et wird (logisches UND). Das resultierende Bitmuster ist die eigentliche IP-Netz-Adresse und muß immer länger sein als die ursprüngliche Netzadresse.

Jede der drei Adressklassen hat eine Default-Subnetzmaske, die exakt so lange ist wie die Netzadresse:

Klasse	Default-Subnetzmaske in Bits	in Dezimalnotation
A	11111111.00000000.00000000.00000000	255.0.0.0
B	11111111.11111111.00000000.00000000	255.255.0.0
C	11111111.11111111.11111111.00000000	255.255.255.0

Tabelle Default Subnetze

Wenn man nun ein Klasse-C-IP-Netz in zwei IP-Netze unterteilen will, setzt man die Subnetzmaske wie folgt:

```
Altes IP-Netz:         192     .168     .137     .xxx
Binär:                 11000000.10101000.10001001.xxxxxxxx
Default Subnetzmaske:  11111111.11111111.11111111.00000000
Neue Subnetzmaske:     11111111.11111111.11111111.10000000
```

Damit entstehen 2 IP-Netze, eines mit Netz-Nummern

```
von 192.168.137.1:     11000000.10101000.10001001.00000001
bis 192.168.137.127:   11000000.10101000.10001001.01111110
```

und eines mit Netz-Nummern

```
von 192.168.137.128    11000000.10101000.10001001.10000001
bis 192.168.137.254    11000000.10101000.10001001.11111110
```

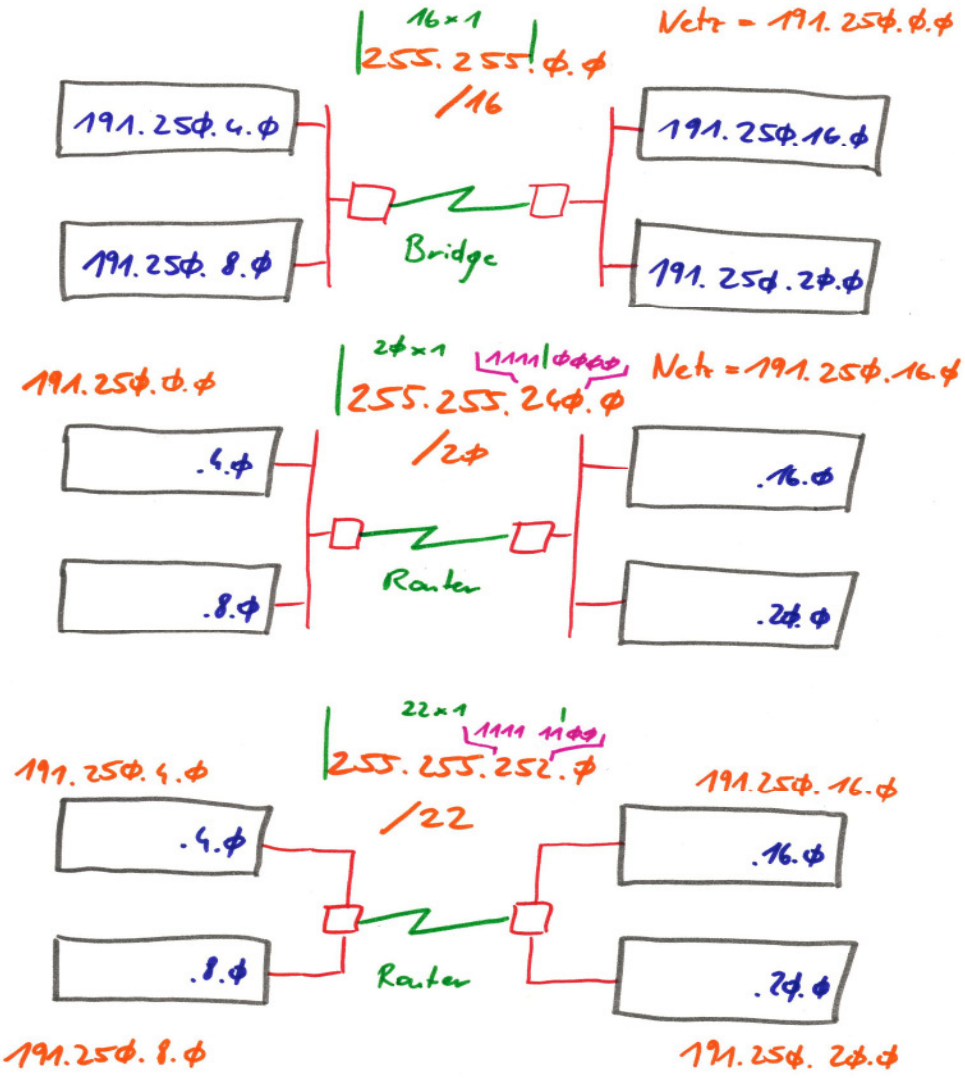
Die Subnetzmaske wird immer in Dezimalnotation angegeben, daher muß man sie noch umrechnen in 192.168.137.128.

Da eine Subnetzmaske immer mit “Einsen” beginnen sollte und mit “Nullen” enden sollte, kann man sie auch mit einer einzigen Zahl beschreiben. Es werden die Anzahl der “Einsen” gezählt und in der Schreibweise “/x” hinter die IP-Adresse bzw den Netzanteil der IP-Adresse gestellt. Die neue Subnetzmaske aus dem obigen Beispiel

```
11111111.11111111.11111111.10000000
```

kann man also auch schreiben als “/25”.

Im untenstehenden komplizierteren Beispiel wird ein einziges gebridgtes IP-Netzwerk in zwei und dann in vier Subnetze mithilfe der Subnetzmaske unterteilt. Die “richtige” Wahl der Netz-Adresse von Anfang an (wie im Beispiel) erleichtert die Aufgabe wesentlich (ist aber in der Praxis unwahrscheinlich).



Skizze IP-Subnetze
oben: ein Netz mit Netz-Id 191.250.0.0/16
mitte: zwei Netze (Subnetze) mit Netz-Ids 191.250.0.0/20 links und 191.250.16.0/20 rechts
unten: vier Netze (Subnetze) mit den Netz-Ids
191.250.4.0/22, 191.250.8.0/22, 191.250.16.0/22 und 191.250.20.0/22

Die Router im TCP/IP-Netz werden aus historischen Gründen auch heute noch als Gateways bezeichnet, obwohl das aus OSI-Sicht falsch ist. Aber TCP/IP war eben vor OSI da und hat seinerzeit den Namen geprägt.

IP Multicast

Wir kennen bereits die Begriffe "Unicast", "Multicast" und "Broadcast". (Mit IPv6 kommt noch der "Anycast" hinzu.)

Der Grund für die in der letzten Zeit stark steigende Bedeutung des Multicasts liegt darin, daß ein Broadcast-Paket in jeder einzelnen Station vom IP-System empfangen und bearbeitet werden muß. Dies kostet CPU-Zyklen in jeder Empfangsstation und damit Last auf allen Systemen.

Schicht	Protokoll	Grund für das Verwefen des Pakets (zB)

NIC	PHY	CRC Fehler?
Card Driver / Device Driver	MAC	Schicht 3 Protokoll unterstützt? Station adressiert?
IP	IP	Station addressiert? IP-Prüfsumme ok?
UDP	UDP	Port addressiert? UDP-Prüfsumme ok?

Tabelle Paket-Filterung - Gründe, warum eine Schicht ein Datenpaket verwirft

Dieses Dilemma kann man mit Multicasts verbessern, da hier die Selektion, ob ein Paket für die Station relevant ist oder nicht, bereits von der MAC-Schicht durchgeführt wird. Da diese zumeist stark hardwareunterstützt arbeitet, wird die Haupt-CPU der Station entlastet. Die Voraussetzung hierfür ist die Abbildung von IP-Multicast-Adressen auf MAC-Multicast-Adressen (das sind MAC-Adressen, bei denen das “Individual Address / Group Address” Bit auf Eins gesetzt ist, siehe unter MAC-Adressen).

Das Multicasten des IP kann nur von Anwendungen verwendet werden, die UDP verwenden oder direkt IP als Transportprotokoll verwenden (dies ist recht selten und meist nur von Routing-Protokollen praktiziert). Mit TCP macht ein Multicast keinen Sinn.

Beim IP-Multicasting wird der IP-Adressraum `224.0.0.0` bis `239.255.255.255` (die gesamte IP-Netzklasse D) verwendet. Es stehen also 3x8 Bits für insgesamt 2 ^ 24 verschiedene Multicast-Adressen zur Verfügung. Eine Station, die als Empfänger an einem Multicast teilnehmen will, muß die entsprechende Multicast-Adresse im IP als Empfängeradresse registrieren. Diese Registrierung und auch die Entregistrierung erfolgt dynamisch, also zur Laufzeit.

Wie weiß nun die Station, was auf welcher Multicast-Adresse gesendet wird? Dafür gibt es einerseits "Permanent Host Groups" (dies entspricht in etwa den "Well Known Ports" des TCP und UDP). ZB lauschen auf der Adresse `224.0.0.1` alle Stationen des Subnetzes, in dem der Sender sich befindet. Damit ist ein Subnetz-Broadcast implementierbar. Auf `224.0.0.2` kann man die Router des lokalen Netzes ansprechen, unter `224.0.1.1` antworten die NTP-Server ("Network Time Protocol"), etc.

Andererseits ist der MAC-Adressen-Bereich `00:00:5E:xx:xx:xx` für ein spezielles Konsortium namens IANA reserviert. Diese MAC-Adressen dürfen nur von Ethernet-NICs stammen, die von der IANA erzeugt werden. Nun erzeugt die IANA aber keine NICs, sondern diese Adressen werden als Ethernet-Multicast-Adressen verwendet. IP-Multicast-Adressen können damit auf IANA-MAC-Multicast-Adressen abgebildet werden:

```
Klasse-D-IP-Adresse:                              1110aaaa.abbbbbbb.cccccccc.dddddddd (32 Bit)
Ethernet-Multicast-Adresse:      00000001:00000000:01011110:0bbbbbbb.cccccccc.dddddddd (48 Bit)
```

Dabei werden die Bits "a" der Multicast-IP-Adresse verworfen, die Bits "b", "c" und "d" bilden die MAC-Multicast-Adresse. Die Abbildung IP-Adressen auf MAC-Adressen ist daher nicht eindeutig, da jeweils 2 ^ 5 = 32 verschiedene IP-Multicast-Adressen auf eine MAC-Multicast-Adresse abgebildet werden (die „weggeworfenen a“-Bits). Dies muß in der IP-Multicast-Software entsprechend behandelt werden, was aber unproblematisch ist, da sie im IP-Paket ohnehin die Multicast-Adresse als Destination Address vorfindet.

Interessant am Multicast ist auch, daß mehrere Prozesse auf einer Maschine dieselbe Multicast-Adresse dynamisch registrieren können. Das IP-System muß dann für alle diese Prozesse das Datenpaket kopieren und an diese weiterleiten.

Wirklich interessant wird es aber erst beim nächsten Router. Wie weiß ein Router, in welche seiner Netze er ein Multicast-Paket weiterleiten muß? Dies ist die Aufgabe des "Internet Group Management Protocols" ("IGMP", RFC 1112), das ebenso wie das ICMP ein Teil des IP-Systems ist.

Private IP Addressen

Die folgenden IP-Adressen sind vom IETF für “Intranets” reserviert. Maschinen, die diese Adressen verwenden, dürfen nicht direkt in das Internet geroutet werden. Maschinen, die in einem Intranet vernetzt sind, sollen Netz-Adressen aus diesem Adreßraum verwenden:

```
10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
```

Automatisch vergebene IP-Adressen im Windows (APIPA)

W2K vergibt automatisch IP-Adressen im Bereich `169.254.xxx.xxx/16` für Maschinen, die ohne irgendeine IP-Konfiguration gebootet werden. Dieses Feature hat den klangvollen Namen APIPA („Automatic Private IP Adressing“). Es ermöglicht die völlig konfigurations- und DHCP-lose Installation eines kleinen Netzes. Die Adressen werden beim booten „auf Verwendung geprüft“, später allerdings nicht mehr!

Ping

Um den Weg eines Datenpaketes zu einem Ziel zu ergründen, kann man den Befehl “Ping” verwenden. Das Kommando Ping startet eigentlich der ICMP-Echo-Dienst, der auf praktisch allen Plattformen zwecks Debugging von Netzwerkstrecken implementiert ist.

```
Pinging gw-vienna.bmc.com [192.168.137.253] with 32 bytes of data:
Reply from 192.168.137.253: bytes=32 time<10ms TTL=255
Reply from 192.168.137.253: bytes=32 time<10ms TTL=255
Reply from 192.168.137.253: bytes=32 time<10ms TTL=255
Reply from 192.168.137.253: bytes=32 time<10ms TTL=255
```

TraceRoute (tracert)

Ein weiterer, tiefergehender Befehl ist “tracert”, kurz für “Trace Route”. "tracert" gibt die einzelnen Teilstrecken aus, die ein Paket auf seinem Weg zum Ziel durchquert hat samt der in diesen Teilstrecken auftretenden Verzögerungen (eigentlich zeigt tracert die “Sub-pings” der Teilnetze an). Ferner wird die “DNS Inverse Address Resolution” (ein sogenannter “PTR-Request”) verwendet, um aus den IP-Adressen der Router deren DNS-Namen “zurückzurechnen”. Daher dauert ein "tracert" auch meist erstaunlich lange.

```
Tracing route to hpsrv01.infosys.tuwien.ac.at [128.131.172.20]
over a maximum of 30 hops:

  1    10 ms   <10 ms   <10 ms  gw-vienna.bmc.com [192.168.137.253]
  2    20 ms    30 ms    81 ms  192.168.105.85
  3   170 ms   201 ms   180 ms  192.168.97.97
  4   170 ms   170 ms   181 ms  gate25ha2.bmc.com [172.25.0.252]
  5     *        *        *     Request timed out.
  6   180 ms   221 ms   180 ms  fp-us-hou1.bmc.com [172.17.19.15]
  7   181 ms     *      191 ms  fw-us-hou2.bmc.com [172.17.1.236]
  8   180 ms   191 ms   180 ms  ben.bmc.com [198.207.223.226]
  9   171 ms   180 ms   190 ms  Internet-7206.bmc.com [198.207.223.253]
 10   190 ms   230 ms   221 ms  500.Serial0-1-1.GW1.HOU1.ALTER.NET [157.130.137.117]
 11   180 ms   180 ms   181 ms  113.ATM2-0.XR2.HOU4.ALTER.NET [146.188.240.158]
 12   180 ms   191 ms   190 ms  152.63.97.37
 13   180 ms   190 ms   181 ms  184.ATM7-0.BR1.DFW9.ALTER.NET [152.63.98.133]
 14   180 ms   271 ms   260 ms  137.39.23.218
```

```
 15   180 ms   190 ms   191 ms  dal-core-02.inet.qwest.net [205.171.25.49]
 16   190 ms   201 ms   190 ms  hou-core-01.inet.qwest.net [205.171.5.169]
 17   211 ms   220 ms   250 ms  wdc-core-03.inet.qwest.net [205.171.5.185]
 18   220 ms   220 ms   221 ms  wdc-core-02.inet.qwest.net [205.171.24.5]
 19   221 ms   230 ms   220 ms  jfk-core-01.inet.qwest.net [205.171.5.233]
 20   320 ms   261 ms   250 ms  Nyk-cr02.NY.US.kpnqwest.net [205.171.30.146]
 21   301 ms   310 ms   301 ms  Ledn-cr01.NL.kpnqwest.net [134.222.228.238]
 22   301 ms   320 ms   311 ms  Ffm-nr03.DE.kpnqwest.net [134.222.228.198]
 23   370 ms   311 ms   320 ms  Ffm-nr04.DE.kpnqwest.net [134.222.162.2]
 24   330 ms   321 ms   330 ms  Wie-ar02.at.kpnqwest.net [134.222.198.46]
 25   501 ms   331 ms   330 ms  feth1-0-0.cc02-wien.at.kpnqwest.net [134.222.161.21]
 26   320 ms   331 ms   340 ms  feth3-0-0.cc01-wien.AT.KPNQwest.net [193.154.145.11]
 27   321 ms   370 ms   351 ms  ser1-0.cc03-wcity.AT.KPNQwest.net [193.83.155.162]
 28   321 ms   350 ms   331 ms  feth0-0.cc02-wcity.AT.KPNQwest.net [193.83.156.11]
 29   320 ms   361 ms   330 ms  TUWIEN-KPNQwest.AT.KPNQwest.net [193.83.144.214]
 30   330 ms   371 ms   340 ms  fwb.nat.tuwien.ac.at [192.35.241.117]

Trace complete.
```

Netstat

Die Ausgabe des Kommandos “netstat -r” zeigt die aktuelle Routing-Tabelle an und sieht zB unter Windows NT / Windows2000 folgend aus:

```
C:\>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 10 4b f6 d2 37 ...... 3Com 3C575 Ethernet Adapter
0x3 ...44 45 53 54 42 00 ...... NOC Extranet Access Adapter
0x4 ...00 00 00 00 00 00 ...... NdisWan Adapter
0x5 ...00 00 00 00 00 00 ...... NdisWan Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway        Interface Metric
          0.0.0.0          0.0.0.0  192.168.137.253  192.168.137.190      1   (1)
        127.0.0.0        255.0.0.0        127.0.0.1        127.0.0.1      1   (2)
    127.127.127.0    255.255.255.0  127.127.127.127  127.127.127.127      1   (3)
  127.127.127.127  255.255.255.255        127.0.0.1        127.0.0.1      1   (4)
    192.168.137.0    255.255.255.0  192.168.137.190  192.168.137.190      1   (5)
  192.168.137.190  255.255.255.255        127.0.0.1        127.0.0.1      1   (6)
  192.168.137.255  255.255.255.255  192.168.137.190  192.168.137.190      1   (7)
        224.0.0.0        224.0.0.0  127.127.127.127  127.127.127.127      1   (8)
        224.0.0.0        224.0.0.0  192.168.137.190  192.168.137.190      1   (9)
  255.255.255.255  255.255.255.255  192.168.137.190  192.168.137.190      1  (10)
===========================================================================
```

Die konfigurierten Routen sind samt ihren Subnetzmasken ablesbar. Wenn ein Datenpaket weitergegeben werden muß ("geroutet werden muß"), wird es auf einer Ausgangs-IP-Adresse ausgegeben. Man nennt diese Ausgangs-IP-Adresse das "Interface". Unter dem "Gateway" meint das netstat-Kommando denjenigen Router, der das Paket dann ins nächste Netz weiterleiten wird.

Man sieht erst einmal den Default-Gateway-Eintrag (1). Er hat den Eintrag “`0.0.0.0`”, das steht hier für “alle anderen IP-Adressen". Er führt zum Router `192.168.137.253` via der eigenen IP-Adresse `192.168.137.190` (das ist die konfigurierte Default Gateway Adresse in der eigenen Station).

Dann folgen die Routing-Einträge für das Loopback-Routing (2, 3, 4). Dorthin weitergeleitete Daten gelangen nie an die unteren Schichten und damit auch nie “auf die Leitung". Anschließend kommt der Eintrag für das lokale Teilnetz `192.168.137` (5).

Der Eintrag (6) gibt an, daß Daten an die eigene IP-Adresse nicht über die eigene IP-Adresse versendet werden, sondern über den Loopback-Adapter gehen. Daher werden Daten an die eigene IP-Adresse nicht wirklich “über die Leitung” versendet.

Broadcast-Pakete an das eigene Subnetz (7) gehen über das eigene Interface `192.158.137.190` heraus.

Multicast-Pakete (8, 9) gehen sowohl an die Loopback-IP-Adresse als auch an die eigene IP-Adresse heraus.

Alle “vollen Broadcasts” (10) gehen über die eigene IP-Adresse weiter.

Für alle Routen ist ein einziger “Hop” notwendig (“Metric = 1”). Unter UNIX werden auch die Routing-Flags mitangezeigt:

U: Die Route ist aktiv (“up”).

G: Die Route führt zu einem vordefinierten Gateway. Fehlt dieses Flag, so sind Sender und Empfänger in selben Netz.

H: Die Route führt zu einem Host, die eingetragene IP-Adresse ist eine Stationsadresse, keine Netz-Adresse.
Fehlt dieses Flag, so führt die Route zu einem Netzwerk (die Host-Adresse ist dann 0).

D: Die Route wurde aufgrund eines "ICMP redirect" neu erstellt.

M: Die Route wurde aufgrund eines "ICMP redirect" modifiziert.

Ferner kann man mit “netstat -a” die TCP und UDP Ports auf ihren Status abfragen, zB:

```
C:\>netstat -an | find "ESTABLISHED"
  TCP    127.0.0.1:1027         127.0.0.1:1032         ESTABLISHED
  TCP    127.0.0.1:1032         127.0.0.1:1027         ESTABLISHED
  TCP    192.168.137.190:139    198.170.131.9:2613     ESTABLISHED
  TCP    192.168.137.190:199    192.168.137.190:1312   ESTABLISHED
  TCP    192.168.137.190:199    192.168.137.190:1313   ESTABLISHED
  TCP    192.168.137.190:1037   192.168.137.1:139      ESTABLISHED
  TCP    192.168.137.190:1172   198.170.129.5:1349     ESTABLISHED
  TCP    192.168.137.190:1177   198.170.129.5:1078     ESTABLISHED
  TCP    192.168.137.190:1182   198.170.129.5:1349     ESTABLISHED
  TCP    192.168.137.190:1308   172.17.3.110:139       ESTABLISHED
  TCP    192.168.137.190:1312   192.168.137.190:199    ESTABLISHED
  TCP    192.168.137.190:1313   192.168.137.190:199    ESTABLISHED
  TCP    192.168.137.190:1355   198.170.129.5:1078     ESTABLISHED
  TCP    192.168.137.190:1359   172.17.0.151:1145      ESTABLISHED
  TCP    192.168.137.190:1379   172.31.5.8:139         ESTABLISHED
```

Hier sieht man alle bestehenden TCP-Verbindungen (“ESTABLISHED”) des lokalen Rechners (“find” unter NT entspricht “grep” unter UNIX). Die Angaben erfolgen in der Form

```
Protokoll / lokale IP-Adresse:Portnummer / remote IP-Adresse:Portnummer / Zustand
```

Auch kann man damit zB diejenigen Ports, auf denen gerade ein UDP-Datagramm empfangen werden kann, auflisten:

```
C:\>netstat -an | find "UDP"
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:1168           *:*
  UDP    0.0.0.0:1169           *:*
  UDP    0.0.0.0:1170           *:*
  UDP    0.0.0.0:1178           *:*
  UDP    0.0.0.0:1179           *:*
  UDP    0.0.0.0:1180           *:*
  UDP    0.0.0.0:1314           *:*
  UDP    0.0.0.0:1318           *:*
  UDP    0.0.0.0:3181           *:*
  UDP    0.0.0.0:8161           *:*
  UDP    127.0.0.1:1256         *:*
  UDP    127.127.127.127:137    *:*
  UDP    127.127.127.127:138    *:*
  UDP    192.168.137.190:137    *:*
  UDP    192.168.137.190:138    *:*
```

"*:*" bedeutet, daß die remote IP-Adresse und der remote Port nicht bekannt sind. Bei UDP sind sie nur während des Empfangens eines Datagrammes bekannt und man sieht sie daher mit netstat nicht. Die Option „-n“ gibt die IP-Adressen numerisch aus. Dies erspart einen Reverse Lookup pro Ausgabe und damit etwas Zeit.

IPConfig / IFConfig

Unter MS Windows gibt es das Kommando “IPconfig”, mit dem man sich die Konfiguration der lokalen Maschine ansehen kann:

```
C:\>ipconfig /all

Windows NT IP Configuration

        Host Name . . . . . . . . . : cdemuth.bmc.com
        DNS Servers . . . . . . . . : 172.17.0.252
                                      172.19.0.252
        Node Type . . . . . . . . . : Hybrid
        NetBIOS Scope ID. . . . . . :
        IP Routing Enabled. . . . . : No
        WINS Proxy Enabled. . . . . : No
        NetBIOS Resolution Uses DNS : Yes

Ethernet adapter Elpc5751:

        Description . . . . . . . . : 3Com 3C575 Ethernet Adapter
        Physical Address. . . . . . : 00-10-4B-F6-D2-37
        DHCP Enabled. . . . . . . . : Yes
        IP Address. . . . . . . . . : 192.168.137.190
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.168.137.253
        DHCP Server . . . . . . . . : 192.168.137.1
        Primary WINS Server . . . . : 172.17.0.251
        Secondary WINS Server . . . : 172.19.0.251
        Lease Obtained. . . . . . . : Monday, February 19, 2001 8:08:53 AM
        Lease Expires . . . . . . . : Wednesday, February 21, 2001 8:08:53 AM
…
```

Das Kommando gibt die wichtigsten IP-Konfigurationsdaten und DHCP-Konfigurationsdaten aus. Man sieht oben unter “Windows NT IP Configuration” den DNS Namen der Maschine, die konfigurierten DNS-Server und ob das Routing aktiviert ist. Im unteren Abschnitt finden sich dann die IP-Angaben, die spezifisch für ein Interface vereinbart wurden. Interfaces können entweder „richtige“ NICs sein (wie in diesem Beispiel), aber auch „virtuelle Treiber“ wie zB VPN-Treiber (“Virtual Private Network”) oder RAS-Treiber (“Remote Access Server”). Die für die im PC eingebaute Ethernet-NIC mit dem Treibernamen „ELPC5751“ festgelegten Daten sind physische Adresse (MAC-Adresse), die IP-Adresse, die Subnetz-Maske, das Default Gateway und der DHCP-Server (von dem alle diese Daten bezogen wurden).

Unter Unix gibt es das Kommando “ifconfig -a”, das ähnliche Ausgaben liefert.

Autonome Systeme

Autonome Systeme sind IP-Netze, deren IP-Adressen nicht registriert sind und die daher nicht im Internet eingebunden werden können.

Um eine IP-Adresse zu registrieren, wendet man sich an einen IP-Adress-Verwalter (früher war das zentral das InterNIC, heute sind es lokale Betreiber, in Österreich zB www.nic.at) und bezieht von dort Netz-Adressen für Klasse A, B und C Netze (die Klassen A und B sind aber bereits alle vergeben). Nur mit solchen registrierten Adressen kann man ein autonomes System direkt ins Internet verbinden, da die Adress-Verwalter dafür sorgen, daß jede IP-Netzadresse nur ein einziges Mal vergeben wird.

In letzter Zeit haben aber viele Unternehmen, die sich auf TCP/IP als Netzwerkprotokoll standardisiert haben, große interne IP Netze als autonome Systeme aufgebaut. Diese werden als “Corporate Internets” oder besser als “Intranets” bezeichnet.

Sie verwenden meist “private IP-Adressen”. Man schafft dann einen Internet-Zugang mittels einer speziellen Art von Router, einer sogenannten “Network Address Translation” (“NAT”) Firewall. Diese übernimmt neben den Aufgaben einer konventionellen Firewall (Schutz vor Zugriffen von außen aus dem Internet) auch die Aufgabe, dem Intranet Zugriff ins Internet zu ermöglichen. Dazu erfolgt ein sogenanntes “Address Masquerading”. Die Firewall konvertiert die IP-Adressen aus dem Intranet, die ja im Internet illegal wären, auf eine einzige IP-Adresse im Internet. Diese allerdings muß klarerweise registriert sein. Die Firewall “merkt” sich die Intranet-IP-Adresse und den Port und die Internet-IP-Adresse samt Port. So werden auch retournierte Pakete aus dem Internet korrekt an die jeweiligen Knoten im Intranet zurückgeleitet werden. Die Firewall spielt also “Verkleiden” (“masquerading”) und täuscht dem Internet einen einzigen Knoten vor, hinter dem sich aber etliche tausend Intranet-Knoten verstecken können.

Für Mail, FTP und HTTP ist NAT eine praktikable Lösung. Vor allem kann sie auch bei bereits bestehenden Autonomen Systemen, die nicht dem IP-Adreß-Standard des Internets entsprechen, im Nachhinein angewendet werden. Das Konzept an sich ist aber nicht vollkommen transparent. Manche Protokolle höherer Schichten testen zB die IP-Adresse von Quelle und Ziel auch in der Schicht 7 nochmals nach (zB Oracle SQL*Net). Solche Protokolle sind daher nicht NAT-fähig. Auch einen Server kann man zB im Intranet nicht „hinter einer NAT“ stehen haben, da die interne IP-Adresse des Servers 1.) niemand im Internet kennt und diese 2.) im Internet nicht gültig wäre.

ICMP

Internet Control Message Protocol

ICMP dient der Benachrichtigung des sendenden Knotens im Fehlerfall (obwohl der häufigste Fehlerfall, nämlich verlorene Pakete, gar nicht gemeldet wird). ICMP transportiert seine Meldungen mittels IP-Paketen. Die meisten ICMP Pakete werden in demjenigen Knoten / Router generiert, der ein Problem erkannt hat und an die ursprünglich sendende Station retourniert. Der Echo Request ist eine Ausnahme, er dient nicht der Fehleranzeige, sondern dem Testen der Verbindung. Zu den wichtigsten ICMP-Meldungen gehören:

Echo-Request (+ Echo Reply)

Dieses Test-Paket wird auf den meisten Plattformen mit dem Kommando “ping <hostname>” ausgelöst. Es dient dem Testen von Verbindungen, sowohl End-to-End (Host-to-Host) als auch von einem Host zu einem Router. Für ein Beispiel zum ping siehe weiter oben.

<Destination> Unreachable

Anzeige des ICMP, daß ein bestimmter Teil des adressierten Systems nicht erreicht (angesprochen) werden konnte. Die <Destination> kann dabei sein:

Host (dieser Fehler wird von einem Router gemeldet),

Network (dieser Fehler wird von einem Router gemeldet),

Port (dieser Fehler wird vom Ziel-Knoten gemeldet) oder

Protocol (dieser Fehler wird vom Ziel-Knoten gemeldet).

Damit wird angezeigt, daß das IP-Paket nicht zustellbar ist und warum nicht. Zusätzlich wird ein

Problem im Zusammenhang mit dem „Don't Fragment" Bit angezeigt (Sender wollte, daß das IP-Paket nicht frqgmentiert werden darf, dies ist aber in einem Router nicht möglich, da dort eine zu kleine MTU vorgegeben ist).

Source Quench

Dies ist eine Flußkontroll-Nachricht. Der Empfänger bzw ein im Weg liegender Router zeigt damit an, daß er überlastet ist und fordert den Sender auf, die Paketrate zu drosseln ("quench"), andernfalls er IP-Pakete von diesem Knoten verwerfen wird (er "droht" damit quasi dem sendenden Knoten, dessen Pakete zu vernichten, was er auch darf). Der Ziel-Knoten bzw Router sollte Source Quench bereits absetzen, bevor seine internen Puffer überlaufen, er muß sie aber spätestens absenden, wenn sie überlaufen (RFC 1009). In einem Subnetz ausgesendete Source Quench Pakete deuten auf eine Überlastung von Routern dieses Netzes hin und sind daher auch für die Performance-Auswertung wichtig.

Route Change

Dieses Paket wird nur von Routern versendet und zwar dann, wenn er glaubt, daß er nicht den besten Weg zum Ziel für diese Pakete darstellt. Er muß in diesem Paket eine andere Route vorschlagen und diese dem ursprünglichen Sender des Pakets zusenden. Ein Router erkennt, daß etwas mit der "default route" nicht paßt, wenn das Netz, in das er ein Paket "forwarden" (weiterrouten) will, dasselbe ist, aus dem er das Paket erhalten hat. Die sendende Station ist also im selben Netz wie der Router, der das Paket weitersenden soll. Die sendende Station hat aber diesen Router nicht verwendet. Also empfiehlt der "falsche" Router, den "richtigen" zu verwenden. Route Change Pakete kommen also von Routern und sind für Stationen bestimmt.

Dieses Paket kommt in Netzen mit gut implementiertem Routing kaum vor, meist nur dann, wenn bei einer Station ein falsches Default Gateway konfiguriert ist.

Time Exceeded

Wenn die TTL auf Null heruntergezählt worden ist (was in einem Router geschieht), so verwirft der Router das IP-Paket und sendet stattdessen ein "Time Exceeded" ICMP-Paket an den Sender des IP-Paketes zurück. Das kann geschehen, wenn entweder die TTL zu klein gesetzt wurde (der Default-Wert beträgt meist 32 Hops bzw 32 Sekunden) oder eine zu lange Route verwendet werden mußte. Auch wenn die Reassemblierung des Datagrammes aus einzelnen Fragmenten im empfangenden Knoten nicht innerhalb eines bestimmten Zeitraums möglich war, sendet dieser ein "Time Exceeded" an den sendenden Knoten zurück.

Es gibt noch weitere ICMP-Pakete, deren Bedeutung allerdings nicht so groß ist bzw die heute nicht mehr oft verwendet werden.

ARP

Address Resolution Protocol

ARP ist ebenfalls ein Subprotokoll des IP, das die Umwandlung einer IP-Adresse in eine Schicht 2 Adresse (bei LANs: MAC-Adresse) vornimmt. ARP wird normalerweise nur bei Verwendung von LANs in den unterliegenden Schichten als Schnittstellenprotokoll verwendet.

Es wird ein Broadcast-Paket (Ziel-MAC-Adresse = `FF:FF:FF:FF:FF:FF`) abgesendet, in dem die gesuchte Ziel-IP-Adresse steht. Alle Stationen empfangen den Broadcast. Wenn die Station, deren IP-Adresse im ARP-Paket steht, das Paket empfängt, antwortet sie der sendenden Station mit einem

Antwort-Paket, in dem die IP-Adresse steht. In diesem Paket steht klarerweise auch immer die MAC-Adresse der Station (als Absender-Adresse in der SA).

Alle Stationen im Subnetz nehmen die IP-Adresse und die dazugehörige MAC-Adresse aus dem Antwort-Paket in ihren sogenannten ARP-Cache auf. Dieser ist bei manchen Betriebssystemen mit dem Befehl "`arp -a`" auslesbar:

```
C:\>arp -a
Interface: 192.168.137.190 on Interface 2
  Internet Address      Physical Address      Type
  192.168.137.1         00-50-8b-93-63-34     dynamic
  192.168.137.99        08-00-20-7c-23-48     dynamic
  192.168.137.100       08-00-20-86-c4-3a     dynamic
  192.168.137.253       00-e0-b0-63-f5-88     dynamic
```

RARP

Reverse Address Resolution Protocol

RARP ist ein Subprotokoll des IP, das für die IP-Konfiguration von disklosen Systemen geschaffen wurde. Es funktioniert ungefähr wie ein "verkehrtes" ARP (daher auch der Name). Die sendende Station kennt ihre eigene IP-Adresse nicht (und auch keine anderen IP-Adressen). Sie sendet ein RARP-Paket per Broadcast aus und im Subnetz muß ein RARP-Server konfiguriert sein, der auf RARP-Pakete antwortet.

Dieser hat eine feste Tabelle mit MAC-Adressen und zugeordneten IP-Adressen. Wenn sich ein Knoten per RARP meldet, verwendet der Server die Quell-MAC-Adresse des RARP-Pakets und sucht für den Knoten dessen IP-Adresse aus der Tabelle heraus. Diese wird ihm dann per Unicast-Paket zugesendet.

RARP ist ein Vorläufer des DHCP, das wesentlich leistungsfähiger (aber auch komplizierter) ist. DHCP ist für die einfache Konfiguration von vielen Arbeitsplätzen in einem IP Netz gedacht, RARP als sehr einfaches Protokoll für das Booten von disklosen PCs und Workstations (zusammen mit dem TFTP, dem „Trivial File Transfer Protocol" für das Laden des Bootcodes), die RARP als Protokoll im ROM integriert haben.

IPv6 alias IPng

Der Nachfolger des seit 1984 im RFC 791 definierten und seither in Verwendung befindlichen IP in der Version 4 ist das IP Version 6 bzw IPng ("IP Next Generation", RFC 2373). IPv4 hat einige Schwächen, die aus dem für EDV-Verhältnisse biblischen Alter des Entwurfes resultieren. Es hat eigentlich kaum jemand vermutet, daß das IP in dieser Version so lange und vor allem so erfolgreich im Einsatz bleiben wird. Daher sind einige Einschränkungen der Version 4 heute bereits problematisch:

1. Die Erschöpfung des 32 Bit Adressraums und die daraus resultierende "Zusammenstückelung" von größeren Netzen aus Klasse C Subnetzen. Damit wird die Administration erschwert und viele Unternehmen weichen auf eigentlich "unsaubere" Hilfstechnologien wie NAT ("Network Address Translation") aus. Diese bringen aber ihrerseits wieder Probleme, da sie nicht vollständig transparent für alle Anwendungen arbeiten.

2. Die Routing-Tabellen in den Backbone-Routern haben derzeit an die 70.000 Einträge. Diese Menge war nie vorgesehen und ist auch nur mühsam zu administrieren.

3. Die Konfiguration des IPv4 ist nicht automatisch möglich. Hilfsprotokolle wie RARP oder

DHCP, die Mängel des IP kaschieren, werden daher häufig verwendet.

4. Security-Aspekte (IPSec) werden im derzeitigen IP nur optional implementiert und man kann sich daher nicht auf deren Existenz in allen Systemen und Routern verlassen.

Die Version 6 unterscheidet sich daher von der Version 4 in den folgenden Punkten:

1. Neues IP-Header-Format, bei dem alle nicht immer gebrauchten Felder in sogenannte "Extension Headers" verschoben wurden. Die Extension Headers sind beliebig frei erweiterbar. Es gibt keine Prüfsumme mehr im Header.

2. 128 Bit IP-Adressen, damit 4x soviele Bits und 3.4×10^{38} individuelle IP-Adressen verfügbar. Zum Vergleich: das entspricht 6.5×10^{23} IP-Adressen für jeden Quadratmeter Erdoberfläche! Dies sollte Hilfstechnologien wie NAT wieder eindämmen.

3. Die Routingtabellen werden durch hierarchische IP-Adressvergabe kleiner.

4. IPSec ist ein Pflichtbestandteil von IP Version 6. Damit ist Security nun zwingender Bestandteil jeder IP-Implementation.

5. Unterstützung von sowohl dynamischen IP Konfigurationen (via zB DHCP) als auch "Selbstkonfiguration" ("Autoconfiguration") der einzelnen Knoten. Dazu wird die IP-Adresse aus der MAC-Adresse und einem Default Prefix berechnet.

6. V6 unterstützt erweiterte QoS-Kontrollmechanismen und die besondere Behandlung für IP-Pakete, die zu einem längeren Datenstrom ("flow") gehören.

7. Ein multicast-basiertes "Neighbor Discovery Protocol" ersetzt das broadcast-basierte ARP der v4 und einige Teile des ICMP der v4. Durch den forcierten Einsatz von Multicast-Adressen hat jeder IPv6 Knoten bereits defaultmäßig mehr als eine IP-Adresse, normalerweise eine automatisch gebildete und einige Multicast-IP-Adressen.

8. Es gibt keinen Broadcast "an alle Hosts in einem Netz" mehr. Er wird durch einen Multicast "an alle Hosts in einem Netz" ersetzt. Ferner wird der "Anycast" eingeführt, das ist ein Senden an den erstbesten Knoten, der sich von einer bestimmten Multicast-Adresse adressiert fühlt und antwortet. Der Anycast ist vor allem dadurch interessant, daß er eine Art funktionaler Adressierung a la Token Ring ermöglicht, dies aber über Routergrenzen hinweg.

Interessante Konzepte sind zB die Autokonfiguration. Jeder Knoten (genauer: jedes IP-Interface) erhält eine eindeutige sogenannte "Link Local" IP-Adresse, die folgend gebildet wird:

Die MAC-Adresse wird verwendet und das U/L-Bit wird invertiert. Diese Adresse wird in 2x24 Bit aufgeteilt. Dann wird `FF:FE` "dazwischengestellt" und damit eine 64 Bit Adresse, die kompatibel zur sogenannten EIU-64 Adressform ist, gebildet. Anschließend wird das Bitmuster `FE:80:00:00:00:00:00:00` davorgestellt.

ZB: die Ethernet-MAC-Adresse ist `00:AA:00:3F:2A:1C`. Nun wird das U/L Bit invertiert, das ergibt `02:AA:00:3F:2A:1C`. Dann wird die Adresse aufgeteilt in `02:AA:00` und `3F:2A:1C`. Nun wird `FF:FE` eingefügt, es folgt `00:AA:00:FF:FE:3F:2A:1C`, die EIU-64 Adresse. Anschließend noch der Prefix dazu, und schon ist die 128 Bit IPv6-Adresse fertig:
`FE:80:00:00:00:00:00:00:AA:00:FF:FE:3F:2A:1C`. Da im IPv6 die IP-Adressen in Hexadezimaler Notation geschrieben werden (so wie heute bei MAC-Adressen) und man jeweils 16 Bit zusammenfaßt, sieht die korrekte Schreibweise folgend aus:
`FE80:0000:0000:0000:00AA:00FF:FE3F:2A1C`

Da die neuen Adressen 128 Bit lang sind und dabei oft viele Nullen hintereinander beinhalten, kann man die IP-Adresse in einer Kurzform angeben, bei der die führenden Nullen jeder Teilsequenz weggelassen werden und zusätzlich, wenn alle 16 Bits einer Teilsequenz Null sind, diese durch ":" ersetzt wird. Aus der obigen Beispieladresse wird daher: `FE80::::AA:FF:FE3F:2A1C`, oder noch kürzer: `FE80::AA:FF:FE3F:2A1C`. Hier denotieren die "::" eine beliebige Anzahl von Null-Sequenzen. Insgesamt kann aus Gründen der Eindeutigkeit in einer IP-Adresse nur ein einziges Mal die Folge "::" vorkommen.

UDP

User Datagram Protocol

Das Schicht 4 Protokoll UDP ist die "Verlängerung" des IP-Protokolls in die Schicht 4. Der wichtigste Mechanismus, der dem IP hinzugefügt wird, ist der sogenannten "**Port**". Er ist eine 2 Byte Zahl (0..65535) und identifiziert eine Anwendung der oberen Schichten, die damit via UDP und IP auf die Transportmechanismen der Schichten zwei und eins zugreifen können.

UDP-Pakete sind sehr einfach aufgebaut:

Feld	**Bits**	**Bedeutung**
Absender-Port	16	Portnummer des Senders
Empfänger-Port	16	Portnummer des Empfängers
Länge	16	Länge des UDP-Pakets in Bytes
Prüfsumme	16	Prüfsumme über Header und Daten, Berechnung ähnlich wie beim IP-Header, allerdings nur 16 Bit
Daten	Var	Daten der oberen Schichten

Tabelle UDP Header

Die "Maximum Datagram Size" ("MDS") ist die maximale Länge eines UDP-Datagrammes. Die MDS ist einstellbar.

Wenn ein UDP-Datagramm nicht zustellbar ist - wenn also der Port auf der Empfängerseite nicht mit einem Betriebssystem-Aufruf für den Empfang des Datagrammes vorbereitet ist - wird ein ICMP-Datagramm "Port Unreacheable" an den Sender zurückgesendet. Dies ist allerdings von der Implementierung des TCP/IP-Stacks abhängig. Bei einigen TCP/IP-Varianten ist dieses Verhalten auch als Option einstellbar.

Die Übertragung eines UDP-Pakets ist nicht garantiert, ein UDP-Paket darf laut UDP-Definition auf seinem Weg zum Ziel verloren gehen oder vernichtet werden. Ob man ein gesendetes UDP-Paket auf der Empfangsseite auch als ein einziges UDP-Paket empfangen muß oder ob man es in mehreren Stücken empfangen kann, ist ebenfalls implementierungsabhängig.

TCP

TCP – Transport Control Protocol

TCP wurde implementiert, um die grundsätzlichen Schwächen von UDP zu verringern. TCP setzt ebenso wie UDP direkt auf dem verbindungslosen IP-Protokoll auf. TCP ist im Gegensatz zu UDP jedoch verbindungsorientiert, garantiert daher die Reihenfolge der Ankunft der Datenpakete. Ferner

werden verlorene IP-Pakete automatisch neu gesendet und doppelt ankommende Pakete eliminiert. Zusätzlich wird ein eigener Flußkontroll-Mechanismus integriert und Fehler in den IP-Paketen werden durch Neusenden behoben.

Dabei wird allerdings auf der Empfängerseite ein Datenstrom aus den einzelnen gesendeten Paketen erstellt. Diese werden also nicht in denselben Paket-Einheiten empfangen wie sie gesendet wurden. Dies ist beim ersten Verwenden des TCP-Protokolls etwas gewöhnungsbedürftig. Wenn man zB zwei Datenstrukturen mit ja 100 Bytes per TCP versendet, kommen "auf der anderen Seite" 200 Bytes ohne definierte Grenze zwischen den ersten 100 und den zweiten 100 Bytes an. Man kann diese 200 Bytes sogar mit einem einzigen API-Aufruf abholen und tut dies normalerweise auch. Man nennt daher einen TCP-Datenstrom auch folgerichtig einen "stream". Die Anwendung muß dafür sorgen, daß die einzelnen von der Anwendung erstellten Datenpakete "auseinandergehalten" werden können.

Die Implementierung des Datenstroms erfolgt mittels Sequenznummern in den einzelnen Datenpaketen. Es werden Rückmelde-Pakete (sogenannte "ACKs", kurz für "Acknowledge") verwendet und Flußkontroll-Pakete. Zusätzlich gibt es "Timeout"-Mechanismen und das damit verbundene wiederholte Senden von Paketen. Das Schließen der Verbindung erfolgt ebenfalls in kontrollierter Weise und kann von beiden Seiten der Verbindung her initiiert werden.

Jedes einzelne TCP-Paketes sieht folgend aus:

Feld	**Bits**	**Bedeutung**
Absender-Port	16	Portnummer des Senders
Empfänger-Port	16	Portnummer des Empfängers
Sequenznummer	32	Startposition (in Bytes) der Daten, die sich im Paket befinden, innerhalb des gesamten Datenstroms seit Verbindungsaufbau.
Bestätigungsnummer	32	Das korrekte Empfangen aller Bytes bis zu dieser Position (minus eins) wird vom empfangenden Knoten bestätigt.
Offset	8	Abstand in 4-Byte-Einheiten bis zum Feld "Daten". Werden keine Optionen im TCP verwende, beträgt der Wert hier 5 (=20 Bytes)
Flags	8	Siehe unten
Window-Size	16	Größe des Puffers in Byte, den der Endknoten (Empfänger) für diese Session reserviert hat. Der Sender darf nie mehr als diese Anzahl von Bytes senden ohne vorher eine Bestätigung zu erhalten.
Prüfsumme	16	Wie beim UDP berechnet.
Dringliche Daten Offset	16	Abstand in Bytes der "Dringlichen Daten" vom Beginn der Daten weg. Daten ab diesem Abstand werden als "Out of Band" betrachtet und vom Empfänger sofort bearbeitet. Dieses Feld gilt nur, wenn das Flag "URG" gesetzt ist.
Optionen	Var	Variable Anzahl an Options-Feldern im TCP-Header. Die wichtigste ist das Setzen der "Maximum Segment Size" ("MSS"), der "Windows Scale Option" und der "Timestamp Option"
Daten	Var	Daten der oberen Schichten + Dringliche Daten

Tabelle TCP-Header

Man sieht also, daß die Steuerinformation des Sliding Windows Mechanismus in jedem einzelnen Datenpaket der Verbindung mitgesendet wird. Man nennt dieses Verfahren "Piggy Backing", die Steuerdaten "reiten" quasi auf dem eigentlichen Datenpaket "oben drauf mit". (Warum aber das

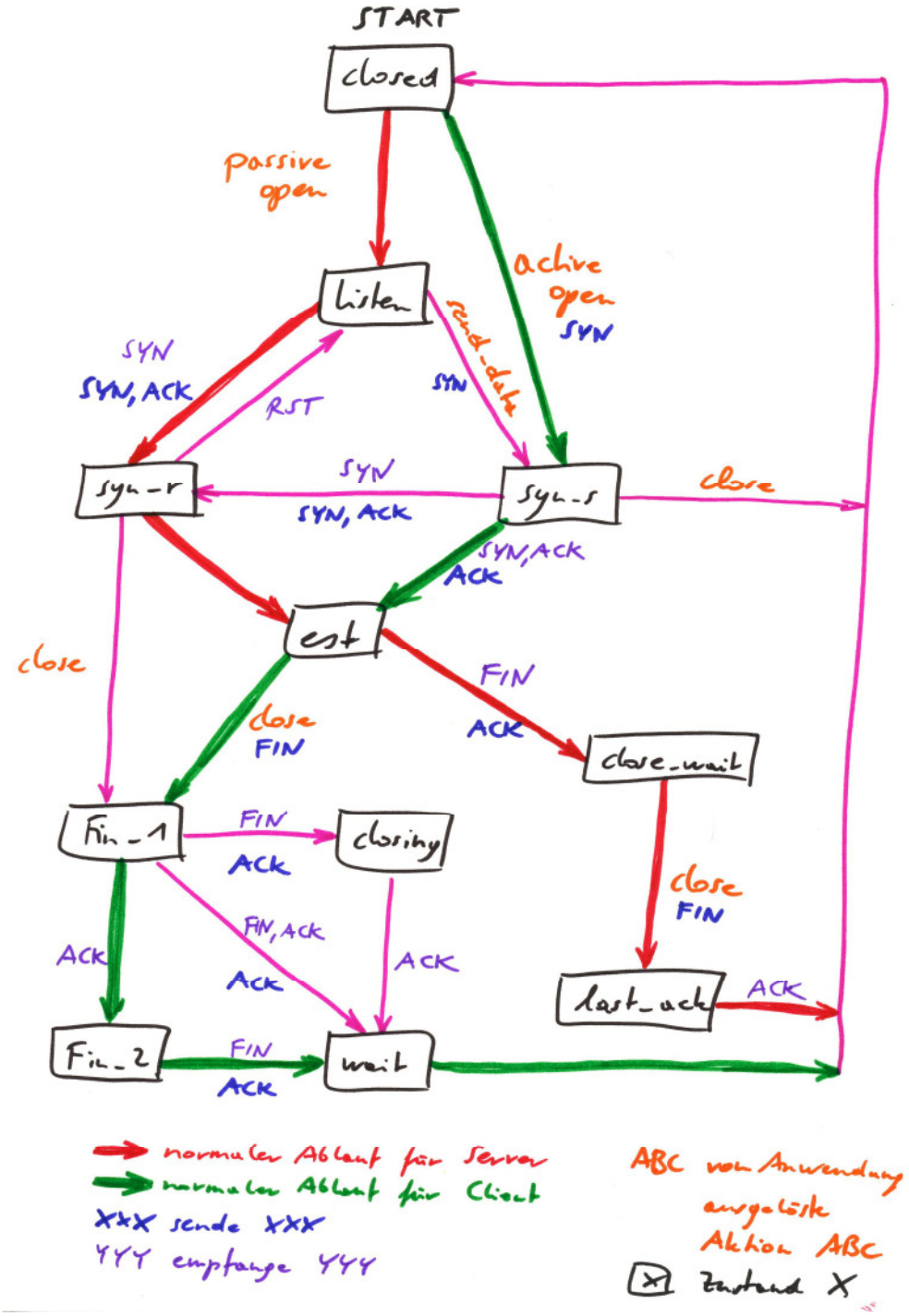
Datenpaket gerade als Schwein bezeichnet wird, ist interessant.)

TCP-Flags

Flag-Name	**Bedeutung**
“URG” (“urgent data”)	Mit diesem Flag zeigt der Sender an, daß die Daten, die im Feld “Daten” ab der Position “Offset” stehen, nicht zum normalen Datenstrom gehören, sondern sofort vom Empfänger aus dem Datenstrom entfernt und bearbeitet werden müssen. Damit kann man zB Breaks an eine Anwendung senden.
“ACK” (“acknowledge”)	Zeigt an, daß der Wert in “Bestätigungsnummer” korrekt und gültig ist. Wird auch beim Verbindungsaufbau verwendet,
“PSH” (“push”)	Weist den Empfänger an, die empfangenen Daten nicht weiter zu puffern, sondern sie sofort an die darüberliegenden Schichten weiterzuleiten. Anderfalls würde die TCP-Protokollmaschine solange IP-Pakete puffern, bis der Empfangspuffer des TCP voll ist und erst dann die Daten an die oberen Schichten weiterreichen. Das "PSH" Flag kann meist nicht direkt von der Anwendung angesteuert werden, sondern wird oft gesetzt, wenn die Anwendung keine Daten mehr an das TCP senden möchte (also wenn der Sendepuffer geleert wurde). Detto gibt es keine Methode, den Status des "PSH" Flags auf der Empfängerseite abzufragen.
“RST” (“reset”)	Wird nur im Notfall gesetzt. Zeigt einen unbehebbaren Fehlerzustand in der Verbindung an und trennt diese sofort. Ein gesetztes “RST” Flag des Empfängers (Verbindungs-Annehmenden) wird beim Verbindungsaufbau vom Sender (Verbindungs-Aufbauenden) als Ablehnung interpretiert.
“SYN” (“synchronize”)	Wird beim Verbindungsaufbau verwendet, bevor die Bestätigungsnummern ausgehandelt wurden.
“FIN” (“finalize”)	Wird beim Verbindungsabbau verwendet, um das Trennen der Verbindung zu initiieren. Sobald der Partner mit einem Paket mit gesetztem “FIN” Flag antwortet, wird die Verbindung tatsächlich getrennt. Bis dahin gilt die Verbindung nur als “einseitig gelöst” (“half close”) und wird weitergeführt. Sofortiges Terminieren der Verbindung ist mit “RST” möglich.

Tabelle Flags des TCP-Headers

Der allgemeine Ablauf der TCP-Protokollmaschine sieht folgend aus:



Skizze TCP Protokollmaschine (State-Transition-Diagram)

Verbindungsaufbau

Der Verbindungsaufbau des TCP-Protokolls erfolgt mit einem sogenannten “3 Way Handshake":

1. Der Client sendet ein IP-Paket mit SYN und einer beliebigen initialen Sequenznummer (CISN) im Feld Sequenznummer.

2. Der Server antwortet mit einem IP-Paket mit SYN und seiner initialen Sequenznummer (SISN) im Feld Sequenznummer. Er beantwortet auch die CISN mit CISN+1 im Feld

Bestätigungsnummer.

3. Der Client beantwortet das SYN des Servers mit einem ACK der SISN+1 im Feld Bestätigungsnummer.

Client	→ SYN 77 →	Server
Server	→ SYN 92, ACK 78 →	Client
Client	→ ACK 93 →	Server

Tabelle SYN-SYN-ACK

Die Sequenznummern können am Anfang beliebig gewählt werden, sollen aber möglichst selten wieder vorkommen.

Datenaustausch

TCP verwendet "Cumulative Acknowledgement" für die Datenübertragung. Dieser bewirkt, daß mit einem ACK für die Sequenznummer x alle empfangenen Bytes mit Sequenznummern kleiner als x bestätigt werden. Der Vorteil liegt darin, daß nicht jedes einzelne Paket mittels ACK bestätigt werden muß. Dies erhöht die Effizienz des TCP.

Zusätzlich muß man den Empfänger hindern, für jedes empfangene Datenpaket sofort ein ACK zu senden. Daher wird in den meisten Implementierungen 200msec gewartet. Erst dann wird ein Ack gesendet. Kommen in der Zwischenzeit weiter Daten, wird wieder 200msec gewartet ("Delayed Acknowledgement").

Ferner kommt der "Sliding Window Algorithmus" zum Einsatz. Das Window wird in Bytes gemessen und wird mit jedem empfangenen Paket um die Anzahl der empfangegen Bytes kleiner. Ist es auf 0 reduziert, so darf die sendende Station keine Daten mehr abschicken. Damit kann der Empfänger den Sender "drosseln" und eine einfache, aber effektive Form der "Congestion Control" (Flußkontrolle) anwenden. Das ACK des Empfängers trägt ebenfalls ein Feld "Window Size" mit und öffnet damit das Fenster wieder. Der Empfänger kann auch "Window Update" ACKs senden, mit denen er keine Datenpakete bestätigt (indem er bereits bestätigte Sequenznummern nochmals bestätigt), sondern nur das Fenster wieder für mehr Daten öffnet.

```
1     2     3     4     5     6     7     8     9     10     11     12  Datenbytes
--------------->                                                        gesendet + bestätigt
                ------------------>                                     gesendet, nicht bestätigt
                                   ------------------>                  kann noch gesendet werden
                                                       ----------->     noch nicht sendbare Daten
                <-----------------------------------><                  das Sliding Window
                                   <------------------>                 das "offene" Fenster
```

Skizze Sliding Window

Skizze Sliding Window Bewegung

Timeouts spielen beim TCP eine wichtige Rolle. Sie können an fast jeder Stelle des Protokollablaufs auftreten und müssen entsprechend behandelt werden. TCP führt vier Timer:

1. **Retransmission Timer**: jede TCP-Verbindung mißt den "Round Trip Delay" ("RTD") der Verbindung, indem für ein beliebiges gesendetes Byte im Datenstrom die Zeit zwischen dem Senden und dem ACK für dieses Byte gemessen wird. Diese Messung erfolgt laufend und der jeweils ermittelte RTD-Wert wird mit einer Konstante multipliziert und exponentiell geglättet. Er dient als Basis für den Retransmission Timer. Erfolgt für ein Datenpaket kein ACK vor Timerablauf, so wird das Paket oder das ACK als verloren eingestuft und wiederholt. Reagiert die andere Seite daraufhin wieder nicht, so wird eine Zeit gewartet und dann nochmals wiederholt. Dies geschieht mehrmals, mit "Binary Exponential Backoff" (siehe CSMA/CD), sodaß insgesamt einige Minuten mit Retries vergehen, bis die sendende Seite mit RST aufgibt.

2. **Persist Timer**: sorgt dafür, daß der Sender periodisch "Window Size" ACKs sendet, wenn die "Window Size" auf 0 gefallen ist. Dies ist dann wichtig, wenn das das Fenster öffnende "Window Size" Paket des Empfängers an den Sender verlorengeht (dies kann geschehen, da ACK Pakete im TCP nie von der anderen Seite ge-ACK-ed werden).

3. **Keepalive Timer**: Sorgt für den Wiederanlauf, wenn die Empfänger-Station abstürzt oder rebootet wird. Eine TCP-Verbindung, auf der keine Benutzerdaten gesendet werden, tauscht keine Daten zwischen den Stationen aus. Ist Keepalive implementiert (ist nicht zwingend), kann die "andere Seite der Verbindung" per Polling geprüft werden. Wird vom Server (normalerweise der passive Part, also der wartende Empfänger) verwendet, um das "Verschwinden" des Clients zu bemerken. Das Keepalive kann aber auch falschen Alarm geben, wenn zB nur ein Router auf der Strecke zwischen Sender und Empfänger kurzfristig ausfällt.

4. **2MSL Timer**: Mißt die Zeit, die die TCP-Verbindung im Zustand TIME_WAIT liegt. Dies ist ein Zustand beim aktiven Schließen der Verbindung. Ist der Timer-Wert größer als 2 mal die "Maximum Segment Lifetime" (ein fest eingesteller Wert der TCP-Implementierung, oft 1 oder 2 Minuten), so wird das letzte ACK-Paket als verloren betrachtet und wiederholt. Da meist der Client die Verbindung aktiv beendet und der Client ohnehin mit keinem festen TCP-Port arbeitet, bereitet das kein Problem. Schlimmer ist es, wenn der Server die Verbindung aktiv beendet und in den 2MSL Timer läuft. Dann gibt es die berühmte "Port alredy in use" Meldung des TCP, die anzeigt, daß der Port noch immer belegt ist, obwohl die Anwendung, die diesen Port verwendet, längst nicht mehr läuft. Abhilfe schafft das Wartenauf 2xMSL, also (meist) schlimmstenfalls 4 Minuten.

Im Laufe der Zeit wurde das TCP-Protokoll immer weiter optimiert. ZB ist eine "**Slow Start**" Optimierung heute in allen TCP-Implementierungen vorhanden. Dabei beginnt der Sender langsam mit dem Senden der Daten und steigert erst dann seine Sendegeschwindigkeit, wenn die ACKs auch entsprechend schneller zurückkommen. Damit wird verhindert, daß ein in der Strecke liegender Router von einem zu schnellen Start des Senders "überrumpelt" wird und dessen Puffer in der Folge überlaufen. In diesem Falle würde der Router nämlich einfach die zu schnell eintreffenden IP-Pakete verwerfen bzw mit Source Quench reagieren (aber das ist dann zu spät).

Auch der sogenannte "**Nagle-Algorithmus**" wurde bald zum Standard. Bei interaktiven Anwendungen, also solche, die eher "kleine" Pakete (aber dafür deren viele senden), ist der Overhead von IP (mindestens 20 Bytes) und TCP (nochmals mindestens 20 Bytes) und MAC (nochmals einige Byte) gegenüber den Nutzdaten (ein Tastendruck generiert ein paar Byte) ungünstig groß. Dies ist ein speziell bei langsamen Leitungen unangenehmes Verhalten. Der Nagle-Algorithmus (RFC 896) sagt, daß nur ein einziges "kleines" Datenpaket unbestätigt bleiben darf -

und hindert so den Sender, weitere "kleine" Datenpakete ohne ACK des Empfängers zu schicken, bis das Sliding Window auf 0 reduziert ist. Der Sender muß also die "kleinen" Pakete zusammensammeln, bis er ein ACK von der Gegenseite erhält und mudann die gesammelten kleinen Pakete als ein (relativ) großes Paket senden (nicht vergessen: TCP kennt keine Paketgrenzen und "darf" daher "Pakete zusammenlegen"!). Die Eleganz des Algorithmus liegt in der Selbst-Taktung: je schneller der Empfänger die ACKs sendet, desto öfter kann der Sender einzelne kleine Pakete schicken.

Verbindungsabbau

Der Abbau der Verbindung kann durch viele Mechanismen ausgelöst werden und in jedem Zustand des TCP-Protokolls auftreten. Der “normale” Abbau erfolgt in vier Schritten:

1. Der aktiv die Verbindung abbauende Rechner A setzt in einem IP-Paket das “FIN” Bit.
2. Der passive Rechner P setzt ein “ACK” Bit im nächsten Paket. Die Verbindung ist nun “halb geschlossen". Die Anwendung (die nächstobere Schicht) auf der Seite von P wird davon verständigt (“EOF” der Daten von A nach P).
3. P kann nach wie vor Daten an A senden, A aber nicht mehr an P.
4. P sendet ein IP-Paket mit dem “FIN” Bit gesetzt und schließt die Verbindung auf seiner Seite.
5. A bestätigt das Beenden mit einem Paket, in dem das “FIN” Bit mit einem “ACK” Bit bestätigt wird. Die Anwendung (die nächstobere Schicht) auf der Seite von A wird davon verständigt (“EOF” der Daten von P nach A).

A	→ FIN 192 →	P
P	→ ACK 193 →	A
P	→ Daten →	A
…		
P	→ Daten →	A
A	→ FIN 211 →	P
P	→ ACK 212 →	A

Tabelle Verbindungsabbau im TCP: FIN-ACK-FIN-ACK-Sequenz

Zusätzlich kann ein Paket mit gesetztem „RST“ Bit jederzeit und ohne ACK die Verbindung beenden. Dabei werden aber alle gesammelten und noch nicht gesendeten / empfangenen Daten verworfen.

TCP Optionen

Einige Optionen des TCP werden öfter eingesetzt. Die Verwendung der Option muß beim Verbindungsaufbau ausgehandelt werden.

ZB dient die “Timestamp-Option” dazu, den “Round Trip Delay” (“RTD”) zu bestimmen. Der RTD ist diejenige Zeit, die ein Datenpaket “hin und zurück” benötigt. Die Timestamp Option des TCP sendet dazu im ACK-Paket einen 32-Bit Wert mit, der ohne Einheit ist. Er muß nur aufsteigend

sein. Die meisten Implementierungen wählen zwischen einer Millisekunde bis zu einer Sekunde als Einheit. Der Empfänger retourniert in seinem nächsten ACK-Paket diesen Zahlenwert. Daraus kann der Sender den RTD bestimmen.

Die “Windows Scale Option” erweitert das Window von 16 Bit auf 32 Bit. Diese großen Window-Sizes sind notwendig, um bei sehr schnellen und langen Netzen (Gbps auf viele km) entsprechend effiziente Übertragungen zu gewährleisten.

Socket-Interface

Das Tupel (IP-Adresse + Portnummer + Schicht-4-Protokoll) definiert eine Anwendung der oberen Schichten. Diese Kombination bildet einen sogenannten “Socket” (RFC 793). Sockets werden sowohl zum Adressieren des Ziel-Systems verwendet aber auch auf der Sender-Seite gebildet. Ein Server-Programm “lauscht” (C-Funktionsaufruf “listen”) also auf einem definierten Port auf die Daten eines Clients, der eine Verbindung aufbauen will (im Falle von TCP) bzw ein Datagramm senden will (im Falle von UDP).

Wenn ein Client die Verbindung anfordert (TCP) bzw Daten sendet (UDP), so wird auch am Client ein Socket gebildet, und zwar mit der IP-Adresse des Clients, dem verwendeten Protokoll und einer freien, zufällig vom System gewählten Portnummer.

Die sogenannte “Socket-Abstraktion” bzw das “Socket-Interface” wird unter anderem unter Unix und Windows verwendet, um auf die Protokolle TCP und UDP (auch IP selbst ist möglich, über einen sogenannten “Raw Socket”) zuzugreifen. Sockets sind daher ein API, ein “Application Program Interface”, für die TCP/IP-Protokolle.

Ein Socket sieht für ein Anwendungsprogramm wie eine “File Handle” aus, d.h. man kann mit den konventionellen C-Funktionsaufrufen “read” und “write” darauf zugreifen, was die Möglichkeiten dieses Interfaces stark erweitert, da alle Anwendungen, die mittels Filezugriff ihre Daten aus Dateien holen, diese ohne Programmänderungen auch aus (allerdings bereits vor-geöffneten) Sockets holen können.

Die Socket-Schnittstelle nennt man unter Windows (Windows9x / WindowsME / Windows TN/ Windows 2000) auch “WinSock”.

Alternativen zur Socket-Schnittstelle sind zB das Streams-Interface unter Unix oder das TLI-Interface unter Windows NT.

TCP und Firewalls

Wenn mittels TCP eine Verbindung aufgebaut wird, so erfolgt auf der passiven („listen“, meist am Server) Seite nur der eigentliche Verbindungsaufbau auf dem angegebenen Port. Die eigentliche Datenübertragung erfolgt auf der passiven Seite dann über einen anderen, dynamisch gewählten (sogenannten „ephemeral“, Deutsch: flüchtig) Port. In den meisten Utilities wird aber auf Seite des passiven Verbindungsaufbaus dann der Ursprungs-Port angegeben.

Die Tatsache des „ephemeral Ports“ machte besonders einigen Firewalls älterer Bauart Probleme, da es für TCP-taugliche Firewalls notwendig ist, in die Datenpakete des TCP-Verbindungsaufbaus „reinzuschnüffeln“. Das konnten ältere Firewalls nicht.

Erst dort erfährt die Firewall dann, welcher Port tatsächlich für die Datenübertragung freizuschalten ist (das erfolgt dynamisch). Auch in die Datenpakete des Verbindungsabbaus muß die Firewall hineinschauen, um den dann nicht mehr benötigten Datenübertragungs-Port zu schließen.

Mit UDP ist es einfacher, da hier keine flüchtigen Ports verwendet werden.

DHCP

Dynamic Host Configuration Protocol

Das DHCP dient dem konfigurationslosen Betrieb von Workstations im Netz.

Das DHCP liefert ausgehend von einer MAC-Adresse die für die Konfiguation einer Workstation in einem TCP/IP Netz notwendigen Daten wie eine eindeutige IP-Adresse, die Subnetz-Maske, den Router (Default Gateway), den Domainname und die Domain. Es können noch andere Daten (frei definierbar, zB Time Server, Name Server, Log Server) mitgeliefert werden.

Dafür müssen sogenannte DHCP-Scopes definiert werden. Das sind Bereiche von Subnetz-Adressen, die verfügbar gemacht werden und "verleast" werden. Dazu muß auch noch eine "Lease-Dauer" definiert werden.

Ferner muß ein DHCP-Server pro Subnetz existieren, da die DHCP-Requests per Broadcast ausgesendet wird. Broadcasts dürfen aber im TCP/IP die Grenze des Subnetzes nicht passieren und werden daher vom Router nicht weitergeleitet. Alternativ kann man Router verwenden, die speziell nur die DHCP-Pakete durchlassen (Router nach dem RFC 1542).

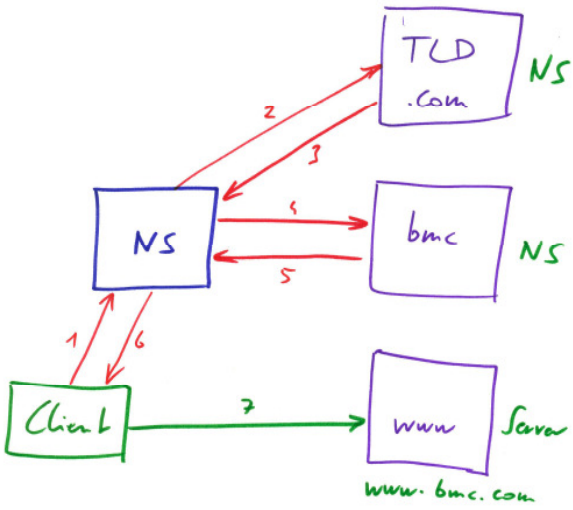
DNS

DNS - Domain Name Service

Der DNS bildet einen für Menschen "freundlichen" hierarchischen Rechnernamen wie zB `hpsrv01.infosys.tuwien.ac.at` in eine für den Computer "freundliche" IP-Adresse wie `128.131.172.20` ab. Intern im TCP/IP wird ausschließlich mit der IP-Adresse gearbeitet. Die Umwandlung erfolgt bereits ganz zu Anfang.

Für die Umwandlung wird eine Gruppe von DNS Servern verwendet. Diese replizieren untereinander die Informationen und sind außerdem für unterschiedliche Ebenen des Namensbaums zuständig.

Der Prozeß der Namensauflösung erfolgt in Form von rekursiven Aufrufen eines DNS-Servers an den "nächstobersten". Zusätzlich verfügt jeder DNS-Server über einen Cache, dessen Einträge oft nur sehr langsam ausalten. Da sich DNS-Namen und deren zugehörige IP-Adressen aber nur langsam ändern, macht das außer beim Testen selten Probleme.



Skizze DNS Lookup

Der sogenannte Reverse Lookup liefert aus einer IP-Adresse wieder einen DNS-Namen (Netz-Namen oder Host-Namen). Dazu muß die IP-Adresse in einen „DNS-tauglichen“ String konvertiert werden, dies erfolgt durch Rückwärts-Auflistung der IP-Adresse und Abfragen nach einer bestimmten Stamm-Domain:

Die Suche nach dem DNS-Namen der IP-Adresse `128.131.172.20` wird durch einen nslookup nach der DNS-Adresse `20.172.131.128.in-addr.arpa` ausgelöst.

Routing im IP

Routing ist eine der wichtigsten Aufgaben in einem Internet. Es ermöglicht das Zusammensetzen eines Gesamtnetzes aus einzelnen Teilnetzen und deren koordiniertes Zusammenspielen in einem Internet. Die Komponenten, die die einzelnen Teilnetz verbinden, sind die sogenannten “Router”. Die Aufgabe der Router besteht darin, Datenpakete der IP-Schicht von einem Teilnetz in das nächste weiterzuleiten. Da sich ein Router ausschließlich mit IP-Paketen beschäftigt, kennt er nur Datagramme und routet daher jedes Datenpaket einzeln.

Aufgaben eines Routers

Eine Metrik für das Routen von Paketen verwenden. Mit Metrik ist hier eine Bewertung der einzelnen Wege zum Ziel anhand bestimmter Kriterien gemeint. Die am häufigsten verwendeten Metriken sind “Number opf Hops” (Anzahl der “Hüpfer” = Teilnetze auf dem Weg) und kostenbasierende Metriken (“Kosten” der Ablieferung eines Datenpaketes am Ziel anhand der vordefinierten “Kosten” der Teilstrecken). Gute Metriken bewirken, daß Router den effizientesten Weg zum Ziel finden.

Ein Router sollte die Möglichkeit bieten, redundante Strecken zumindest zu kennen und diese im Bedarfsfall auch zu verwenden (“Fail over”). Eine bessere Strategie ist es, falls mehrere Wege zum Ziel existieren, diese auch zugleich zu verwenden (“Load Balancing”). Diese gleichzeitige Verwendung sollte aber nicht wahllos erfolgen, sondern immer an eine Metrik gebunden sein. Ansonsten würde zB bei zwei Strecken zum Ziel, wobei die erste eine schnelle 155 Mbps ATM-Strecke ist und das Backup eine 64Kbps ISDN-Wählleitung, immer die Hälfte der Datenpakete über die ISDN-Leitung gehen. Diese ist aber hier als reines Backup gedacht. Ein intelligenter Router schickt alle Pakete über die ATM-Strecke und öffnet die ISDN-Leitung nur im Fall eines ATM-

Defekts bzw wenn die ATM-Leitung überlastet ist (dann helfen die zusätzlichen 64Kbps aber auch nicht mehr viel).

Klarerweise sollte ein Router nicht sehr viele Datenpakete selbst erstellen und versenden, sondern möglichst wenig Netzlast konsumieren. Zusätzlich ist es heute oft von Vorteil, wenn das Routing-Protokoll ohne IP-Broadcasts auskommt. Diese werden nämlich auf LAN-Broadcasts umgesetzt und belasten damit alle in einem Subnetz befindlichen Systeme, da ein Broadcast von allen Stationen eines Netzes empfangen und verarbeitet wird.

Das rasche Konvergieren der Routing-Tabellen der einzelnen Stationen ist ebenfalls wichtig. Damit ist gemeint, daß sich Strukturänderungen im Netz möglichst rasch zu allen Routern fortpflanzen und sich die Routing-Tabellen auf demselben Wissensstand befinden.

Und nicht zuletzt sollte ein gutes Routing-Protokoll in der Lage sein zu skalieren, also nicht nur in kleinen Netzen mit -zig Routern zu funktionieren und effizient zu sein, sondern auch in großen Netzen mit tausenden Routern.

Typen von Routing-Protokollen

1. Solche, bei denen jeder Router jeden anderen Router kennt. Diese Protokolle verwenden einen sogenannten “Vector-Distance” Algorithmus. Durch die vollständige Bekanntschaft aller Router sind diese Algorithmen allerdings nur für eine überschaubar kleine Anzahl von Routern und damit eigentlich nur als IGPs geeignet.

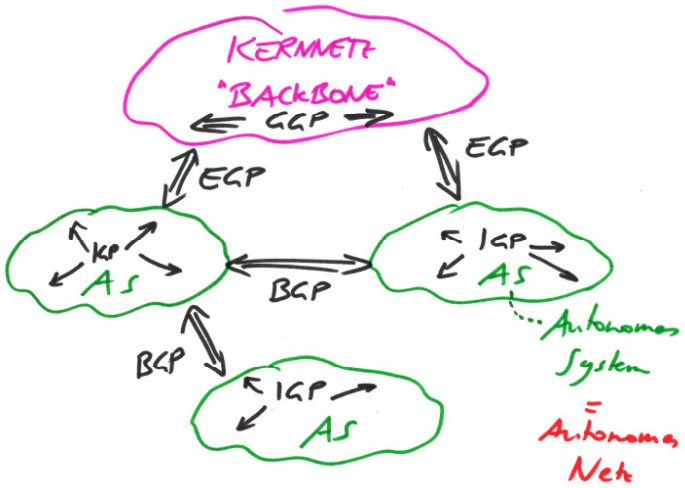
2. Algorithmen, bei denen jeder Router nur seine direkten “Nachbarrouter” kennt (“Link-State-Algorithm”). Diese Router-Typen sind skalierbar, aber komplexer zu erstellen und komplexer in der Anwendung.

Fixe Routenwahl

Alternativ zum Routing-Protokoll kann man im Router auch eine fixe Routing-Tabelle vorprogrammieren. Dies ist in “Endnetzen” sinnvoll, die nur über einen einzigen Router an das Internet angebunden sind. In diesem Fall muß nur “der Weg raus” voreingestellt werden (“Default Route”). Der “Weg rein” muß als “statische Route” beim Uplink Provider installiert werden.

Arten von Routing-Protokollen im TCP/IP

Im Internet unterscheidet man mehrere Arten von Routing-Protokollen, je nachdem, an welcher Stelle des Netzes diese eingesetzt werden. Die Grenzen zwischen diesen Protokoll-Arten sind aber z.T. unscharf, so kann zB ein EGP auch ein IGP sein.



Skizze Routing-Protokoll-Arten

Im Bereich des Internets kommen eine Menge anderer Routing-Protokolle zum Einsatz, die insgesamt als xGPs (xGP = x **Gateway Protocols**) bezeichnet werden. Es gibt die folgenden Varianten:

IGP	"Interior-Gateway-Protocol": Protokoll-Art, die innerhalb von Autonomen Systemen verwendet wird. Viele verschiedene IGPs sind bekannt und im Einsatz.
EGP	"Exterior-Gateway-Protocol": Protokoll-Art, die die Anbindung eines Autonomen Systems an den Internet-Backbone ermöglicht.
BGP	"Border-Gateway-Protocol": Protokoll-Art, die ein AS an ein anderes AS anbindet (sogenanntes "Peering"). Derzeit ist nur ein einziges BGP in breiter Verwendung, das "BGP-4". BGP-4 erlaubt das direkte und auch das indirekte Anbinden von AS an das Internet. Dabei wird ein AS über ein anderes AS an das Internet gebunden.
GGP	"Gateway-to-Gateway-Protocol": Einsatz als Backbone-Routing-Protokoll im Kernbereich des Internets. Verwalten sehr großer Routing-Tabellen. Benötigt meist einen spezialisierten Computer.

Tabelle Routingprotokollarten

RIP

Routing Information Protocol

Das RIP ist eines der Ur-IGPs. Es ist in so ziemlich allen UNIX-Derivaten in Form des "routed" Daemon implementiert. RIP ist ein sehr einfaches Protokoll, bei dem die aktuelle Routing-Tabelle des Routers alle 30 Sekunden per IP-Broadcast versendet wird.

Die initiale Routing-Tabelle eines Routers wird aus den eigenen IP-Adressen des Routers zusammengestellt. Wenn der Router eine Routing-Tabelle eines anderen Routers empfängt, so bildet er die Vereinigungsmenge der eigenen Routing-Tabelle und der empfangenen und verwendet diese Menge als neue eigene Routing-Tabelle. Da jeder Router seine Routingtabelle alle 30 Sekunden in alle ihm bekannten Netze weitersendet, wird die akkumulierte Routingtabelle schlußendlich in allen Routern bekannt (sogenannter "Vector Distance Algorithmus").

Sendet ein Router nicht mehr regelmäßig seine Tabelle, so werden dessen Einträge aus den Tabellen der anderen Router "herausgealtert" (mittels Timeout im Minutenbereich).

Der Nachteil dieses sehr einfachen Routing-Verfahrens liegt darin, daß sich Änderungen in der

Routing-Tabelle eines Knotens (wenn zB dessen Nachbar sich nicht mehr meldet) nur sehr langsam von Router zu Router weiterbewegen und damit lange Zeit falsche oder nicht mehr existierende Wege versucht werden. Dies wird vom ICMP gemeldet, führt aber dazu, daß Pakete verlorengehen und von höheren Schichten nochmals gesendet werden müssen.

Die produzierte Netzwerklast, die Broadcasts und die langsame Konvergenz des Protokolls limitieren es auf den Einsatz als reines IGP eines AS und zusätzlich auf maximal 15 Router innerhalb des AS.

Durch “Patches” wurden einige der gröbsten Unzulänglichkeiten des RIP im Laufe der Zeit ausgebessert, aber RIP ist dennoch nur in kleinen AS sinnvoll einsetzbar. Der große Vorteil des RIP liegt sicherlich darin, daß es konfigurationslos und weit verfügbar ist.

OSPF

Open Shortes Path First

OSPF ist ein relativ neues Routing-Protokoll, das 1997 in der v2 herauskam (RFC 2178). OSPF ist ein IGP und verwendet einen “Link-State”-Algorithmus, der als “Dijkstra-Algorithmus” bekannt ist. OSPF sendet seine Daten direkt über IP, merkt sich die Subnetzmaske jedes Teilnetzes zusätzlich (für Supernetting, CIDR) und verwendet effizientes IP-Multicasting, um mit den anderen Routern zu kommunizieren.

Bei OSPF werden in den einzelnen Routern die Verbindungsinformation (“Link”) und der Zustand der Verbindungen (“State”) verwaltet. Wenn ein Link “up” ist, können Daten über diese Verbindung zum nächsten Netz transportiert werden. Über einen Link, der “down” ist, können keine Daten transportiert werden. Die einzelnen Router eines Systems haben alle dieselbe Link-State Datenbank (im Normalfall, außer wenn sich das Netz gerade rekonfiguriert).

Wenn ein Router ein Datenpaket weiterleiten soll, errechnet er für dieses Datenpaket aus der Link-State Datenbank den kürzesten Weg des Datenpakets zum Ziel. Derjenige Ausgang, der auf dem kürzesten Weg liegt, wird für das Weiterleiten des Pakets verwendet. Die Routing-Tabelle existiert also in einem Link-State Router nicht wirklich, sondern wird pro IP-Paket “on the fly” erstellt und wieder verworfen. Alternativ zum kürzesten Weg können auch andere Werte (QoS) die Wegwahl beeinflussen. Damit kann für ein und denselben Weg van A nach B eine verschiedene Route abnhängig von den QoS-Parametern des IP-Datagrammes bestimmt werden.

Im OSPF wird eine einheitslose Metrik (man kann sie als Kosten interpretieren) verwendet, also jedem Port des Routers wird eine Zahl zugewiesen. Der Dijkstra-Algorithmus sorgt dafür, daß immer der metrik- (kosten-) mäßig minimale Weg verwendet wird. Im Falle eines Leitungs- bzw Routerausfalls rekonfiguriert sich die Link-State Datenbank und der nächste verfügbare Weg wird gewählt, so noch einer existiert.

OSPF unterstützt auch in eingeschränkter Form Load Balancing. Bei zwei oder mehreren gleich großen Gesamt-Metrikwerten, die aber über verschiedene Ports des Routers gehen würden, können die Datenpakete auf einem beliebigen dieser Wege gesendet werden. Dabei wird auf gute Streuung auf alle möglichen Ports geachtet (“Round Robin” Verfahren).

EGP

Exterior Gateway Protocol

Das EGP ist ein Vertreter der EGPs. Ab hier verwenden wir EGP als Abkürzung für das 1982

definierte Protokoll “EGP” (RFCs 827, 904). Das “Ur-EGP” kennt nur eine baumförmige Topologie, mit der Wurzel des Baumes als einziger Verbindung in den Internet-Backbone.

EGP sendet alle zwei Minuten die vollständige Routing-Tabelle an die anderen EGP-Router. EGP pollt alle 30 Sekunden alle seine EGP-Partner. EGP versendet seine Daten direkt über IP.

Das EGP kann man technisch gesehen auch als IGP verwenden.

BGP-4

Border Gateway Protocol

BGPs werden zum Anbinden von AS an andere AS verwendet. Dies führt zu zusätzlicher Redundanz im IP-Netz, da außer den direkten "Uplinks" (das sind die Anbindungen des AS an den Internet-Kernbereich) auch andere ‚alternative Leitungen zu anderen AS verwendet werden können. BGPs sind nicht für alle Betreiber von AS sinnvoll, aber größere kommerzielle AS Betreiber verwenden BGPs gerne für das sogenannte "Peering". Dabei wird zu einem anderen AS eine direkte Verbindung aufgebaut, die den Umweg über den Internet-Kernbereich unnötig macht und damit die Verbindung effizienter. Speziell die AS einer lokalen geografischen Region sind oft per Peering verbunden, da die Internet-Nutzer oft innerhalb dieser Region ihre Internet-Dienste beziehen. Es haben sich daher auch sogenannte Börsen (Englisch: Exchange) gebildet, die als Knotenpunkt für den Datenaustausch mehrerer AS fungieren.

BGP-4 (eigentlich “BGP v4”, RFC-1771) ist der derzeit wichtigste Vertreter der BGPs. Das Protokoll wurde 1995 definiert und ist heute in weiter Verwendung. Es verbindet Autonome Systeme direkt und wird auch als EGP bzw als IGP verwendet (“eBGP” und “iBGP”).

BGP-4 baut auf TCP auf (Port 179), verwendet einen OSPF-ähnlichen Algorithmus (Link State) und versendet nur die Änderungen an den Routen. Metrik-Daten eines IGPs können in BGP-4 übernommen werden.

BGP-4 verwendet kein “Load Balancing”, es kennt und installiert nur jeweils eine “beste Route”. Als iBGP eingesetzt müssen alle AS wechselseitig bekannt gemacht werden (sogenanntes “Route Mesh”), da ein iBGP-Router seine Routing-Informationen nicht transitiv an iBGP-Router in anderen AS weitergibt (dies wurde ausgeschlossen wegen möglicher Schleifenbildung). Eine Sonderform des BGP-4 (“BGP-4 mit Route-Reflector”) unterstützt aber die Weitergabe der Routing-Informationen über mehrere AS hinweg.

Unix GATED bzw ROUTED

Diese Routing-Daemone aus der Unix-Welt beherrschen die folgenden Routing-Protokolle:

Protokoll	**RIP**	**OSPF**	**EGP**	**BGP**
`routed`		`v1`		
`gated v2`	`v1`		`v1`	`v1`
`gated v3`	`v1, v2`	`v2`	`v1`	`v2, v3`

Tabelle Routing-Daemone im Unix

CIDR

Classless Inter Domain Routing (RFC 1518, RFC 1519)

Hinter diesem Schlagwort verbirgt sich ein Konzept, das man einfacher und treffender als

"Supernetting" (im Gegensatz zum Subnetting) bezeichnen könnte. Es erlaubt, zB mehrere Klasse-C-Netz-Adressen als eine einzige Netz-Adresse zu betrachten. Dabei kann man sich vorstellen, daß die Subnetzmaske (siehe oben) kürzer wird, also die Anzahl der Eins-Bits geringer wird.

Voraussetzungen für das Supernetting sind:

1. Die betroffenen Netz-Adressen haben dieselben Bits (von links her gesehen).
2. Es muß die Subnetzmaske immer mitgeführt werden, auch in den Routingtabellen der Routingprotokolle (derzeit in allen wichtigen Routing-Protokollen wie OSPF, RIP v2 und BGP-4 implementiert).

Der Vorteil des CIDR liegt darin, daß man sich in den Routern Einträge erspar. Wenn zB ein ISP einem Kunden 16 Klasse-C-Netzandressen übergibt, da dieser mehrere hundert Maschinen anschließen will, so kann er die 16 Netzadressen so wählen, daß sie dieselben "High Order Bits" haben (also von links gelesen sich nur in den letzten 4 Bit unterscheiden). Dann kann er diese 16 verschiedenen Netze mit einem einzigen Eintrag in seinen Routing-Tabellen in das Routing-System einbinden, wenn seine Router CIDR beherrschen.

MPLS

Multi Protocol Label Switching

Hier werden spezielle Router vorausgesetzt, die MPLS verstehen. Diese Router bilden ein in sich geschlossenes Teil-Netz mit mehreren Zugangsstellen. Innerhalb dieses Teilnetzes wird nicht nach einem Routingprotokoll geroutet, sondern nach fest eingestellten Wegen geswitcht. MPLS ist also eine performancesteigernde Technologie.

Dazu wird jedem IP-Paket beim Eintritt („ingress“) ins MPLS-Teilnetz eine „Label“ (32 Bit, hauptsächlich besteht aus dem eigentlichen Label mit 20b und einem eigenen Label-TTL mit 8b) verpaßt. Dieses Label wird anschließend für das Switchen des IP-Pakets auf Schicht 3 durch die MPLS-Router verwendet. Beim Austritt („egress“) aus dem MPLS-Teilnetz wird das Label wieder entfernt. MPLS ist also vollständig transparent, sogar für IP-Router, und damit protokollunabhängig.

Das Label bestimmt den Weg des Pakets durch das Teilnetz und damit dessen QoS. MPLS wird daher auch als „Traffic Engineering“ Verfahren gewertet.

Zusammenfassung

Die Lokalen Netze verschmelzen heutzutage immer mehr mit dem Internet. Die meisten der heute in Betrieb befindlichen Stationen eines LANs sind bereits mit dem Internet verbunden. Die "Lokalität" der Lokalen Netze ist daher heute nur noch durch die Definition "ein lokales Netz ist im Privatbesitz" zu finden.

Nach wie vor gilt aber als Grunddefinition der LANs der Privatbesitz und die beschränkte geografische Erstreckung (bis einige km). Die Datenrate der LANs dagegen hat sich im letzten Jahrzehnt von Mbps zu Gbps vertausendfacht. Die im Gbps Bereich arbeitenden LANs sind mit den MANs verschmolzen. Die IEEE 802, das Ur-Normungsgremium der LANs, nannte sich daher auch (kurzzeitig) LMSC („LAN and MAN Standards Committee“).

Die eigentliche Zusammenschaltung der LANs zu firmenweiten Intranets erfolgt heute über Frame

Relay- oder ATM-Technologien. Es kristallisiert sich aber immer mehr heraus, daß das Ethernet sich anschickt, den gesamten LAN- und MAN-Bereich zu erobern und auch in den Wirkungsbereich der WANs vorzudringen. Der Preis dafür ist aber, daß das Ethernet - so wie es heute in den Hochgeschwindigkeits-Varianten existiert - kaum noch etwas mit der ursprünglichen Definition gemein hat außer dem Namen und dem groben Frame-Format.

Lichtwellenleiter etablieren sich auf kurz oder lang als das zukünftige Kabelsystem. Die Twisted Pair Technologie ist ausgereizt, die LWLs sind heute bereits günstiger als hochqualitative TPs. Durch den verstärkten Einsatz der SMFs werden Entfernungen ermöglicht, die vor wenigen Jahren noch den WANs vorbehalten waren - und das unter Verwendung bekannter LAN-Protokolle. WDMF erschließt den SMFs heute schon den Tbps-Bereich und den Pbps-Bereich in naher Zukunft.

Als zukunftssicherste Topologie der LANs hat sich der Stern (Baum) erwiesen. Er erfüllt alle Anforderungen in Punkto Ausfallssicherheit und Skalierbarkeit. Durch die hohe Zuverlässigkeit der Technologie ist die Redundanz der Netzwerk-Hardware nicht mehr Aufgabe des LANs, sondern wird ausgelagert und extern behandelt (zB durch Verdoppelung des gesamten LANs).

Im Endknotenbereich setzen sich immer mehr die WLANs („Wireless LANs“, „WiFi“) durch. Sie verwenden Radiosignale zur Kommunikation und benötigen daher keine Verkabelung. Sie sind framemässig ebenfalls Ethernet-basierend. Da die Datenraten auch schon in die 50 Mbps reichen, ist eine weitere Verbreitung dieser Technologie vorhersehbar. Die Kosten sind bereits im auch für Privatanwender leistbaren Bereich, der Knackpunkt liegt in der LAN-typischen „einer sendet alle hören mit“ Problematik, die bei Funknetzen auch stark missbräuchliche Verwendungen ermöglicht.

Bei der Kodierung sind effiziente Verfahren, die den Einsatz hochspezialisierter Hardware (zB digitale Signalprozessoren, "DSPs") voraussetzen, der Standard.

Aus der Sicht der "oberen Schichten" (ab Schicht 3) wird die Dominanz des Protokollsystems TCP/IP immer deutlicher. Durch das Internet und seine Haupt-Anwendungen eMail und WWW getrieben, ist TCP/IP heute praktisch das einzig relevante Protokollsystem auf den Schichten "über dem LAN". Auch im Bereich der Mobiltelefonie setzt sich - zumindest in Europa - bereits mit GPRS und erst recht beim Hochgeschwindigkeits-Standard UMTS die Verwendung von IP als Trägerprotokoll bei den "Handies" durch.

Der langfristige Trend im Bereich der Computer Netze ist derzeit eindeutig die Konsolidierung des gesamten Marktes auf ein einziges Produkt, das Ethernet. Dieses ist heute bereits mit einer Datenrate von 1 Gbps günstigst verfügbar. Der Übergang auf 10Gbps erfolgt aus heutiger Sicht bereits im Jahr 2005. (Anm: Wir schreiben 2005, aber offensichtlich ist 10 Gbps doch etwas mehr als man im Durchschnitt bei einem LAN braucht und der Durchbruch des 10GBPS Ethernat hat noch nicht stattgefunden.) Ferner wird die maximale Leitungslänge des Ethernet (bei Verwendung von hochqualitativen LWLs) in den Bereich bis zu 50 km vordringen. Damit ist das Ethernet in den Bereich der MANs gelangt und ragt auch immer weiter in den WAN-Bereich hinein. Eine Zukunft, in der sowohl die LAN-, als auch MAN- und auch die WAN Technologie Ethernet heißt, ist also aus heutiger Sicht nicht mehr unmöglich, sie ist sogar wahrscheinlich.

Insgesamt kann man sagen, daß im Bereich der Computer Netze nur noch zwei Produkte eine entscheidende Rolle spielen: Ethernet (IEEE 802.3, CSMA/CD) für die Schichten 1 und 2 und das TCP/IP Protokollsystem für die Schichten 3 bis 7.

Man wird vielleicht in einigen Jahren nicht mehr von LANs, MANs oder WANs sprechen, sondern nur noch von "dem Netzwerk". Die Trägertechnologie dieses universellen Netzes stammt aber in diesem Szenario zum größten Teil aus derjenigen technologischen Ecke, die man heute als "Lokale

Netze" bezeichnet.

Dr. Christian Demuth
Externer Lektor
Technische Universität Wien
Institut für Informationssysteme