TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

Institut für Breitbandkommunikation

Unterlagen zu den Vorlesungen

DATENKOMMUNIKATION

und

KOMMUNIKATIONSPROTOKOLLE

Teil 3

o. Univ. Prof. Dr. Harmen R. van As

Technische Universität Wien, Institut für Breitbandkommunikation

A-1040 Wien, Favoritenstr. 9-11/388

Teil 3: Internet-Referenzmodell und dessen Realisierung

Version: Dez. 2003

Teil 3. Internet-Referenzmodell und dessen Realisierung

Version: Dez. 2003

- Referenzmodell und interne TCP/IP Protokolladressierung
- Entwicklung des Internet

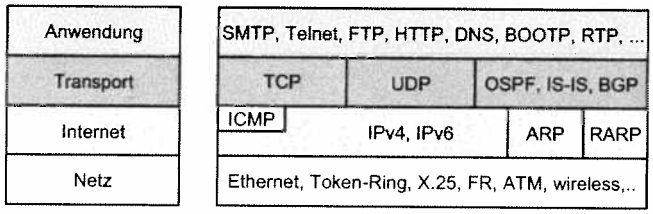
Weitere Unterkapitel

- Netzzugangsschicht: Ethernet-Vernetzung, Netztechnologien
- Internetschicht: IPv4, IPv6, Routing, MPLS, Intserv, Diffserv
- Transportschicht: TCP, UDP, Flusskontrolle
- Applikationsschicht: FTP, Email, Websurfen, HTTP

- Netzzugangsschicht: Übertragungstechnologien
- Internetschicht: IPv4, IPv6, MPLS, Routing, Intserv, Diffserv
- Transportschicht: TCP, UDP, Flusskontrolle
- Applikationsschicht: FTP, Email, Websurfen, HTTP

Komponenten der TCP/IP-Protokollfamilie

TCP/IP und das Internet sind den meisten aufgrund ihrer populären Anwendungen wie dem World Wide Web (WWW) vertraut. Das dem www zugrundeliegende Hypertext Transfer Protocol HTTP soll in diesem Kapitel zusammen mit den anderen wichtigen Standardanwendungen TELNET, FTP (Filetransfer) und dem E-Mail Protokoll SMTP besprochen werden.



TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
ICMP	Internet Control Message Protocol
ARP	Address Resolution Protocol
RARP	Reverse Address Resolution Protocol
OSPF	Open Shortest Path First
IS-IS	Intermediate System to Intermediate System
BGP	Border Gateway Protocol

SMTP	Simple Mail Transfer Protocol Protocol
FTP	File Transfer
TELNET	Terminal Emulation
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
DNS	Domain Name Service
BOOTP	Bootstrap Protocol
RTP	Real Time protocol
SNMP	Simple Network Management Protocol
FR	Frame Relay
ATM	Asynchronous Transfer Mode

Bild: Internet Referenzmodell

Das Internet- oder TCP/IP-Schichtenmodell sieht lediglich vier Kommunikationsschichten vor:

- **Netzschicht**, die mit dem jeweiligen LAN bzw. Trägernetz identifiziert wird, entsprechend den Schichten 1 und 2 im OSI-Referenzmodell.
- **Internetschicht** mit den Protokollen IPv4 sowie IPv6 und DHCP und weiteren, die funktionsidentisch ist mit der Schicht 3 im Siebenschichtenmodell
- **Transportschicht** mit der Protokollen TCP und UDP und andere als Implementierungen entsprechend den Aufgaben der OSI-Schicht 4.
- **Anwendungsschicht**, auf der die TCP/UDP-Dienste wie TELNET oder WWW zu finden sind.

Das TCP/IP-Modell verzichtet auf die Unterteilung in die Schichten 5 und 6. Die hier anfallenden Aufgaben und Funktionen sind in die Applikationen integriert. Wie daraus ersichtlich ist, besteht der TCP/IP-Protokollsatz nicht nur aus den Protokollen TCP und IP, sondern beinhaltet eine ganze Reihe weiterer Protokolle. Das TCP/IP-Kommunikationsmodell macht für ein einzelnes Protokoll nur sehr spärliche Aussagen, auf welcher expliziten Schicht es angesiedelt ist. Unter Berücksichtigung dieses Umstands wollen wir die wichtigsten Internet-Protokolle zunächst schichtenspezifisch in Kurzform vorstellen.

IPv4 und IPv6 benötigen aufgrund der unterschiedlichen Adressstrukturen schicht jeweils angepasste Versionen der Protokolle ARP/RARP, ICMP/IGMP und BGP/OSPF sowie alle Dienste, deren Aufgabe die Verwaltung von IP-Adressen und von Ressourcen (z.B. Bandbreite) auf Schicht 3 ist.

- **IP: Internet Protocol**
  Das IP-Protokoll liegt sowohl in der alten Version 4 (IPv4) als auch in der aktuellen Version 6 (IPv6) vor. Dieses Protokoll stellt einen verbindungslosen Datagrammdienst für das TCP- und das UDP-Protokoll dar. IPv4 und IPv6 sind unterschiedliche Implementierungen auf der Netzschicht und nutzen getrennte Adressräume bzw. Adressierungsverfahren.
- **ARP: Address Resolution Protocol**
  Dieses Protokoll ist ein Broadcast-Dienst zur dynamischen Ermittlung einer MAC-Adresse aufgrund einer IP-Adresse.
- **RARP: Reverse Address Resolution Protocol**
  Dieses Protokoll unterstützt ebenfalls die Adressierung und stellt das Gegenstück zu ARP dar. Es hat die Aufgabe, für eine bekannte MAC-Adresse die zugewiesene IP-Adresse zu bestimmen.
- **ICMP: Internet Control Message Protocol**
  Dieses Protokoll dient der Übertragung von Fehlermeldungen und anderen Steuerungsinformationen.
- **IGMP: Internet Group Management Protocol**

Dieses Protokoll gilt als Erweiterung von ICMP und dient vornehmlich dazu, IP-Systeme in sog. Multicast-Gruppen aufzunehmen bzw. hieraus zu entfernen.

- **OSPF: Open Shortest Path First**
  Dies ist ein Routing-Protokoll ist ein Interior Gateway Protocol (IGP), das in Autonomen Systemen (AS) eingesetzt wird.
- **BGP: Border Gateway Protocol**
  Das BGP-Protokoll stellt eine aktuelle Implementierung eines Exterior Gateway Protocols (EGP) dar und dient zum Routing zwischen Autonomen Systemen.

Auf der Transportschicht befinden sich die Protokolle TCP und UDP und einige weitere auf diese Transportschichten aufbauende Protokolle, die mit weiterführenden Transport- und Steuerungsaufgaben betraut sind.

- **TCP: Transmission Control Protocol**
  liefert eine verbindungsorientierte, zuverlässige Datenkommunikation auf der Transportschicht. TCP ist kein Datagramm Protokoll, sondern ein Bytestrom-Protokoll, d.h. relevant für die Übertragung sind keine Pakete, sondern die übermittelten Nutzdaten in Form eines Datenstroms. TCP besitzt für die höheren Protokolle eine Programmierschnittstelle.
- **T/TCP: Transaction TCP**
  eine Erweiterung und Ergänzung von TCP im Hinblick auf die Bereitstellung eines zusätzlichen schnellen Request/Response-Modus.
- **UDP: User Datagram Protocol**
  ist ein Protokoll für die verbindungslose und nicht zuverlässige Kommunikation zwischen zwei entfernten Anwenderprozessen. UDP bietet im Gegensatz zum TCP einen Datagrammdienst und gibt keine Garantie für die korrekte Übermittlung der IP-Pakete.
- **RIP: Routing Information Protocol**
  dient als internes Routing-Protokoll vornehmlich in kleineren Netzen.
- **RTP: Real Time Protocol**
  hat die Aufgabe, zeitkritische Anwendungen, wie Audio- und Videoübertragungen, über ein UDP/IP-Netz zu unterstützen. Ihm steht das Real Time Control Protocol RTCP zur Seite.
- **RSVP: Resource Reservation Protocol**
  Dem RSVP fällt die Steuerungsaufgabe zu, um Quality of Service QoS für die Übertragungsstrecke zu sichern.
- **NBoT: NetBios over TCP/IP**
  NetBIOS (Network Basic Input Output System) ist eine Programmschnittstelle (API), die häufig von Windows- und OS/2-Endsystemen genutzt wird. Das Protokoll NBoT legt fest, wie NetBIOS über TCP/IP-Netze zu übertragen ist.
- **TLS: Transport Layer Security**
  ist der offizielle Nachfolger der Secure Socket Layer (SSLv3)-Implementierung von Netscape und bietet eine Applikations-transparente Sitzungs-Authentisierung und Verschlüsselung auf Grundlage von TCP.

Anwendungsprotokolle.

- **TELNET**
  Dieses Protokoll, mit dem sich der Anwender in einer interaktiven Sitzung auf einem entfernten Computer einloggen kann, kann als Urvater der anwendungsbezogenen TCP/IP-Protokolle verstanden werden.
- **FTP: File Transfer Protocol**
  FTP dient zur Übertragung von Dateien zwischen zwei über ein TCP/IPNetz verbundene Endsysteme. Es ist bewußt einfach und robust aufgebaut, so dass die Nutzdatenübertragung auch über schlechte Verbindungen (Satellitenkommunikation) und zwischen sehr unterschiedlichen Rechnersystemen möglich ist.
- **SMTP: Simple Mail Transport Protocol**
  Die Übertragung von elektronischer Post geschieht im Internet mittels des SMTP. Heute wird in der Regel das Extended SMTP (ESMTP) eingesetzt, das eine 8-Bit-transparente Übertragung der Nachrichten ermöglicht.
- **HTTP: Hypertext Transport Protocol**
  Neben SMTP ist HTTP die wichtigste Anwendung im Internet, da es für die Übertragung zwischen Web-Browser und Web-Client sorgt. HTTP ist eine Weiterentwicklung des Network News Transport Protocol NNTP, unterscheidet sich aber von diesem inhaltlich durch die Kenntnis und Mitteilung des sog. MIME-Contents (Multipurpose Internet Mail Extension) der Nachricht.
- **Weitere auf TCP basierende Anwendungen**
  Hierzu zählen solche Anwendungen, die nicht originär als Internet-RFCs, sondern von dritter Seite entwickelt wurden. Als Beispiel hierfür kann das X-Windows-System (aktuell X.11R6) genannt werden, das einen graphischen, fensterorientierten Zugriff mit Mausunterstützung auf ein bzw. mehrere entfernte (vorwiegend UNIX-) Systeme bietet. Auch die Erweiterung NetBios über TCP/IP läßt sich hier einreihen, sowie die Übertragung von Sprache über TCP/IP (Voice over IP VoIP) oder mittels des ITU-Standards H.323 auf RTP/UDP.
- **UNIX-Kommandos**
  Im Rahmen der Entwicklung von UNIX BSD haben einige spezifische UNIX-Kommandos eine Netz-Erweiterung (Kommando remote) erfahren.

Hierzu zählen rlogin, rcp, rexec und Protokolle zur Druckeransteuerung. Aufbauend auf den SUN OS Remote Procedure Calls RPC hat sich das Network File System NFS entwickelt, das in UNIX-Netzen stark verbreitet ist. Diese Protokolle setzen sowohl auf TCP als auch bevorzugt auf UDP auf.

- **Netzanwendung**
  Zum Einrichten und zum Management großer IP-basierender Netze wurden eine ganze Reihe von Internet-Protokollen geschaffen. Das wohl wichtigste Protokoll im Internet ist das Domain Name System DNS, mit dem der Internet-Namensraum dynamisch verwaltet wird. Zur Konfiguration von IPv4-Clients hat sich das aus dem BOOTP hervorgegangene Protokoll DHCP (Dynamic Host Configuration Protocol) bewährt, die Überwachung von IP-Netzen obliegt dem Simple Network Management Protocol SNMP.
- **Weitere Dienste**
  Zu diesen lassen sich bereits erwähnte NFS zählen wie auch die verteilten Datenbankanwendungen Network Information Service NIS (früher auch Yellow Pages YP genannt) und das aktuelle Lightweight Directory Access Protocol LDAP.

Kommunikationsprinzipien der TCP/IP-Protokollfamilie

Die TCP/IP-Protokollfamilie gliedert sich lediglich in die drei Schichten Applikation, Transport und Netz. Applikations- und Transportschicht werden auch häufig gemeinsam als Hostschicht, die Netzschicht auch als Gatewayschicht bezeichnet. Eine vierte Schicht - Data-Link (DL) -rundet das TCP/IP-Schichtenmodell nach unten ab. Anwendungsdaten werden um den Applikationsheader ergänzt und als Nachricht von der Transportschicht entgegengenommen. Unter Nutzung der Dienste von TCP wird eine Nachricht in Segmente eingeteilt und um den TCP-Header bereichert.

Würde die Applikation hingegen auf dem verbindungslosen UDP aufsetzen, so würde dies in der Erzeugung eines UDP-Datagramms mit entsprechendem UDP-Header resultieren. TCP bzw. UDP reichen das Segment bzw. das Datagramm an die Netzschicht weiter, wo entweder ein IPv4- oder ein IPv6-Paket erzeugt wird. Letzteres liegt allein in der Kontrolle der sendenden Station und ist von der Applikation abgeschirmt. Dieses Verfahren wird generell als Verkapselung (engl. Encapsulation) bezeichnet. Hierbei kommen auf jeder Schicht entsprechende Dienstgrundfunktionen zum Zuge. Im Gegensatz zum OSI/ISO-Referenzmodell sind diese weniger formell als funktional definiert, worauf wir weiter unten eingehen werden.

Eine wichtige innere Eigenschaft einer Schicht ist ihr Vermögen, eine Nachricht entsprechend ihrem internen Datenpuffer bzw. den Anforderungen der unterliegenden Schicht zu fragmentieren, in kleinere Segmente aufzusplitten und mit einer Sequenznummer eindeutig zu kennzeichnen. Im Fall von TCP beträgt die maximale Größe eines Segments 64 KByte. Die TCP-Instanz im Zielrechner setzt die empfangenen Segmente mit einer eindeutigen Sequenznummer wieder zusammen. Gehen Daten bei der Übertragung verloren bzw. werden diese verfälscht, so fordert die Empfänger-TCP-Instanz die Retransmission der Daten ab dem letzten korrekt empfangenen Segment erneut an. Diese Aufgabe ist beim Einsatz von UDP auf der Transportschicht nicht vorgesehen und muss daher von der Applikationsschicht vorgenommen werden.

Eine Fragmentierung wird nicht nur von der UDP- und TCP-Instanz des Senders vorgenommen, indem die Nachricht in Segmente bzw. UDP-Datagramme aufgespalten wird, sondern auch von den im Übertragungsweg liegenden Routern bzw. Gateways auf der Netzschicht, d.h. durch die Teilung von IPv4- und IPv6-Paketen und Erzeugung von neuen IP-Headern mit entsprechender Kontrollinformation.

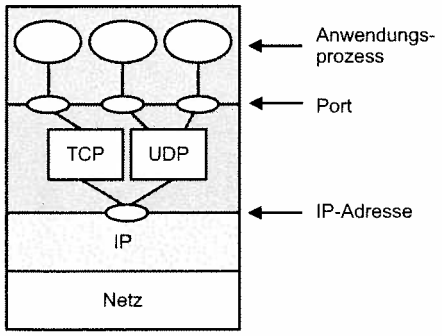


Bild: Transport-Adressen im Internet

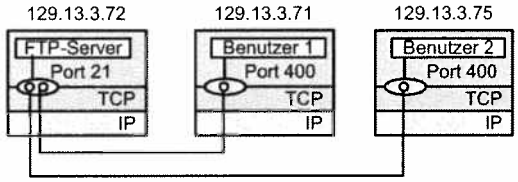


Bild: Adressierung in TCP (IP-Adresse + Port-Adresse)

Im Internet geschieht die Adressierung der Anwendungsprozesse über die Kette:

- IP-Adresse,
- Port-Nummer (Well-known Ports oder anwenderspezifizierte Ports).

Dabei kann ein Port über verschiedene Protokolle erreicht werden Dieses Protokoll ist durch eine Protokollnummer spezifiziert. Die Protokollnummer befindet sich bei IPv4 direkt im Header. Bei IPv6 ist das Protokoll im Header-Extention vorhanden. Die Portnummer ist im TCP oder UDP-Header zu finden.

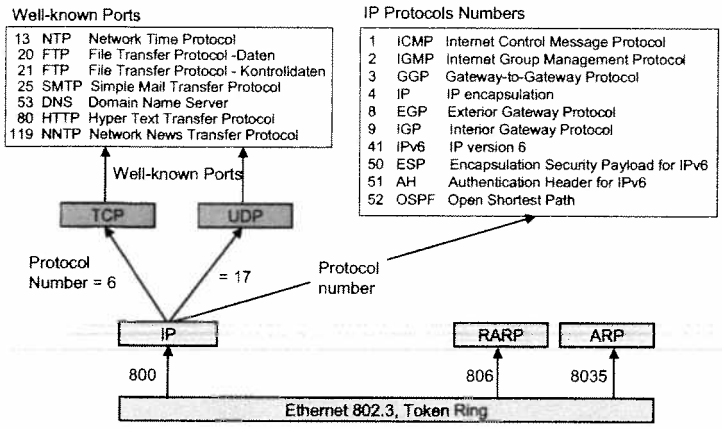


Bild: Adressierung von Internetanwendungen

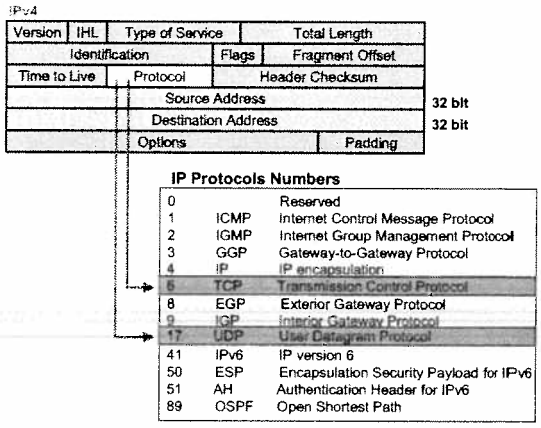


Bild: IPv4 Header und Protokollnummern

Die wichtigsten Angaben, die den zu übertragenden Daten von IP hinzugefügt werden, sind die Adresse von Quell- und Zielrechner. Für die Adressierung wird bei IPv4 eine 32 Bit lange IP-Adresse benutzt. IPv6-Adressen besitzen hingegen eine variable Länge, die sich nach den Erfordernissen der Schicht-3 Kommunikation ausrichtet.
Man unterscheidet die IP-Adressen nach

- Unicast- Adressen,
- Multicast- Adressen,
- Broadcast-Adressen.

Während für die TCP/IP-Kommunikation in der Regel Unicast-Empfänger-Adressen genutzt werden, finden Multicast- bzw. Broadcast-Adressen für bestimmte Kontrollaufgaben Verwendung. Der Kommunikationsaufbau über TCP ist ähnlich wie beim Telefonieren. Es gibt einen aktiven Partner - den Anrufer - und einen passiven - den Angerufenen. Bevor zwei Programme miteinander kommunizieren können, müssen sich die Kommunikationsendpunkte untereinander verständigen. Diese Punkte werden als Empfängerport und Sender-Port bezeichnet und müssen als Adressen bei den Protokollangaben in den einzelnen Schichten verwendet werden. Die Portnummern haben somit die gleiche Funktion inne wie die Service Access Points (SAP) im OSI-Referenzmodell. Beide Kommunikationspartner müssen daher eine Ziel-Portnummer vereinbart haben, unter der ein passiver Partner auf das Zustandekommen einer Verbindung wartet.

Die TCP/IP-Applikationen, wie z.B. SMTP oder FTP, sind feste Standardanwendungen, die unter den allgemein bekannten und weltweit eindeutigen Portnummern (in Zielrechnern!) erreichbar sind. Eine derartige Nummer wird in der TCP/IP-Welt als Well-Known Port bezeichnet.

Es ist möglich, dass zu einem bestimmten Zeitpunkt mehr als eine Anwendung die Protokolle TCP/IP oder UDP/IP benutzen kann. Um das zu realisieren, muss IP mit TCP bzw. UDP entsprechend zusammenarbeiten. Als Socket definieren wir die beiden Tupel Socket: {(IP-Adresse ,Port) am Sender), (IP-Adresse, Port) am Empfänger)}, d.h. die IP-Adresse und die Portnummer des Senders bzw. Empfängers, die den Endpunkt einer TCP-Verbindung darstellen. Sockets sind somit auf die Kommunikationspartner sowie auf die Zeitdauer der Verbindung beschränkt. Es genügt, wenn eines der vier Kriterien sich unterscheidet, z.B. die Portnummer des Senders. Jedem Socket steht im Rechner ein reservierter Speicherplatz als Kommunikationspuffer zur Verfügung. Die zu übertragenden und zu empfangenden Daten werden jeweils in dem für die Sockets reservierten Kommunikationspuffer abgelegt.

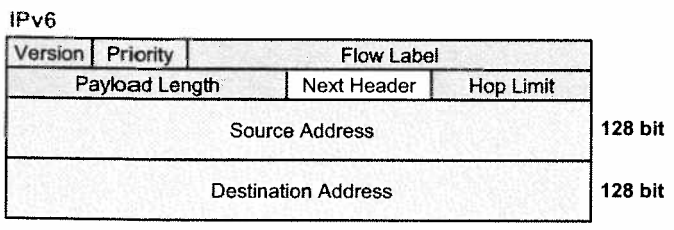


Bild: IPv6 Header

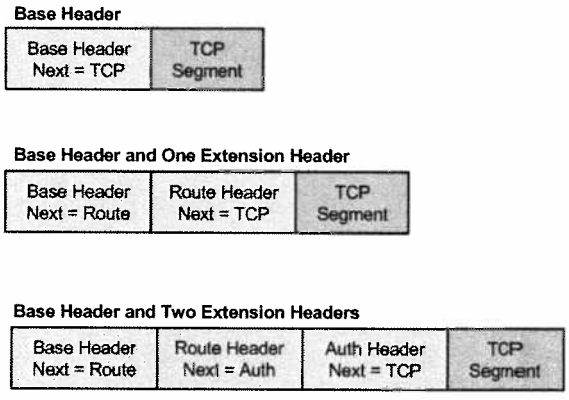


Bild: Extension Header in IPv6

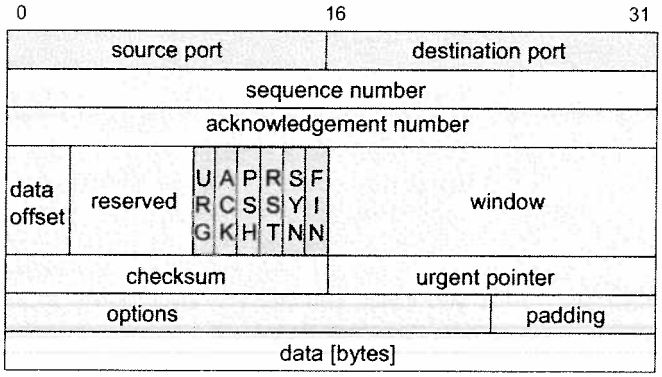


Bild: TCP Header

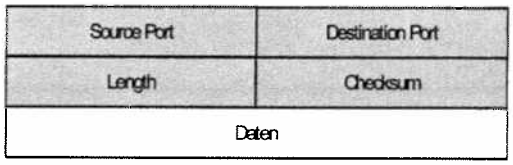


Bild: UDP-Header

20	FTP (Data), File Transfer Protocol	(TCP)
21	FTP (Control)	(TCP)
23	TELNET, Terminal Emulation	(TCP)
25	SMTP, Simple Mail Transfer Protocol	(TCP)
53	DOMAIN, Domain Name Server	(UDP)
67	BOOTPS, Bootstrap Protocol Server	(UDP)
68	BOOTPC, Bootstrap Protocol Client	(UDP)
69	TFTP, Trivial File Transfer Protocol	(UDP)
80	HTTP Hypertext Transfer Protocol (default port)	(TCP)
111	SUN RPC, Run Remote Procedure Call	(TCP)
161	SNMP, Simple Network Management Protocol	(UDP)

Bild: Well Known Ports

Eine TCP-Verbindung zwischen zwei Anwendungen ist vollduplex und setzt sich aus zwei gegengerichteten unidirektionalen Verbindungen zusammen. In bezug auf die oben eingeführten Sockets dreht sich für jede unidirektionale Verbindung die Sender- und Empfängerreihenfolge bei IP-Adresse und Portnummer um. Aufgrund der Lokalität reicht dies für einen wohldefinierten Socket aus.

Das gleiche Konzept gilt auch für UDP-Verbindungen. Da UDP und TCP unterschiedliche Kommunikationsinstanzen sind, können für diese durchaus gleiche Portnummern belegt werden. Ein Problem für das Socketkonzept bei UDP ist seine Verbindungslosigkeit. Während beim TCP der Verbindungsaufbau und auch der -abbau nach Regeln erfolgt und somit die Wiederverwendbarkeit eines Sockets von der TCP-Instanz selbst verwaltet werden kann, benötigt UDP hierzu einen Hilfsdienst, der als Portmapper bezeichnet wird. Der Portmapper kann als Buchhalterdienst verstanden werden, der einerseits auf Anforderung UDP-Ports dynamisch zuweisen kann, andererseits die lokale Freigabe eines Sockets regelt. Seine Hauptaufgabe besteht darin, dafür zu sorgen, dass der gleiche Socket nicht zur gleichen Zeit mehrfach benutzt wird.

Jedes LAN-Endsystem ist unter einer MAC-Adresse erreichbar, die als physikalische LAN-Adresse zu sehen ist. Ob ein MAC-Frame von einem LAN-Endsystem empfangen werden muss, entscheidet man nach der MAC-Adresse.

Eine der Aufgaben der LLC-Schicht in LANs besteht darin, die Multiprotokollfähigkeit zu gewährleisten. Die LLC-Schicht wird oft mittels eines LLC-Treibers implementiert. Beispielsweise wird ein LLC-Treiber unter dem Netzbetriebssystem:

- Windows von Microsoft als NDIS (Network Driver Interface System) bezeichnet.
- NetWare von Novell als ODI (Open Data Link Interface) bezeichnet.

Der LLC-Treiber entscheidet nach der Angabe SAP (Service Access Point) im LLC-Header, zu welchem Netzprotokoll die einlaufenden Daten übergeben werden sollen.

Im IP-Header gibt das Feld Protokoll Auskunft darüber, an welches Transportprotokoll (z.B. UDP, TCP) die Daten weitergegeben werden sollen. Die Portnummern von TCP und UDP dienen also dazu, die Inhalte aus den empfangenen IP-Paketen der richtigen Anwendung zu übergeben.

Das Kommunikationsmodell von UDP und TCP basiert auf einer sog. Peer-to-Peer-Kommunikation, d.h. einem gleichberechtigten Partnermodell. Hierbei gibt es im Gegensatz zum konkurrierenden Client/Server-Kommunikationsmodell keinen bevorzugten, die Kommunikation steuernden Partner.

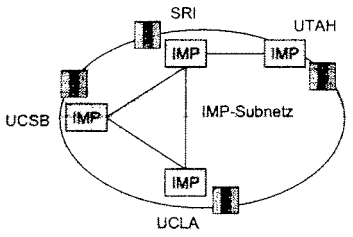
Anders verhält sich der Sachverhalt auf der Applikationsschicht. Hier fällt z.B. dem TELNET-Server die Aufgabe zu, seinen Dienst für einen Anwender, den (TELNET-)Client, bereitzustellen. Client und Server können sich sowohl auf dem gleichen System als auch auf Rechnern befinden, die über ein TCP(UDP)/IP-Netz verbunden sind. Im Gegensatz zu einigen anderen Protokollen, wie beispielsweise NetBIOS, nimmt der Server kein sog. Advertising seines Dienstes vor. Vielmehr sind die Serverdienste explizit durch zugeordnete TCP-Ports allgemein bekannt (Well-Known Port) bzw. werden bei UDP durch den Portmapperdienst, dem selbst ein fester UDP-Port zugewiesen ist, verwaltet.

1961: Leonard Kleinrock's Theorie zeigt die Effektivität von Paketvermittlung für burstartigen Verkehr. Nutzung der Warteschlangentheorie.

1964: Baran untersucht die Sprachübertragung in Paketvermittlung in militärischen Netzen.

1967: Plan zur Erstellung des ersten paketvermittelten Netzes, ARPANET, wird veröffentlicht.

1969: Die ersten Knoten von ARPANET sind funktionsfähig: Erstes Internet.



University of California at Los Angeles (UCLA)
University of California at Santa Barbara (UCSB)
Stanford Research Institute (SRI),
University of Utah

ARPA: Advanced Research Projects Agengy
IMP: Interface Message Processor

Bild: Entwicklung des Internet (60er Jahre)

Die Geschichte des Internet

Aufgrund von theoretischen Arbeiten von Leonard Kleinrock in UCLA (University of California at Los Angeles) begann in 1964 Paul Baran von der Rand Corporation sich in ersten theoretischen Arbeiten mit dem Konzept einer Paket-Vermittlung für Sprache auseinanderzusetzen Es konnte gezeigt werden, dass die integrierte Paketvermittlung von Daten und Sprache effizienter war, als die dauernde Bereitstellung eines reservierten Kanals, wie bei der Leitungsvermittlung. Im Jahre 1967 dann wurde in Amerika eine Forschungsaktivität der Defence Advanced Research Agency (DARPA) gestartet, die 1966 dann zum Plan für ein Computernetz reiften, dem ARPANET. Dabei war wichtig, dass auch bei Ausfall eines Teiles des Netzes der Rest weiterarbeiten kann.

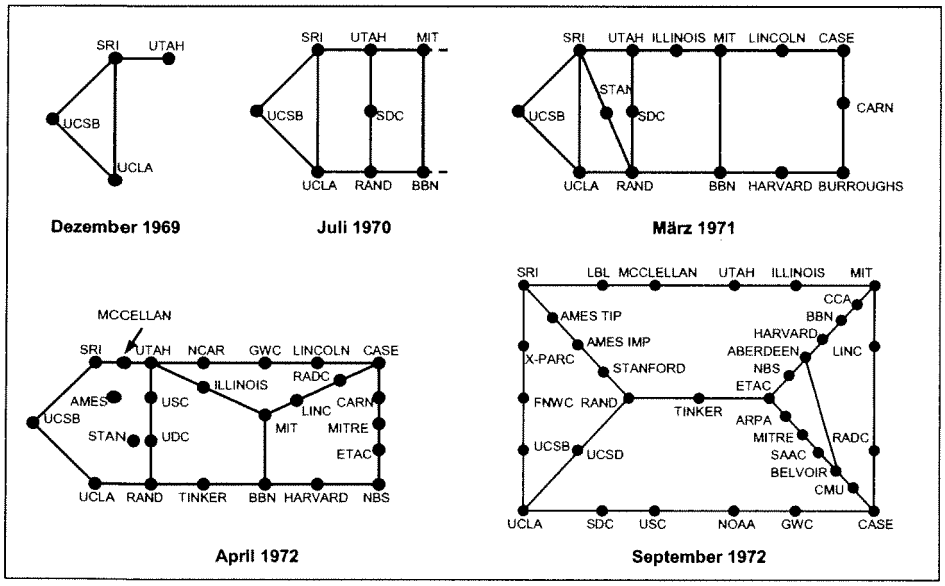


Bild: ARPANET (1969-1972)

Damit waren sowohl eine zentrale Instanz als auch eine Hierarchie von Knoten ausgeschlossen-, die Intelligenz musste verteilt werden. Eines der Schlüssel-Elemente in dieser Architektur war der Interface Message Processor (IMP), an dem Robert Kahn von BBN arbeitete. Das erste wirkliche Netz wurde 1969 zusammengeschaltet: je ein Rechner der Universitäten von Los Angeles, Santa Barbara, Utah und des Stanford Research Institute wurden verbunden.

1971: Betriebsaufnahme ARPANET (erstes Internet-Backbone);
Experiment zum Einloggen auf entfernten Rechnern

1972: Erste öffentliche Demonstration von ARPANET mit ca. 15 Knoten.
RFC001: NCP (Network Control Protocol): erstes Protokoll zwischen zwei Internet-Hosts
Erstes Email-Programm von Ray Tomlinson (BBN, Bold, Beranek and Newman)

1973: Metcalfe entwickelt Grundprinzipien des Ethernet

1974: Cerf und Kahn entwerfen Architektur zum Koppeln von Netzen
- Entwurf von TCP (am Anfang noch mit integriertem IP)
- Ende der 70er Jahre Basisprotokolle des Internets (IP, UDP und TCP) vorhanden
- Internet-Entwurfsprinzipien (die auch heute noch gültig ist !)

Entwicklung einer Reihe proprietärer Netze
- DEC: DECnet, Xerox: XNS-Architektur, IBM: SNA-Architektur
Ende der 70er Jahre erste Ansätze zur Paketvermittlung mit Dateneinheiten fester Länge (Vorläufer von ATM)

1979: ARPANET hat 200 Knoten

Bild: Entwicklung des Internet (70er Jahre)

Schon in diesen frühen Tagen dessen, was wir heute Internet nennen, wurde parallel an der Weiterentwicklung der Technologie (Netzarchitektur und Protokolle) sowie an den Anwendungen gearbeitet. Die Anwendung des Netzes bestand also in der Verbindung von Großrechnern, um gegenseitig Rechnerkapazität auszutauschen. Die erste auch für den menschlichen Benutzer wichtige Anwendung wurde 1972 eingeführt: der Austausch von Nachrichten oder kurz: E-Mail. Ebenfalls im Jahre 1972 fand auf der International Computer Communication Conference (ICCC) die erste öffentliche Demonstration des Netzes statt.

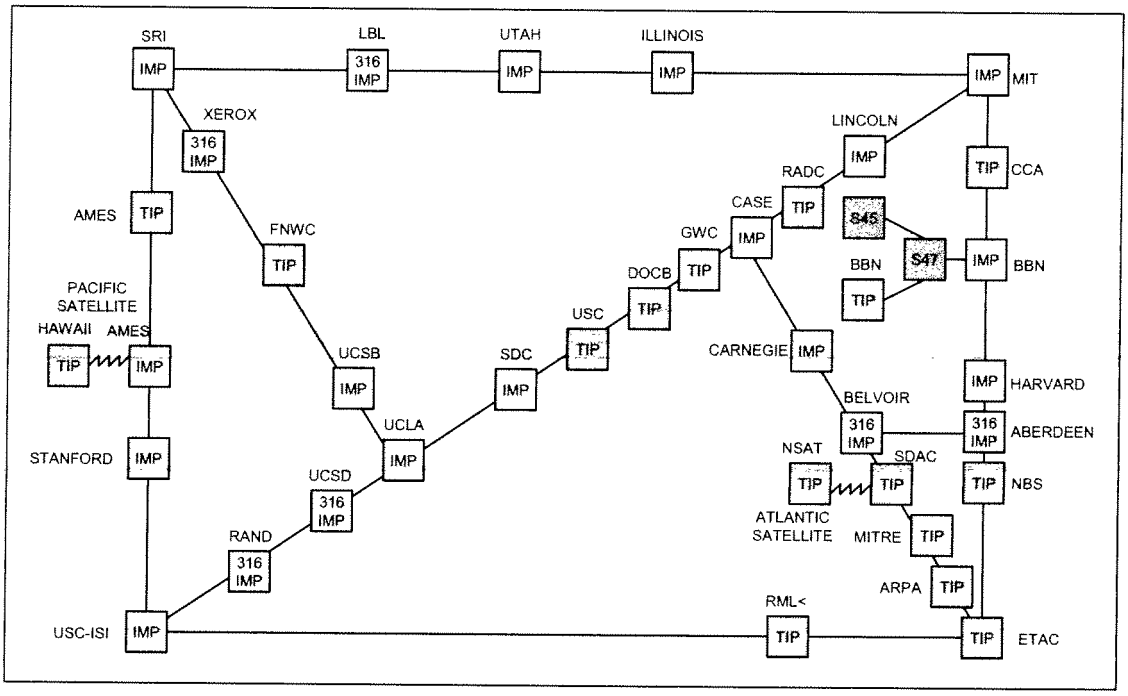


Bild: ARPANET (August 1973)

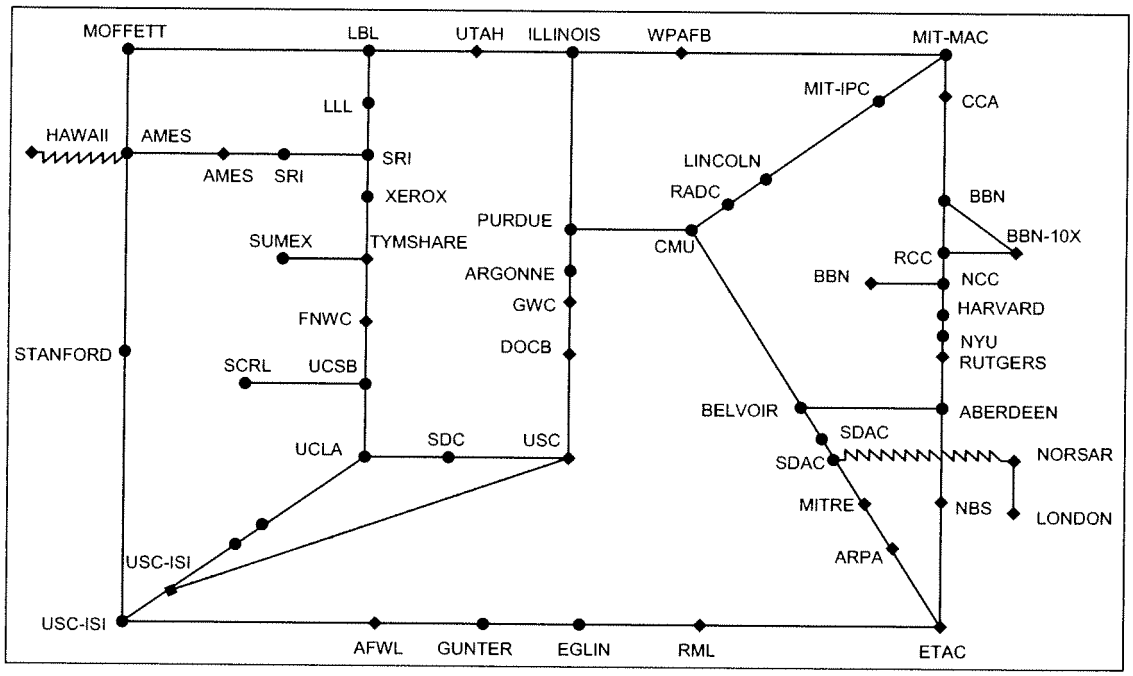
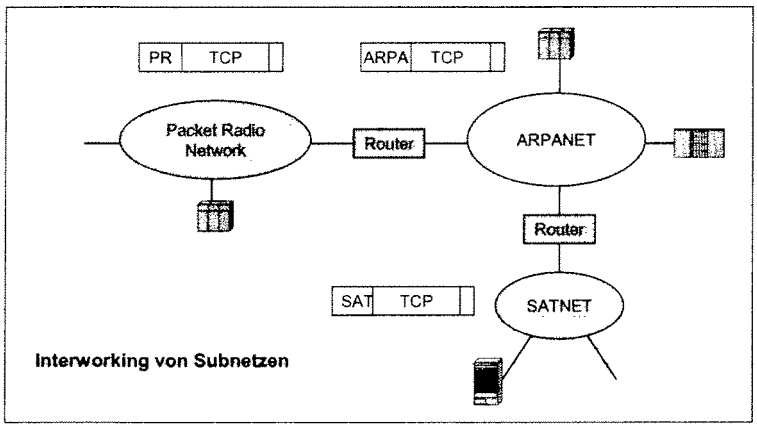


Bild: ARPANET (Juni 1975)

Dieses ARPANET kann als ein homogenes Netz angesehen werden, aber die Gründungsväter hatten schon die Weitsicht zu erkennen, dass es in der Zukunft eine Vielzahl unterschiedlicher Netze in unterschiedlichen Technologien geben wird. Das Ziel war damit, diese unterschiedlichen Netze zusammenzuschalten, also einen kleinsten gemeinsamen Nenner zu finden, um dieses zu bewerkstelligen, ohne in die innere Architektur dieser einzelnen Netze eingreifen zu müssen. Die im ARPANET eingesetzten Protokolle konnten nicht alle Anforderungen erfüllen, so dass Robert Kahn ein neues Protokoll definierte, genannt TCP/IP für Transmission Control Protocol/Internet Protocol. Da es wichtig war, mit den Betriebssystemen zusammenzuarbeiten, wurde der Experte Vinton Cerf aus Stanford hinzugezogen. TCP/IP arbeitete mit Fehlersicherung, also Wiederholungsanforderung bei Paketverlust. Es zeigte sich, dass es Applikationen gibt, die dieses nicht erfordern, sondern mit einem einfachen Datagram-Dienst auskommen. Kahn und Cerf spalteten daher TCP/IP auf und entwickelten einen Satz von drei Protokollen: IP, TCP und für den Datagramm-Dienst das User Datagram Protocol (UDP). Diese drei Protokolle werden bis heute fast unverändert verwendet.



SATNet : Satellitenverbindung mit Europa

Bild: Internet (1977)

Im Jahre 1972 waren es schon 40 Rechner, die über das Netz verbunden waren. 1973 kam eine andere Erfindung dazu, die die Daten-Welt revolutionierte. Bob Metcalfe entwickelte am Ende seines Studiums am MIT einen verteilten Zugriffsmechanismus auf ein gemeinsames Medium (... ein Kabel). Bei XEROX-PARC wurde das Verfahren in die Praxis umgesetzt und jeder kennt es heute: das Ethernet.

Allerdings führte man auch ein Namen-System ein, das die IP-Adressen dem Benutzer verdeckt. Mit dem Wachstum des Netzes benötigte man auch eine leistungsfähige Umwertung - der Domain Name Service war geboren, übrigens auch ein heikler Punkt im heutigen Internet, weniger aus technischen Gesichtspunkten als mehr aus rechtlichen (wem gehört eine Domain?).

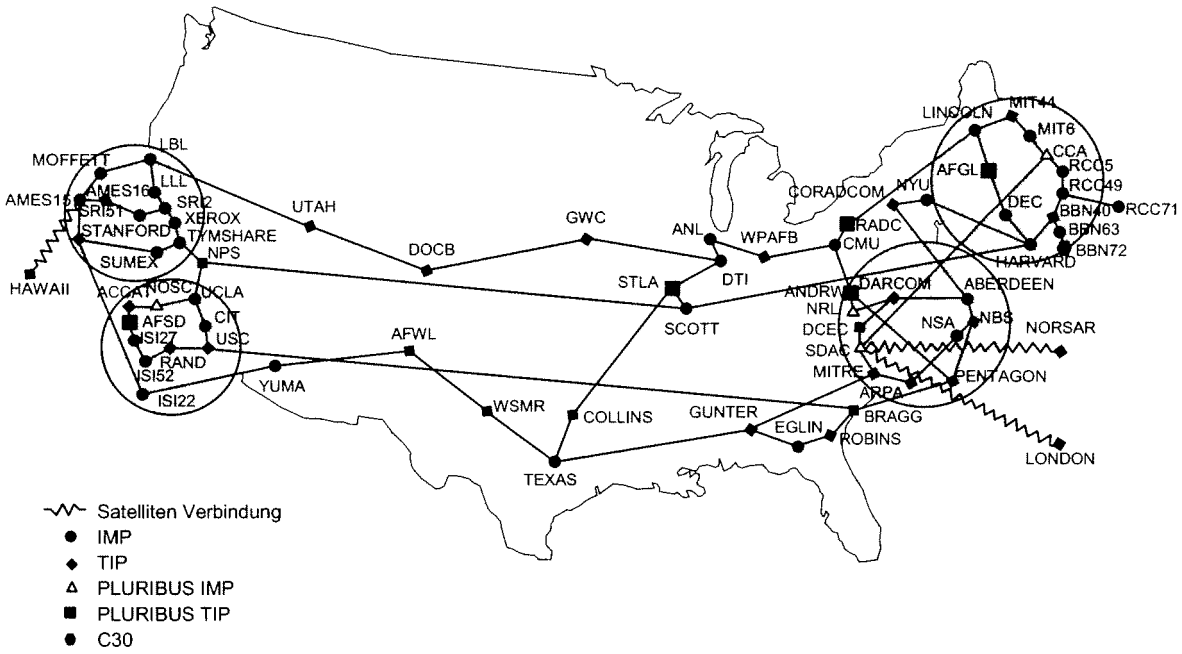


Bild: ARPANET (Oktober 1980)

Das Wachstum erforderte auch eine Strukturierung des Netzes. Waren vorher alle Knoten (Router) gleich, wurden das Netz jetzt in Netz-Bereiche aufgebrochen mit unterschiedlichen Routing-Prozeduren innerhalb und zwischen solchen Netz-Bereichen.

Am 1. Januar 1983 wurde das ARPANET vom alten Protokoll NCP auf TCP/IP umgestellt. Dazu wurde an diesem einen Tag in allen Rechnern und Routern die Netz-Software umgeschaltet. Das ging, da sich alle Betreiber noch kannten. Heute erfordert eine Umstellung auf eine neue Version (IPv6) vielfältige Kapselungs- und Interworking-Spezifikationen, denn es werden auf lange Zeit beide Versionen im Einsatz bleiben.

Die Umstellung erlaubte auch eine andere Veränderung: nachdem das ARPANET von vielen Organisationen, vom Verteidigungsministerium bis zu Forschungseinrichtungen, benutzt wurde, wollte der Verteidigungsbereich wieder ein eigenes Netz haben. Daher wurde der militärische Teil unter dem Namen MILNET abgespalten, das ARPANET blieb der Forschung.

1980 – 1990:
- Aufbau verschiedener Netze, neue Protokolle
- Neue Netze, meist zur Verbindung von Universitäten
- BITnet (Because it's their network)
- CSnet (Computer Science network; kein Anschluss an ARPANET)
- NSFNET (1986, anfangs mit 56 kbit/s im Backbone)

1982: SMTP-Protokoll für E-Mail definiert

1983: Alle Knoten im ARPANET mussten von NCP auf TCP/IP wechseln

1983: DNS zur Übersetzung von Namen auf IP-Adressen definiert

1985: FTP-Protokoll definiert

1988: Das Internet umfasst auch Netze in Europa, Australien und Kanada.

Ende der 80er Jahre: ca. 100.000 Knoten im Internet

Bild: Entwicklung des Internet (80er Jahre)

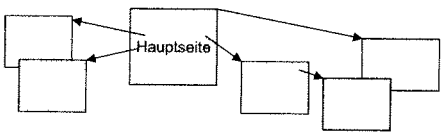
Damit veränderte sich die komplette Datenkommunikations-Struktur: das ursprüngliche Modell ging von wenigen Großrechnern aus, die zu verbinden waren; jetzt wurden daraus eine Vielzahl kleiner Netze mit Workstations und Personal Computers, letzterer begann schließlich 1980 seinen Siegeszug um die Welt. Allerdings musste erst eine TCP/IP-Implementierung für diese kleinen Maschinen entwickelt werden, etwas was viele als illusorisch angesehen hatten - aber es ging; und diese kleinen TCP/IP-Stacks konnten voll mit denen der Großrechner mithalten.
Auf diese Veränderung wurde auch in anderer Beziehung schnell reagiert und man schuf die Adressklassen für kleine, mittlere und große Netze, ohne allerdings damals zu ahnen, welche Probleme uns heute Ende der 90er Jahre daraus erwachsen sind.

Dann begann eine Zeit rascherer Veränderungen. In vielen Ländern wurden staatliche Programme aufgelegt mit dem Ziel der Kommunikation zwischen den Universitäten. In den USA war es die National Science Foundation (NSF), die im Jahre 1986 aus Steuermitteln das NSFNET aufbaute, beginnend mit fünf vernetzten Großrechnern in USA. Das Internet Advisory Board (IAB, heute: Internet Architecture Board) nahm seine Arbeit auf und das Federal Networking Council (FNC) übernahm als Regierungsstelle die Aufsicht über die Finanzierung des Netzes und die Kontakte zu entsprechenden Netzen in anderen Ländern. Das Jahr 1995 kann als das Ende des staatlich geförderten Internet angesehen werden - das NSFNET wurde aufgelöst und kommerzielle Netzbetreiber haben die Rolle der Internet-Backbones übernommen.

Erst 1987 wurde ein Management-Protokoll spezifiziert, mit dem es möglich war, Netzelemente (hauptsächlich Router) aus der Ferne zu bedienen und warten, das Simple Network Management Protocol (SNMP).

Seit Beginn ging man von freundlichen Nutzern aus und hat das Stichwort Netiquette geprägt. Darunter werden Regeln verstanden für das richtige Verhalten im Netz. Das hat auch gut funktioniert, solange die Nutzer einer mehr oder weniger homogenen Gruppe angehörten (Forscher, Studenten, ...). Mit der Öffnung für jedermann und der Kommerzialisierung des Netzes tauchten aber auch viele schwarze Schafe auf, die sowohl andere Nutzer als auch das Netz selbst als Angriffsziel haben. Daher sind inzwischen auch die Rechtsanwälte mit Netzfragen beschäftigt, einige haben sich sogar auf dieses Themengebiet spezialisiert.

Noch im Jahre 1994 konnte man den Streit verfolgen, der zwischen denen entbrannt ist, die das Internet als Forschungsnetz weiterbetreiben wollten, und denen, die einer Kommerzialisierung positiv gegenüber standen. So haben die Gegner der Öffnung eine amerikanische Anwaltskanzlei mit sogenannten Mail-Bomben belegt, die über News-Gruppen Werbung betrieben haben.



1989 : Tim Bernes-Lee (Cern, Geneva):
Vorschlag für die globale Vernetzung von Dokumenten (Hypertext)
1991 : Erste Vorführung
1993 : Mosaic Browser
1995 : Netscape Browser
Heute: Internet Explorer

www für viele äquivalent zu Internet

Bild: WWW – World Wide Web

1990 – 2000: Kommerzialisierung und das Web

1991: - NSFNET lockert Restriktionen für kommerzielle Nutzung
- EBONE: Europäisches Backbone
- WWW Vorführung am europäischen Kernforschungszentrum CERN

1992: Ca. 200 Web-Server

1995: - Kommerzielle ISPs transportieren NSFNET-Verkehr
- Kommerzialisierung des Web beginnt

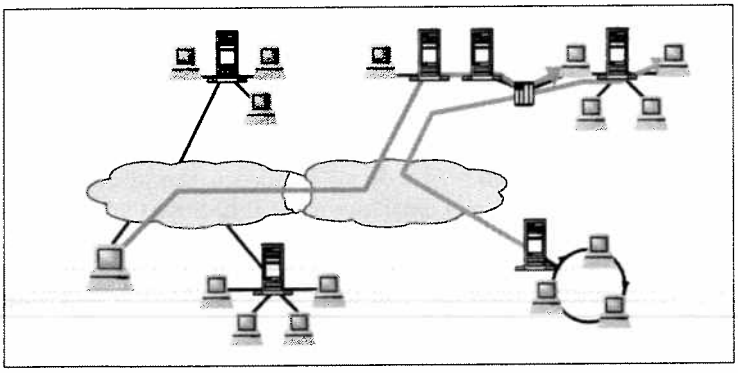
1996: University Corporation for Advanced Internet Development – Internet2

1999: Zweites Internet2-Backbone: Abilene

Ende der 90er Jahre
- Ca. 50 Millionen Computer im Internet
- Ca. 100 Millionen Nutzer
- Im Backbone: Gbit/s auf Übertragungsabschnitten

Bild: Entwicklung des Internet (90er Jahre)

RFC 1883, RFC 1884, RFC 1886, RFC 1970, RFC 1980, RFC 1981, RFC 2014,....



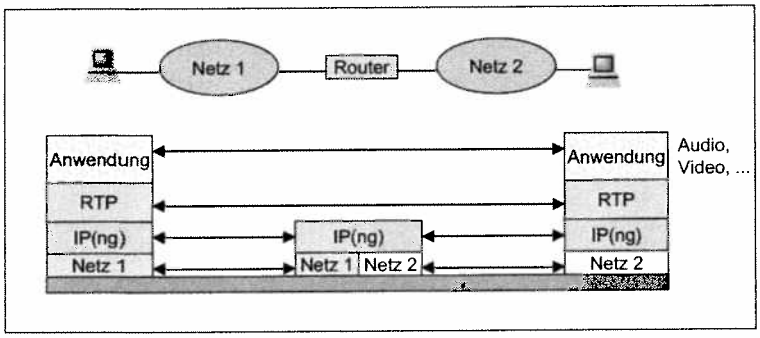
- Erweiterter Adressbereich
- Flussidentifizierung (Flow labels)
- Globale Reservierung von Ressourcen

Bild: IP Next Generation (IPng, IPv6)

Internet Benutzer (Oktober 2001)	
Kanada & USA	177.78 Millionen
Europa	113.97 Millionen
Asien/Pazifik	104.88 Millionen
Latin Amerika	16.45 Millionen
Afrika	3.11 Millionen
Mittelost	2.40 Millionen
Welt total	418.59 Millionen

Internet Hosts (Oktober 2001): 109.574429 Millionen

Bild: Internet-Statistik (Oktober 2001)

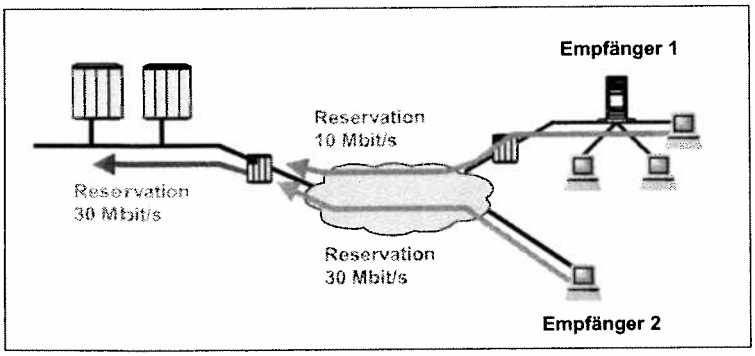


- Echtzeitanwendungen, Multimedia-Kommunikation
- Synchronisation (time stamp)
- Quality-of-Service Kontrolle (RTCP, Real Time Control Protocol)

Bild: Real Time Protocol (RTP)

Heute wird das traditionelle Internet mit sogenannten Best-Effort-Services umgestaltet, um durch benutzer-definierte Qualitätskriterien (QoS) zu erreichen.

RFC 2205, RFC 2206, RFC 2207, RFC 2210, ...



- Ressourcen Reservierung
- Empfängerorientiert
- Multicast

Bild: Resource Reservation Protocol (RSVP)

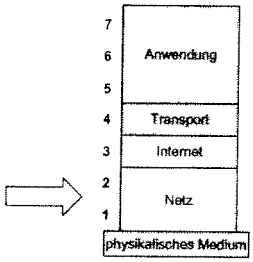
3.1a Internet-Referenzmodell: Netzzugangsschicht

Version Dez..2003

Inhalt

Ethernet Technologien
- Netzstrukturen
- Ethernet
- Fast Ethernet
- Gigabit Ethernet

Andere Netztechnologien
- Leitungsvermittlung: ISDN, GSM
- Paketvermittlung: X.25, FR, GPRS
- Zellenvermittlung: ATM
- Lokale Netze: TR, FDDI, WLAN 802.11



Internet Referenzmodell

Bild: Übersicht

- Geschichte des Ethernet
- Ethernet Topologien und Strukturen
- Der Ethernet-Standard
- Die Bitübertragungsschicht
- MAC-Subschicht
- LLC-Subschicht
- SNAP
- Unterschiedliche Bezeichnungen
- Auto-Negotiation Funktion
- Link Aggregation
- VLAN
- Fast-Ethernet
- Gigabit-Ethernet
- 10 Gigabit-Ethernet

Bild: Inhalt

Lokale Netze (LAN)

Die Auswahl bestehender Netztechnologien im Bereich Local Area Networks (LANs) erscheint unübersichtlich und mannigfaltig. Zusätzliche Evolutionsschritte erschweren zudem noch die Entscheidung zwischen Hochgeschwindigkeitsnetzen, weil neben Ethernet auch Token Ring inzwischen 100- bzw. 1000-Mbit/s Bit-Raten ermöglichen. Daneben existieren noch andere Möglichkeiten wie FDDI, der Asynchrone Transfer Modus (ATM) und Fibre Channel (FC). Allen Technologien ist gemein, dass man sie nicht uneingeschränkt für jede Anwendung oder jedes Szenario einsetzen kann. In Zukunft kommen neue Arbeitsgebiete und Applikationen zusätzlich auf Netzanforderungen hinzu, die neue Maßstäbe an die Netzkonzeption stellen:

- Zentralisierte Programm- und Datenhaltung auf File-Servern,
- Zunehmendes Remote Booting von Arbeitsstationen,
- Speicher- und Ressourcen-intensive Windows-Applikationen,
- Optische Archivierung,
- Workflow-Management im Intranet,
- Multimedia-Anwendungen (Echtzeitdatenverkehr, isochrone Datenströme).

Heute werden verstärkt Router zur Segmentierung der LANs eingesetzt. Die Strukturierung mit Routern bringt jedoch eine Reihe von Nachteilen mit sich. Durch die zusätzliche Verarbeitung der Datenpakte wird das Antwortzeitverhalten negativ beeinflusst und durch unnötig hohe Strukturtiefen im Datennetz entsteht ein nicht unerheblicher Administrationsaufwand. Die Flexibilität bei der Umkonfiguration der Netze wird dabei zudem im erheblichen Maße eingeschränkt.

Während aber die Hersteller sich durch die Bildung von Allianzen einen größeren Marktrückhalt sowie Technologiefortschritt erhoffen, um letztendlich ihr Produkt erfolgreich auf dem Markt positionieren zu können, sollte jeder, der ein Netz plant und konzipiert, andere Maßstäbe ansetzen. Dieser Abschnitt geht auf heutige Hochgeschwindigkeitsnetze ein, vergleicht sie und stellt die Vor- und Nachteile explizit heraus. Der Schwerpunkt liegt dabei aber auf den schnellen Netzen. Ethernet besitzt dabei in Datennetzen, aufgrund seiner Weiterentwicklung und dem Verbreitungsgrad, die größte Bedeutung. Letztendlich sind die eingesetzten Applikationen, Zukunftssicherheit, Skalierbarkeit, Migrationsmöglichkeiten und somit die Kosten für die richtige Auswahl entscheidend. Steigende Benutzerzahlen, Verlagerung der Rechenleistung in das Netz und Server-Engpässe sind dabei die entscheidenden Faktoren für die erhöhten Anforderungen an die Datenrate.

- 1970 wird an der Universität Hawaii das ALOHA-Netz entwickelt und erprobt.
- 1972 wurde die Idee vom XEROX Palo Alto Research Center aufgegriffen. Das Projektziel lautete: experimentales Ethernet (hier wurde der Name erfunden).
- 1979 Das vom Robert Metcalfe entwickelte CSMA/CD Zugriffsverfahren wurde für die Xerox Corporation patentiert
- 1979 Die Firmen DIGITAL, INTEL und Xerox schließen sich zusammen zur Firmengruppe DIX und führen Ethernet zur Produktreife.
- 1982 Veröffentlichung der Ethernet-Version 2.0.
- 1985 weltweite Anerkennung des Ethernet-Standards als ISO/DIS 8802/3 und ANSI/IEEE Standard 802.3.

- 1986 Veröffentlichung des 10Base2- und 10BroadT-Standards.
- 1987 Standardisierung der 10BaseT-Spezifikation.
- 1991 Veröffentlichung des 10BaseF-Standards.
- 1992 Interop San Francisco: Ankündigung von Hewlett-Packard und AT&T Microsystems Fast Ethernet 100Mbit/s auf Twisted Pair.
- 1994 über 10.000 Hersteller unterstützen global das Ethernet-Verfahren.
- 1995 Normung des 100-Mbit/s Ethernet: Standard 802.3u.
- 1998 Normung des Gigabit-Ethernet: Standards 802.3ab und 802.3z.
- 2002 Normung des 10-Gigabit-Ethernet: Standard 802.3ae.

Bild: Geschichte des Ethernet

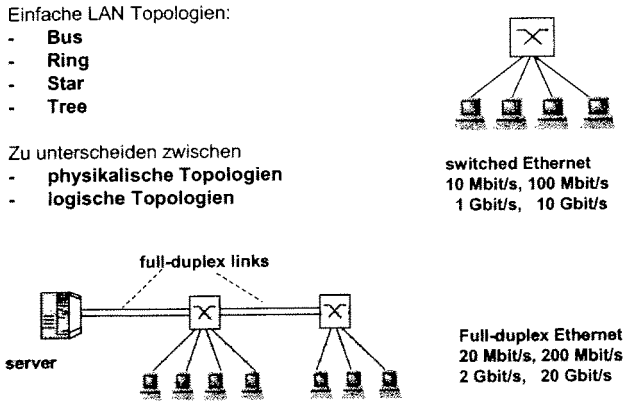


Bild: LAN Topologien

Moderne Ethernet-Vernetzungen

Fast-Ethernet mit Twisted-Pair-Kabeln hat sich mittlerweile selbst für kleinste Heimnetze durchgesetzt. In größeren Netzen ist Gigabit-Ethernet auf Glasfaser oder ebenfalls Twisted-Pair schon zum Standard. Beim Twisted-Pair-Kabel (TP) handelt es sich um ein telefonkabelähnliches Medium, das Segmentlängen bis zu 100 Metern erlaubt. Über Glasfaserkabel findet hingegen optische Datenübertragung statt, die auch größere Distanzen überbrückt. Vernetzt man mit diesen Medien mehr als zwei Stationen, dann erfolgt die Verkabelung nicht mehr wie beim Koaxialkabel kettenartig von Rechner zu Rechner, sondern sternförmig, wobei jede Station an einem aktiven Verteiler hängt. Anfangs waren das Hubs oder Repeater, die vom Datenfluss her nichts anderes als einen Ersatz für den Koaxial-Bus darstellen.

Bei Hubs oder Repeater müssen alle angeschlossenen Geräte halbduplex arbeiten: Es darf immer nur eine Station senden, während alle anderen Stationen zuhören. Grob betrachtet teilt sich dadurch die Bandbreite auf die angeschlossenen Stationen auf. Heute verwendet man Switches. Denn was für den privaten Bereich vor kurzem noch eine sinnvolle Lösung darstellte, ist im professionellen Bereich, wo es darum geht, mehrere bis Hunderte Arbeitsstationen zu vernetzen, längst nicht mehr umsetzbar. Switches stellen ebenfalls Sternverteiler dar, die im Vollduplexbetrieb Punkt-zu-Punkt-Verbindungen zwischen den angeschlossenen Geräten herstellen. Dabei steht aber im Idealfall jeder Station die volle Bandbreite des Mediums (10, 100 oder 1000 Mbit/s) zur Verfügung. Außerdem können Switches je nach Ausführung unterschiedliche Datenraten zwischen den verschiedenen Anschlüssen ausgleichen. Bei einem Ethernet-Netz, das ausschließlich Switches enthält, ist die herkömmliche Betrachtungsweise bezüglich Shared-Medium nicht mehr gültig, weil hierbei ausschließlich vollduplex gearbeitet wird. Kollisionen können gar nicht entstehen.

Schnelles Weiterleiten von Ethernet-Rahmen

Während ein Hub auf der ersten Schicht (Physical Layer) des OSI-Modells arbeitet, so agiert ein Switch funktionell auf der zweiten (Layer-2-Switch, MAC-Layer) und eventuell zusätzlich auf einer der höheren Schichten (Layer-3 ... n-Switch, IP-/TCP-Switching). Seine grundsätzliche Aufgabe ist es, Dateneinheiten (Ethernet-Rahmen) über interne Verbindungen von einem Port (Anschluss) zu einem oder mehreren anderen Ports weiterzuleiten. Dabei können auch mehrere virtuelle Verbindungen parallel laufen.

Eine gewöhnliche Bridge besitzt zwei Ports, über die sie zwei Segmente miteinander verbinden kann. Sie arbeitet im Unterschied zum Hub nicht transparent. Sie leitet also nicht alle Daten weiter, sondern entscheidet anhand der Zieladresse, ob ein Rahmen durchkommt. Damit kann eine Bridge für eine Lasttrennung zwischen den Segmenten eines Netzes sorgen. Multicast- und Broadcast-Rahmen, die an mehrere oder alle Stationen gehen, werden dabei auf jeden Fall weitergeleitet. Über eine Bridge können sich Kollisionen nicht ausbreiten. Kollisionsdomänen werden somit in kleinere Einheiten unterteilt.

Dieses grundsätzliche Funktionsprinzip hat man für Switches übernommen, sie sind quasi eine Multiport-Bridge.

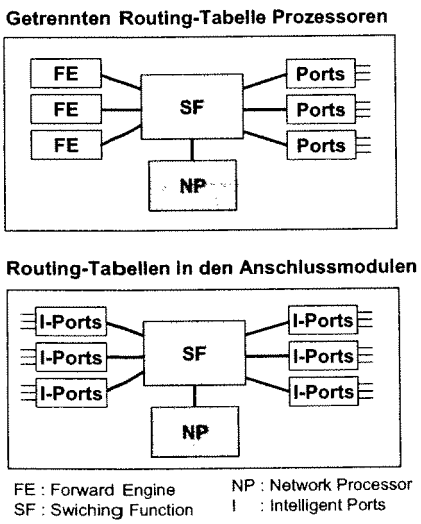


Bild: Aufbau von Ethernet-Switches

Wegelenkung (Wegfindung): MAC-Adressen

Layer-2-Switches treffen ihre Wegwahl anhand der MAC-Adressen, sie arbeiten demzufolge protokollunabhängig. Die Weiterleitung von Unicast-Rahmen (Rahmen für genau eine Station) erfolgt größtenteils zielgerichtet. Empfängt ein Switch ein Unicast-Rahmen, vergleicht er dessen Zieladresse mit den Einträgen in seiner Forwarding-Tabelle.

Dort sind alle bisher gelernten MAC-Adressen mit ihrem zugehörigen Ausgangsport gespeichert. Findet der Switch die Zieladresse, kann er den Rahmen direkt am angegebenen Anschluss ausgeben. Andernfalls leitet er den Rahmen an alle Ausgänge, ausgenommen den empfangenden Port, weiter. Dabei kommt momentan die gleiche Netzlast wie bei einem Hub auf. Die Station, an die der Rahmen gerichtet war, schickt über kurz oder lang selbst ein Rahmen ab, wobei der Switch aus der Quelladresse des Rahmens eine neue Zieladresse für die Forwarding-Tabelle lernt. Nach und nach kann der Switch immer mehr Rahmen zielgerichtet weiterleiten und so das Netz entlasten. Das Ausgeben eines Rahmens mit bislang unbekannter Zieladresse ist aber trotz gleicher Netzlast kein Broadcast. Letzterer hat ein anderes Adressformat, woran der Switch erkennt, dass er Broadcast-Rahmen generell an alle Ports weiterreichen muss.

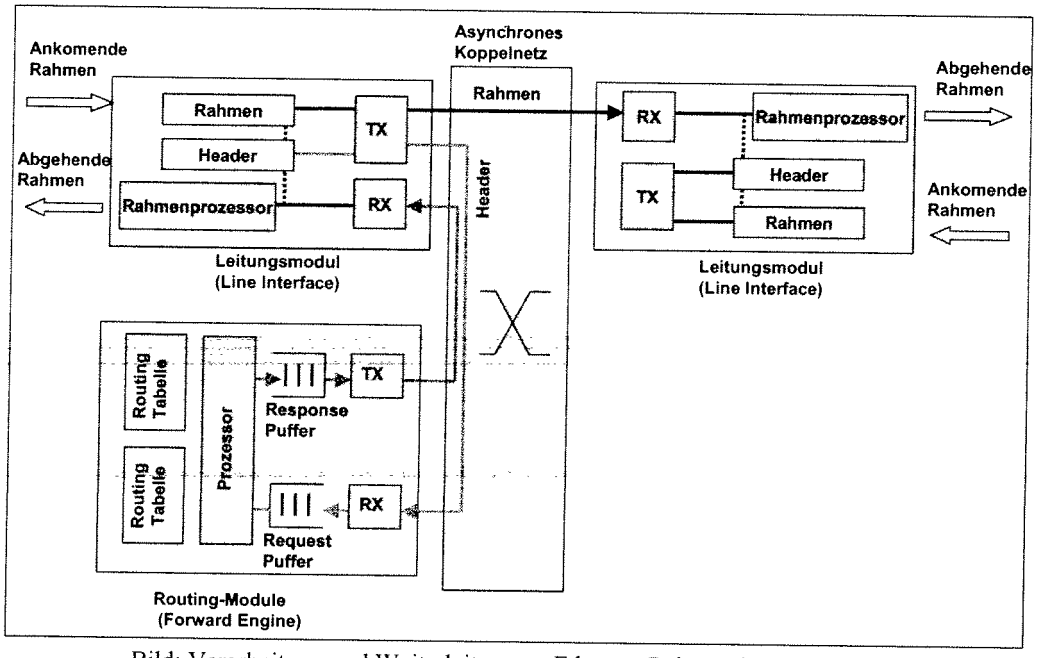


Bild: Verarbeitung und Weiterleiten von Ethernet-Rahmen in L3-Switches

Rahmen mit bereits bekannten MAC-Adressen im Cache-Tabellen des Eingangsport-Moduls werden sofort zu den betreffenden Ausgangsports weitergeleitet.

Sonst muss den Ausgangsport zuerst in einem zentralen Tabellen-Modul ermittelt werden. Meistens wird eine MAC-Tabelle oder eine IP-Tabelle konsultiert.

Die Routing-Tabellen werden durch austausch von Routinginformation aktualisiert.

Bei L2-Switches werden nur MAC-Adressen verwendet. Bei L3-Switches wird die IP-Adresse im Ethernet-Rahmen herangezogen. Bei L4-Switches werden Parameter des Transportprotokolls (TCP, UDP, ICMP) benötigt und bei L7-Switches sind auch die Parameter des betreffenden Anwendungsprogrammes wichtig.
Damit die Forwarding-Tabelle nicht überläuft, unterliegen ihre Daten einem Alterungsprozess. Kommt ein Eintrag eine gewisse Zeit (typisch 300 Sekunden) nicht mehr zum Zug, wird er wieder gelöscht. Das soll sicherstellen, dass beispielsweise abgeschaltete LAN-Stationen automatisch aus der Forwarding-Tabelle entfernt werden,

Wegen der internen Wegwahl leitet ein Switch die Datenrahmen grundsätzlich mit einer gewissen Verzögerung (Latenz, englisch Latency) weiter. Drei Faktoren bestimmen primär den Gesamtdurchsatz: Zum einen die Switch-Architektur, zweitens die interne Bandbreite, die bestimmt, wie viele Verbindungen parallel geschaltet werden können und schließlich die Pufferung von Rahmen, die nicht sofort weitergeleitet werden können.

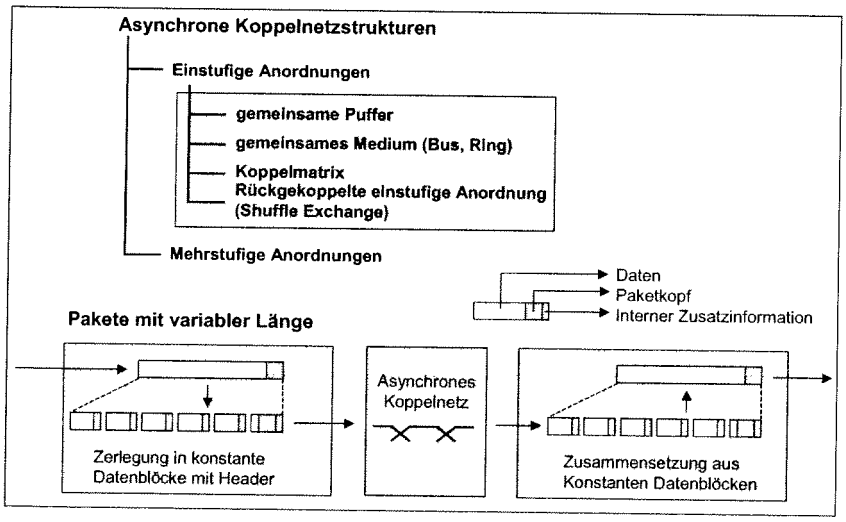


Bild: Asynchrone Koppelnetze

Vermittlungsleistung

Die grobe Architektur eines Switches ähnelt der von Verbindungsnetzen. Eines ihrer entscheidenden Kriterien ist die Fähigkeit zur gleichzeitigen Vermittlung zwischen allen Ein- und Ausgängen.

Bei einer internen Busarchitektur kann das vorhandene Vermittlungsmedium nur nacheinander genutzt werden. Besitzt der Switch-interne Bus jedoch eine Bandbreite, die mindestens so groß wie die Bandbreite aller Eingänge zusammen ist, dann kompensiert er die fehlende Fähigkeit der gleichzeitigen Vermittlung.

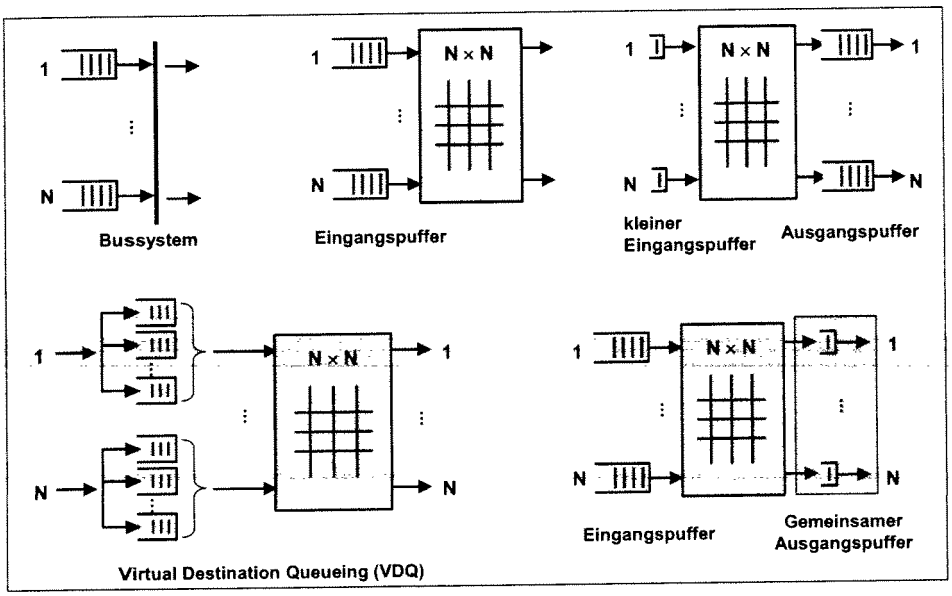


Bild: Koppelnetze in Ethernet-Switches

Die bestmögliche Performance erzielt man mit der Crossbar-Architektur, die jedoch eine hohe Hardwarekomplexität erzwingt.

Sie gestattet ohne Rekonfiguration blockierungsfreie Permutation sowie beliebig viele Multicast- wie auch Broadcast-Verbindungen.

Wenn auch bei einer Crossbar keine internen Konflikte auftreten können, muss man doch mögliche Ausgangskonflikte berücksichtigen. Die minimiert man durch Pufferung. Sie puffert Rahmen gegebenenfalls zwischen, falls der Zielausgang gerade belegt sein sollte und schickt ein Signal an die Steuerlogik. Die wiederum entscheidet, welcher Rahmen als nächstes an den Ausgang weitergeleitet wird. Dabei bestimmt letztendlich die Organisation des Puffers die Leistungsfähigkeit eines Crossbar-Switches.

Pufferung

Die Zwischenpufferung kann man auf vier Weisen organisieren: an den Eingängen, den Ausgängen, verteilt oder zentral. Schaltungstechnisch ist die Eingangspufferung die einfachste Form. Sie hat allerdings den Nachteil, dass Rahmen nicht direkt weitergeleitet werden können, wenn in verschiedenen Eingangspuffern Rahmen für den Ausgang X vorliegen. Dann blockieren sich die Puffer gegenseitig: Rahmen für andere Ziele können so lange nicht weiterlaufen, bis die Rahmen für Ausgang X herausgegangen sind.

Arbeitet man hingegen mit Ausgangspuffern, die nur Rahmen puffern, die zum gleichen Zielport gehören, kann eingangsseitige Head-of-Line Blockierung nicht auftreten. Das stellt jedoch an das Verteilsystem erhöhte Anforderungen, weil an den Eingängen möglicherweise Rahmen für denselben Zielport anstehen. In diesem Fall muss das Verteilungssystem die Rahmen schnell genug zum Ausgangspuffer weiterleiten können.

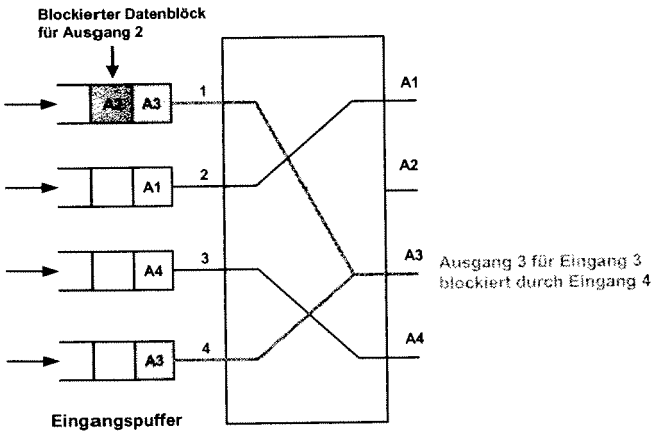


Bild: Head-of-the-Line Blockierung

Head-of-the-Line (HOL) Blockierung

Ein wichtiger Grund für eine verkehrsabhängige Leistungseinbusse von Switches und Routern ist die HOL-Blockierung. Dies passiert, wenn

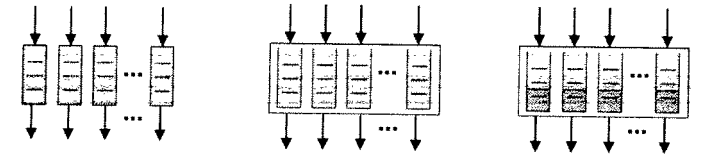
- Rahmen von mehreren Eingängen für den selben Ausgang bestimmt sind.
- der gewünschte Ausgang durch eine temporäre Überlastung nicht zum Übertragung zur Verfügung steht.

Im Bild sind die Rahmen am Eingänge 1 für Ausgang A3 durch Rahmen von Eingang 4 zum selben Ausgang A3 blockiert. Dadurch ist am Eingang 1 auch der Rahmen für Ausgang A2 blockiert.

Bei einem Crossbar-Switch kann man beide Probleme mit einer verteilten Pufferarchitektur vermeiden, bei der jeder Kreuzungspunkt einen eigenen Puffer besitzt. Dadurch steigt aber der Hardware-Aufwand immens. Effizienter ist ein zentraler Speicher. Er nimmt hereinkommende Rahmen über einen Multiplexer entgegen und gibt sie über einen Demultiplexer an den

Zielausgang weiter. Dieses Vorgehen eignet sich besonders für Single-Chip-Lösungen, sie kommen bei einem 8-Port Switch typischerweise mit zwei MByte Gesamtpuffer aus.

Einfache Switches basieren in der Regel auf einer Busarchitektur mit hoher Bandbreite, da diese vom Schaltungsaufwand geringer ist. Dabei fließen die Rahmen von den Eingängen über den ersten Bus an einen zentralen Puffer und von dort über einen zweiten Bus zu den Ausgängen. Für High-End-Switches kommen häufig Crossbars mit zentraler oder verteilter Pufferung zum Einsatz, die in der Regel mittels speziell konstruierter Chips (Application Specific Integrated Circuits, ASICs) aufgebaut werden.



völlig getrennte Pufferbereiche (CP, complete partitioning)

völlig gemeinsamer Pufferbereich (CS, complete sharing)

gemeinsamer Pufferbereich mit richtungsabhängiger Begrenzung (SMXQ, sharing with maximum queue length)

gemeinsamer Pufferbereich mit richtungsabhängiger Reservierung (SMA, sharing with minimum allocation)

gemeinsamer Pufferbereich mit richtungsabhängiger Begrenzung und Reservierung (SMQMA, sharing with maximum queue length and minimum allocation)

Bild: Pufferverwaltung

Ungünstig ist die Zentralpufferung jedoch, wenn ein Ausgang Y - beispielsweise einer, an dem ein Server hängt - häufiger Rahmen erhält als die anderen. Dabei kann der Zentralpuffer mit Rahmen voll laufen, weil Ausgang Y sie nicht schnell genug abnehmen kann. Das kann soweit führen, dass der Zentralpuffer keinen Platz mehr für Rahmen an andere Ausgänge hat.

Um dies zu vermeiden, begrenzt man üblicherweise die Puffermenge, die ein Ausgang belegen darf.

Als Kenngröße ist ferner interessant, wie viele Rahmen ein Switch maximal pro Sekunde weiterleiten kann. Dabei ist das Blocking beziehungsweise, Non-Blocking ein wesentliches Merkmal. Blocking bedeutet, dass der Switch Rahmen mangels ausreichender Bandbreite oder Pufferkapazität verwerfen muss.

Wie viele Verbindungen ein Switch simultan schalten muss, hängt von seiner Port-Anzahl ab. Im schlimmsten Fall empfangen alle Anschlüsse mit voller Geschwindigkeit. Dabei muss die interne Bandbreite für Non-Blocking so groß sein wie die Bandbreite aller Ports zusammen. Bei einem Fast-Ethernet-Switch mit acht Anschlüssen ergeben sich beispielsweise 800 Mbit/s.

Die Anzahl der Rahmen pro Sekunde hängt dagegen von der Bandbreite des Ports und der minimalen Rahmengröße ab, letztere ist bei Ethernet mit 64 Bytes festgelegt, dazu kommt ein gewisser Overhead für Präambel und Inter-Frame-Gap. So liefert ein Port bei 100 Mbit/s maximal 148.800 Rahmen pro Sekunde, bei 10 Mbit/s entsprechend 14.880 und bei 1 Gbit/s 1.488.000 Rahmen/s. Setzt man das für den obigen Switch-Beispiel ein, dann muss dieser über acht simultane Verbindungen knapp 1,2 Millionen Rahmen pro Sekunde. Schafft er das, dann ist er Non-blocking und besitzt die sogenannte Wire-Speed-Fähigkeit.

Vermittlung

Neben der Angabe der Anzahl Rahmen pro Sekunden beeinflusst die Latenz den LAN-Durchsatz. Darunter versteht man die Zeit, die verstreicht, während der Switch ein Datenrahmen verarbeitet. Wie groß die Latenz ist, hängt nun vom verwendeten Switching-Verfahren ab. Dabei unterscheidet man drei Formen: Cut-Through, Store-and-Forward sowie Adaptive-Cut-Through.

- **Cut-Through:** Hier beginnt der Switch mit der Ausgabe des Datenstroms auf den Zielport, sobald er diesen über die Zieladresse identifiziert hat, also schon nachdem die ersten Bytes eines Rahmens hereingekommen sind. Das macht Cut-Through zum schnellsten Verfahren. Ein Nachteil liegt jedoch darin, dass fehlerhafte oder beschädigte Rahmen ungehindert durchlaufen und auch auf der Ausgabeseite eine an sich unnötige Belastung des LAN darstellen.
- **Store-and-Forward:**: Hiermit wird diese Belastung vermeidet, denn der Switch liest erst den vollständige Rahmen ein, puffert es und testet die Richtigkeit anhand der Prüfsumme (CRC). Ist der Rahmen fehlerfrei, gibt der Switch es auf den Zielport aus, wenn nicht, verwirft er es. Wegen der Zwischenpufferung ist Store-and-Forward grundsätzlich langsamer als Cut-Through, dafür minimiert es die Netzbelastung mit fehlerhaften Rahmen. Außerdem kann ein Store-and-Forward-Switch Rahmen zwischen unterschiedlich schnellen Netzsegmenten - beispielsweise von 10 auf 100 Mbit/s -vermitteln.
- **Adaptive-Cut-Through-Verfahren:** Es vereint die Vorteile beider Methoden. Nach der Startphase setzt der Switch zunächst das schnellere Cut-Through ein, prüft jedoch auch dabei anhand der CRC die Fehlerfreiheit. Bei Überschreiten einer festgelegten Fehlerschwelle schaltet er auf Store-and-Forward um. Geht die Fehlerrate später wieder zurück, dann kommt erneut Cut-Through zum Zug. So erreicht man ein Optimum zwischen Performance und Fehlerfreiheit. Adaptive-Cut-Through kommt derzeit nur bei High-End-Switches zur Anwendung, die eine einzige Datenrate unterstützen.

Ferner gibt es Implementierungen, bei denen das Switching-Verfahren von der Rahmenlänge abhängt, falls keine Anpassung der Datenrate notwendig ist. Längere Rahmen leitet der Switch nach dem Empfang der ersten 512 Bytes per Cut-Through weiter. Kürzere Rahmen behandelt er mittels Store-and-Forward. Damit optimiert man dynamisch die Latenz, die sich vor allem bei längeren Rahmen negativ auswirkt. Ein weiterer Ansatz ist Fragment-Free-Cut-Through. Es geht davon aus, dass Übertragungsfehler in der Regel innerhalb der ersten 64 Byte auftreten. Dabei liest ein Switch zunächst 64 Bytes ein, bevor er sie per Cut-Through weiterreicht.

Switch-Durchlaufzeit

Verzögerungen, die Rahmen bei der Weiterleitung erfahren, erfasst man mit der Latenz. Das ist bei Store-and-Forward-Switches die Zeit, die zwischen dem letzten Bit beim Empfang und dem ersten Bit beim Ausgeben verstreicht (LIFO Latency, Last-In First-Out). Anders dagegen beim Cut-Through: Hier misst man die Latenz zwischen dem ersten empfangenen Bit und dem ersten ausgesendeten Bit (FIFO Latency, First In First Out). Diese unterschiedlichen Definitionen erschweren auf den ersten Blick einen direkten Vergleich.

In den Datenblättern der Hersteller hat ein Store-and-Forward-Switch bei 100 Mbit/s typischerweise eine Latenz von 10 bis 18 µs, ein Cut-Through-Switch jedoch zirka 35 µs. Das erscheint zunächst als Widerspruch zur Aussage, dass Store-and-Forward schneller als Cut-Through ist. Berücksichtigt man aber die unterschiedliche Messmethode, dann stimmt das Bild wieder: Ein 512 Byte langes Fast-Ethernet-Rahmen braucht rund 41 µs. Es ist per Cut-Through nach etwa 76 µs (Latenz plus Rahmendauer) beim Empfänger, per Store-and-Forward dagegen erst nach 92 bis 100 µs (Latenz plus zweifache Rahmendauer, vollständiges Empfangen und Wiederaussenden).

Switch-Durchsatz

Bei einem idealen Switch ist die Latenz weitgehend unabhängig von der Anzahl genutzter Ports, der Gesamtlast und der Tatsache, ob die Weiterleitung über die Layer 2 (Bridging) oder Layer 3 (Routing) erfolgt. Bei einem modularen Switch, also einem per Steckmodulen erweiterbaren, kann die Latenz jedoch sehr unterschiedlich ausfallen: Innerhalb eines Moduls entsteht typischerweise eine geringere Verzögerung, als wenn der Rahmen zu einem anderem Modul weitergeleitet werden muss, da sich alle Module die Bandbreite der Backplane (gemeinsames Bussystem) teilen.

Denn auch die Backplane muss für einen verzögerungsfreien Datentransport nichtblockierend arbeiten können. Hat man beispielsweise einen Switch mit zwei Modulen zu je 24 Fast-Ethernet-Ports, dann muss die Backplane 4,8 Gbit/s durchschleusen, um in Hochlastsituationen nicht zum Flaschenhals zu werden.

Diese Betrachtung gilt genauso bei einem Stack, also einem Verbund mehrerer gestapelter Switches, die eine logische Einheit bilden. Bei 100-Mbit-Switches nutzt man gern Gigabit-Ethernet-Ports für die Verbindung. Koppelt man darüber zwei Switches mit je 24 Fast-Ethernet-Ports, dann liegt die verfügbare Bandbreite mit 2 Gbit/s weit unter den geforderten 4,8 Gbit/s. Solch ein Stack wäre nicht nichtblockierend, man muss jedoch im Hochlastfall mit einer erhöhten Latenz rechnen..

Die Datenblattangaben modularer Switches fallen häufig recht verschieden aus. Viele Hersteller addieren einfach die Bandbreite sämtlicher Module. Gleiches gilt für die Angabe der Rahmen pro Sekunde, angegeben als Pakete pro Sekunde (PPS). So liest man bei einem modularen Switch schnell mal von mehr als 30 Gbit/s und einem Durchsatz über 44 MPPS. Ein genaues Betrachten der Datenblätter kann jedoch zeigen, dass die Backplane nur 21 Gbit/s aufweist. Beim einem System mit fünf Modulen zu je 48 Fast-Ethernet-Ports wird die vorhandene Backplane-Leistung im Hochlastfall nicht ausreichen. Rechnerisch wären 24 Gbit/s nötig, wenn der gesamte Datenverkehr im Worst-Case-Fall über die Backplane geht. Das Beispiel zeigt, dass solch ein Switch nur im Idealfall nichtblockierend arbeitet. Bei der Installation von größeren Switch-Anlagen sollte man deshalb durch geschicktes Patching dafür sorgen, dass ein möglichst kleiner Teil der Rahmen über die Backplane läuft und der Großteil innerhalb der Module bleibt.

Die Leistungsfähigkeit eines Switches kann man beispielsweise anhand von Latenz, Durchsatz und verlorenen Rahmen messen, typischerweise in Szenarien, die den Hochlastfall simulieren. Dafür kommen spezielle Lastgeneratoren zum Einsatz, die auf allen Ports des Switches künstlicher Verkehr erzeugen und gleichzeitig Messergebnisse darstellen

Flusskontrolle

Um einen Pufferüberlauf und daraus folgendes Verwerfen von Rahmen zu vermeiden, wurde die so genannte Flow Control im Standard IEEE 802.3x eingeführt. Droht ein Puffer überzulaufen, kann ein Switch ein angeschlossenes Gerät mittels Pause-Rahmen auffordern, vorübergehend keine weiteren Rahmen zu senden. Pause-Rahmen sind spezielle MAC-Kontrollrahmen, die als Multicast an die festgelegte Adresse 01-80-C2-00-00-01 gehen und im Length/Typ-Feld den Wert 88-08 enthalten. Ob bei einer Verbindung Flusskontrolle angewendet wird, entscheiden die Kommunikationspartner während des Link-Aufbaus über Auto-Negotiation-Funktion. Üblicherweise beherrschen alle Gigabit-Ethernet- und eine Vielzahl von Fast-Ethernet-Komponenten das Verfahren.

Flow-Control funktioniert grundsätzlich nur im Vollduplexmodus. Bei Ports, die halbduplex arbeiten, können Switches stattdessen Back-Pressure verwenden, das auf der Simulation von Kollisionen beruht. Der Switch sendet bei drohendem Überlauf

ein JAM-Signal aus, was das angeschlossene Endgerät veranlasst, ihren Sendevorgang abzubrechen und ihre Daten nach einer bestimmten Wartezeit erneut zu schicken.

Falls im Vollduplex kein Flow-Control möglich ist, weil beispielsweise ein Link-Partner das nicht unterstützt, bieten viele Switches eine Head-of-Line-Blocking-Funktion. Sie soll sicherstellen, dass keine Staus entstehen, wenn ein Ausgangspuffer voll ist. Dazu prüft der Switch die Zieladresse und deren Port-Zuordnung. Ist dessen Ausgangspuffer blockiert, dann verwirft er den Rahmen, damit der Eingangsport frei bleibt.

Broadcast-Rahmen können auch ein geswitchtes LAN stark belasten. Als Ursache kommen etwa Print-Server in Frage, die ihre Dienste per Broadcast anbieten, schlimmstenfalls so häufig, dass Broadcast-Bursts entstehen. Um denen entgegenzuwirken, besitzen viele Switches Broadcast-Filter. Diese überwachen den Datenverkehr. Übersteigt der Broadcast-Anteil beispielsweise 20 Prozent der gesamten Port-Kapazität, dann blockiert der Switch weitere Broadcasts auf diesem Port für eine bestimmte Zeit, meistens rund zehn Sekunden. Sinkt der Broadcast-Anteil anschließend, dann lässt der Switch nach weiteren zehn Sekunden die Aussendung wieder zu. Die Ansprechschwelle kann man dabei über die Broadcast Cut-off Rate einstellen.

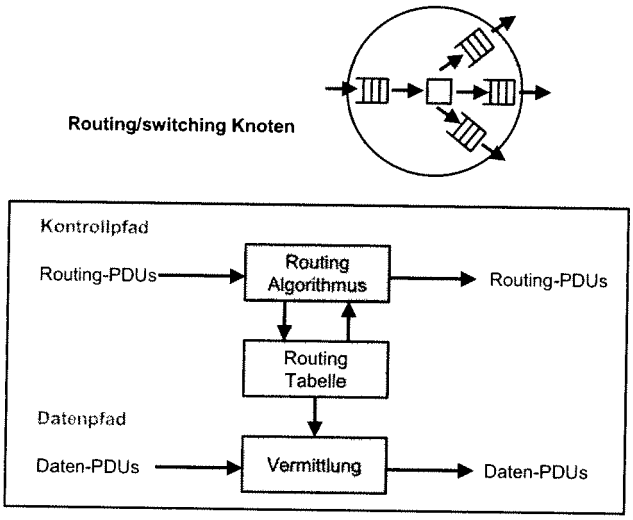


Bild: Kontroll- und Datenpfad

Content Switching

Neben reinen Layer-2-Switches gibt es auch zunehmend Layer-3-Switches, die Adressinformationen der nächst höheren Protokollebene auswerten und für die Wegwahl nutzen können.

In der Regel unterstützen Layer-3-Switches IP-basiertes Routing, wobei sie Protokolle wie RIP (Routing Information Protocol), RIPv2 und OSPF (Open Shortest Path First) nutzen.

Herkömmliche Router haben gegenüber Switches eine höhere Latenz. Leistungsfähige Layer-3-Switches erledigen hingegen das Routing mit Wire-Speed, weshalb sie traditionellen Routern inzwischen den Rang ablaufen.

Einige Hersteller von Layer-3-Switches haben zudem einen Mechanismus für die Verringerung der Latenz nach der auf dem Grundsatz 'erst Layer 3, dann Layer 2' implementiert. Dabei nutzt man die aus dem Routing gewonnenen Informationen, um eine Port-Zuordnung für die MAC-Ebene zu bilden, die das Switching dann verwertet. Damit geht konventionelles Routing nach einer bestimmten Zeit in schnelleres Switching über.

Die nächste Steigerung sind Layer-4- bis Layer-7-Switches, die für die Wegwahl Informationen der höheren OSI-Schichten nutzen (Content-Switching). So werten Layer-4-Switches beispielsweise fest definierte Ports wie Port 80 (http-Verkehr) aus und leiten Daten dafür an einen Webserver weiter. Layer-7-Switches gehen noch einen Schritt weiter und werten Anwenderinformationen aus, indem sie nach Schlüsselinformationen, zum Beispiel spezielle Cookies, in den Nutzdaten suchen.

Virtuelles vermitteln

Generell geht es beim Content Switching um eine inhaltsbezogene Optimierung des Datenverkehrs. Ein Content-Switch würde Datenverkehr zu Servern, Firewalls und Massenspeichern erkennen und gegenüber dem Anwender als virtueller Server erscheinen, hinter dem sich mehrere Server oder ein Cluster verbirgt. Solches Vorgehen ermöglicht neben der Optimierung der Reaktionszeiten auch eine höhere Ausfallsicherheit, indem Last und Daten auf mehrere Server verteilt wird. Da Wegwahl und Ausführungsentscheidung in Leistungsgeschwindigkeit eine hohe Rechenleistung erfordern, setzen Content-Switches auf einen oder mehrere RISC-Prozessoren, gekoppelt mit einem großen Speicher. Derartige High-End-Lösungen sind besonders für Web-Hosting-Infrastrukturen und E-Commerce-Transaktionen interessant.

- Es wird eine bestimmte Gruppe von Ports von mehreren Switches zu einem virtuell eigenständigen Netz zusammengefasst.
- Dadurch entstehen mehrere Broadcast-Domänen → eine Entlastung des Netzes durch Minderung des Broadcast-basierten Datenverkehrs.
- Es wird ein so genanter VLAN-Tag-Header spezifiziert (802.1q) und im Ethernet-Rahmen zwischen der Quelladresse und dem Längefeld eingebettet (802.3ac - 1998).
- Alle Stationen in einem VLAN gehören zu einer Broadcast-Domäne
- Endstationen können gleichzeitig mehreren VLANs gehören

Bild: Virtuelle LANs (VLANs)

Die meisten managebaren Switches bieten Virtual Bridged Local Area Networks, kurz VLANs, an. Diese stellen, auf der MAC-Ebene implementiert, eine Gruppierung von bestimmten Stationen innerhalb des gesamten LANs dar. Der Vorteil: Das Netz wird in mehrere Broadcast-Domänen aufgeteilt. Außerdem kann man die Datensicherheit verbessern, da nur die Teilnehmer eines VLAN - beispielsweise die Mitarbeiter der einzelnen Abteilungen eines großen Unternehmens - untereinander Daten austauschen können.

- Switches erlauben keine Kommunikation zwischen unterschiedlichen VLANs → Verwendung von Routers
- Routers können entweder direkt an mehrere VLANs angeschlossen werden oder sie unterstützen VLAN trunking (Erkennung, Bearbeitung und Änderung des VLAN-Tags)
- VLAN Arten:
  - VLANs basierend auf der MAC-Adresse
  - Portbasierende VLANs
  - Protokollbasierende VLANs
- Implizites Tagging:
  - Die Zugehörigkeit zu einer VLAN-Gruppe wird durch die MAC-Adresse bestimmt.
- Explizites Tagging:
  - Die Zugehörigkeit zu einer VLAN-Gruppe wird durch eine Markierung (Protokoll-Tag oder Port-Tag) bestimmt.

Bild: Kommunikation in VLANs

Für die Umsetzung gibt es zwei Ansätze: Die einfachste Variante sind port-basierte VLANs, bei denen der Switch bestimmte Ports einem VLAN zuordnet. Er sendet Broadcast-Rahmen nur noch an die Ports aus, die zu dem VLAN gehören, aus dem sie stammen. Für Switch-übergreifende VLANs existiert IEEE-802.1Q. Hierbei verlängert man den Ethernet-Rahmen um vier Bytes und packt zusätzliche Informationen in den Header, die den Datenaustausch innerhalb der VLANs regeln (VLAN-Tagging). Damit kann man VLANs über mehrere Switches hinweg ausdehnen, vorausgesetzt, alle beteiligten Switches können die VLAN-Informationen interpretieren und die Treiber auf den vernetzten Rechnern setzen die VLAN-Information in die Rahmen.

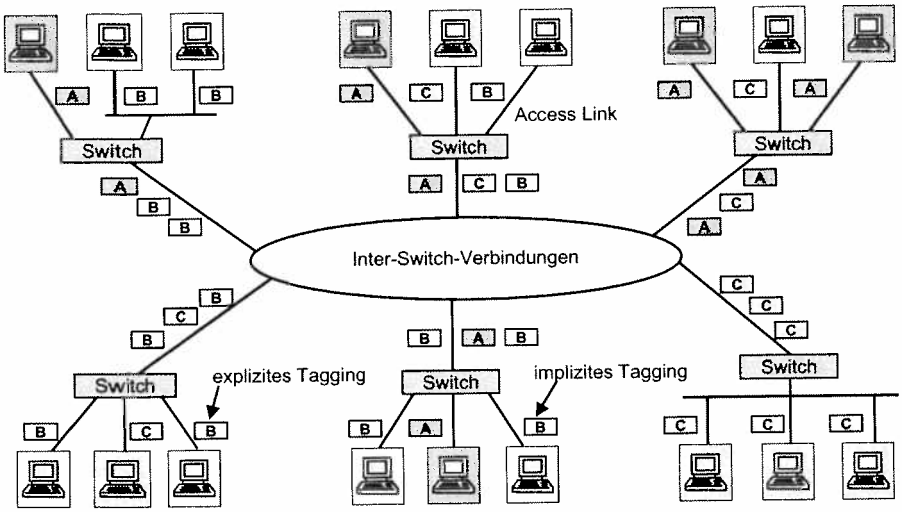


Bild: Virtuelle LANs mit IEEE 802.1Q Tagging

Durch einen sogenannten IEEE 802.1Q Tag können Virtuelle LANs (VLANs) aufgebaut werden. Es gibt verschiedenen Gruppen von Stationen, die zu einem VLAN gehören. Ihre Rahmen werden nur an Stationen innerhalb einer Gruppe weitergeleitet.

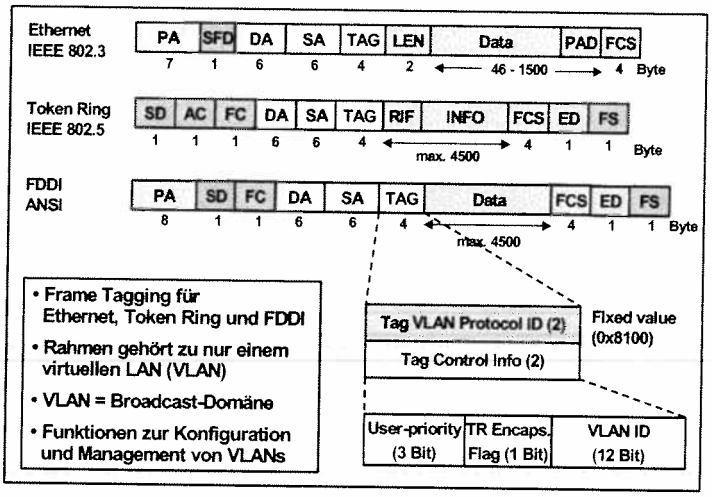


Bild: IEEE 802.1Q

Prioritäten

High-End-Switches bieten häufig auch Quality-of-Service-Funktionen (QoS), mit denen man den weiterzuleitenden Rahmen unterschiedliche Prioritäten einräumen kann. Dieses Feature wird mit dem Aufkommen von Voice-over-IP (VoIP, LAN- beziehungsweise Internet-Telefonie) und Videostreams immer wichtiger, weil vor allem Sprache gegenüber zu hohen Latenzen sehr empfindlich ist. Mittels QoS kann man für solche Anwendungen Bandbreite reservieren und bestimmte Rahmen priorisieren, was jedoch garantierte Verbindungseigenschaften erfordert. Da dies jedoch beim paketvermittelten Ethernet nicht hundertprozentig umsetzbar ist, spricht man eher von Class-of-Service, kurz CoS, das bei Ethernet bereits auf der MAC-Schicht realisierbar ist: Innerhalb des zwei Byte langen VLAN-Tags sind drei Bits für acht unterschiedliche Prioritäten vorgesehen.

Damit ein Switch unterschiedliche Prioritäten realisieren kann, muss er intern verschiedene Queues (Pufferschlangen) bereitstellen, für jede Priorität eine eigene. Stehen in der Queue mit der höchsten Priorität Daten an, so leitet der Switch diese bevorzugt weiter. Nachdem alle mit der höchsten Priorität abgearbeitet sind, kommt die nächst niedrigere Queue dran. Dabei muss man freilich darauf achten, dass Queues mit niedrigerer Priorität nicht zu stark unterdrückt werden. Dazu ordnet man jeder Queue einen Bandbreitenanteil zu und sorgt dafür, dass auch für die am wenigsten dringende Queue eine gewisse Restbandbreite übrig bleibt.

Schließlich gibt es einen Ansatz auf der dritten OSI-Schicht (IP) namens Differential Service, kurz DiffServ. Er nutzt das Type-of-Service-Feld (ToS) im IP-Header für verschiedene Dienstklassen (CoS). Dabei klassifizieren wiederum die auf den Stationen laufenden Applikationen ihre Daten. Die Switches müssen diese dann unter Berücksichtigung einer bestimmten Bandbreitenzuteilung und maximalen Verzögerungszeit weiterleiten.

Bei größeren LANs steht neben der Durchsatzoptimierung auch die Ausfallsicherheit auf der Agenda. Letztere kann man durch Redundanzschaffung verbessern. Dabei verbindet man alle Switches in einem Firmen-LAN über einen oder mehrere Ringe auf unterschiedlichen physischen Wegen, was eigentlich verboten ist. Damit in solchen Schleifen keine Rahmen endlos kreisen, kommt das Spanning-Tree-Verfahren zum Einsatz. Es ist im IEEE 802.1d Standard für die MAC-Ebene spezifiziert und basiert auf dem Austausch von Konfigurationsrahmen (Bridge Protocol Data Units, BPDU), die über Multicast Rahmen an eine bestimmte Adresse (01-80-C2-00-00-10) gehen,

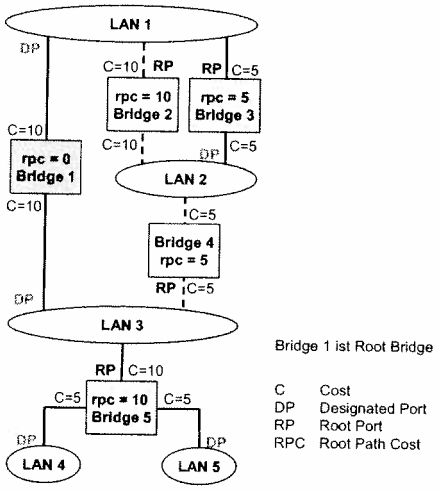


Bild: Spanning Tree

Bäume spannen

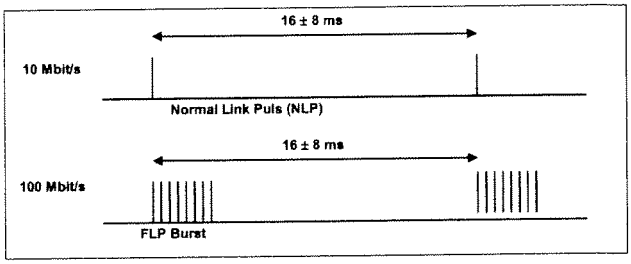
Über BPDUs nehmen die beteiligten Switches eine Topologiekontrolle vor, die redundante Strecken erkennt und zwischen diesen die optimale Route ermittelt. wobei sie unter anderem die möglichen Datenraten und Entfernungen berücksichtigen. Diese Parameter - wie auch Prioritäten und Wegekosten - sind normalerweise konfigurierbar, sodass der Netz-Admin bevorzugte Routen festlegen kann. Die Ports der redundanten Strecken werden zunächst abgeschaltet und erst beim Ausfall der Hauptroute reaktiviert. Jeder Spanning-Treefähige Switch muss dazu über eine eigene Bridge-ID und MAC-Adresse verfügen.

Nachteilig am herkömmlichen Spanning Tree ist, dass es bereits in einem kleinen LAN nach einem Störfall eine halbe bis eine Minute für die Umkonfiguration benötigt. Das in IEEE 802.1w beschriebene, abwärtskompatible Rapid Spanning Tree (RSTP) oder Fast Spanning Tree soll den Vorgang deutlich beschleunigen: Man verspricht sich davon eine Umkonfiguration innerhalb einer Sekunde nach dem Störfall.

Einfache Switches ohne Spanning-Tree besitzen stattdessen häufig eine Loop-Detection-Funktion. Sie soll zumindest eine Schleifenbildung erkennen, wenn schon nicht verhindern, so dass der Administrator eingreifen kann. Dazu sendet ein Switch alle paar Minuten ein Rahmen an eine bestimmte Adresse aus, das ihn über die Quelladresse identifiziert. Empfängt er solch ein Rahmen mit seiner eigenen Adresse, existiert eine Schleife und der Switch zeigt das über eine Loop-LED an.

- Bei der Auto-Negotiation (der automatischen Verhandlung) handeln zwei Geräte in einer Verbindung automatisch die beste gemeinsame Geschwindigkeit aus.
- Das im 802.3 Standard festgelegte Auto-Negotiation-Protocol (ANP) basiert auf dem von National Semiconductor entwickelten Nway-Protokoll.
- ANP wird unmittelbar bei der Initialisierung der Verbindung aufgerufen und verwendet ein dediziertes Signalsystem. Dieses basiert auf den Normal Link Pulses (NLP), die 10Base-T zur Kontrolle der Verbindung regelmäßig versendet.

Bild: Auto-Negotiation



- Equipment detektiert 10 oder 100 Mbit/s auf Punkt-zu-Punkt-Link.
- Senden von Fast Link Pulse (FLP) Bursts.
- Empfangen von FLP Burst erlaubt Benutzen des 100 Mbit/s Modus.

Bild: Informationsaustausch bei Auto-Negation

- ANP sendet Signalfolgen mit 33 Fast Link Pulses (FLP), deren Timing genau den Normal Link Pulses (NLP) entspricht und eine abwechselnde Takt/Daten Sequenz darstellt.
- Anhand einer Prioritätenfolge werden dann Geschwindigkeit und Modus ausgewählt.
- Die Prioritätenfolge sieht so aus (beste Möglichkeit zuerst):
  - 100BASE-TX Vollduplex
  - 100BASE-TX Halbduplex
  - 10BASE-T Vollduplex
  - 10BASE-T Halbduplex
- Wenn ein Gerät nicht auf die FLPs antwortet, greift die sogenannte Parallel Detection, die den Übertragungsstandard anhand der Signalform und der Kodierung erkennt. Dabei wird allerdings standardmäßig der Halbduplex-betrieb ausgewählt.

Bild: Ablauf der Auto-Negotiation (ANP)

Link Code Word (LCW):

Bit(s)	Bedeutung
0 - 4	Bei Ethernet immer 10000. Damit kann der Standard auch auf andere Übertragungssysteme erweitert werden.
5	unterstützt 10Base-T
6	unterstützt 10Base - T Full Duplex
7	unterstützt 100Base -TX
8	unterstützt 100Base - TX Full Duplex
9	Unterstützt 100Base -T4
10	unterstützt Datenflusskontrolle
11	reserviert
12	reserviert
13	Fehlerindikator
14	ACK: Quittierung eines ANP - Datenpakets
15	NP (Next Page): Es folgen weitere Datenpakete mit herstellerspezifischen Informationen.

Bild: Link-Codewort

- Das ANP bietet die Möglichkeit, zusätzliche Informationen auf weiteren Seiten (Next Page) zu übermitteln. Auf diese Weise können auch die Gigabit-Standards in das ANP einbezogen werden.
- Wenn das „Next Page"-Bit (Nr. 15) in beiden Richtungen gesetzt ist, heißt es, dass die „Next Page"-Funktion von beiden Seiten unterstützt wird.

Bit(s)	Bedeutung
0 - 10	Message - Block
11	Toggle- Bit
12	ACK- 2
13	Message - Bit
14	ACK
15	Next-Page- Bit

Bild: „Next Page" Funktion

- Message-Bit auf logisch 1 gesetzt: eine Message-Seite wird übertragen
  - Bits 0 bis 10 beinhalten eine von IEEE-Komitee definierte Message
- Message-Bit auf logisch 0 gesetzt: eine Unformatierte-Seite wird übertragen
  - Bits 0 bis 10 beinhalten proprietäre Informationen (z. B. Hersteller- und Geräteinformationen)
- Toggle-Bit unterscheidet die nachfolgenden Seiten, wobei es immer den entgegengesetzten Wert des vorgegangenen Codewords annimmt.
- ACK-2-Bit wird verwendet um den Empfang des Message-Blocks zu bestätigen.
- ACK-Bit bestätigt den Empfand des vorherigen Codeworts
- Der Next-Page-Bit zeigt dem Verbindungspartner an, dass noch weitere „Next Pages" folgen.

Bild: „Next Page" Codewort

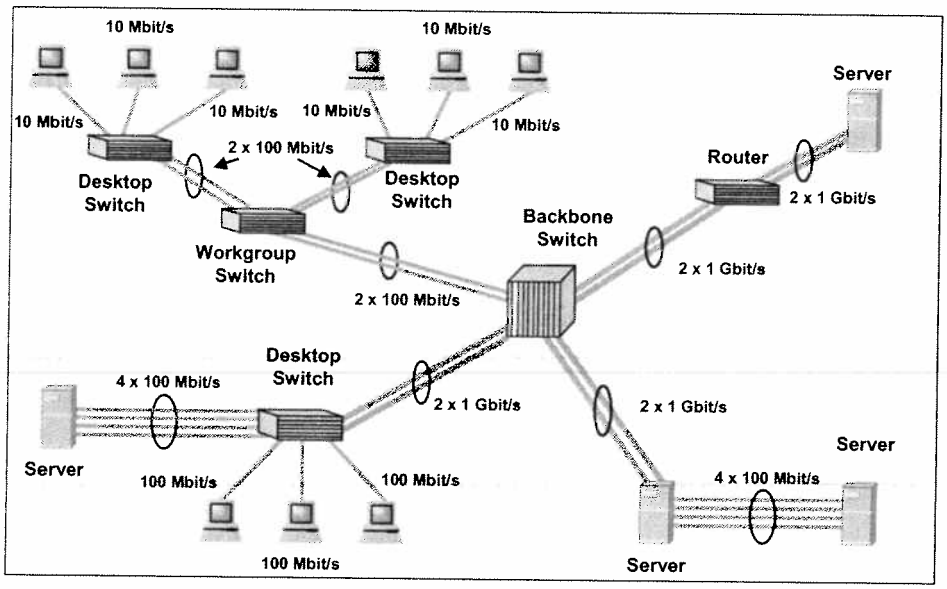


Bild: Link Aggregation

Erhöhung der Linkkapazität

Mit Spanning Tree verwandt ist die Link Aggregation (IEEE 802.3ad). Dabei bündelt man im Punkt-zu-Punkt-Betrieb - also beispielsweise zwischen zwei Switches oder Switch und Server - mehrere Leitungen, um die Bandbreite zu erhöhen oder Redundanz zu schaffen. Das funktioniert grundsätzlich nur bei Ports mit derselben Datenrate und im Vollduplexbetrieb. Die zusammengefassten Anschlüsse bilden logisch betrachtet einen einzelnen Port mit vervielfachter Bandbreite.

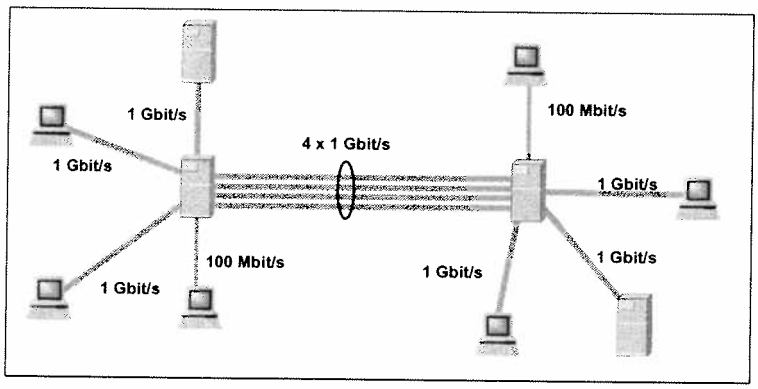


Bild: Netzknoten-zu-Netzknoten Verbindung

Fällt nun eine der Strecken aus, kann der Datenverkehr über die restlichen Leitungen weiterlaufen.
Anfangs wurde diese Parallelschaltung auch als Trunking oder Channel Bundling bezeichnet, wobei die Hersteller unterschiedliche Lösungsansätze implementierten, die zueinander inkompatibel waren. Mit der Verabschiedung des IEEE-Standards entstand eine einheitliche Lösung unter dem Begriff Link Aggregation. Für die Umsetzung wurde in die MAC-Schicht ein Link Aggregation Control Layer eingeschoben, der für die Verteilung des Datenstroms sorgt.

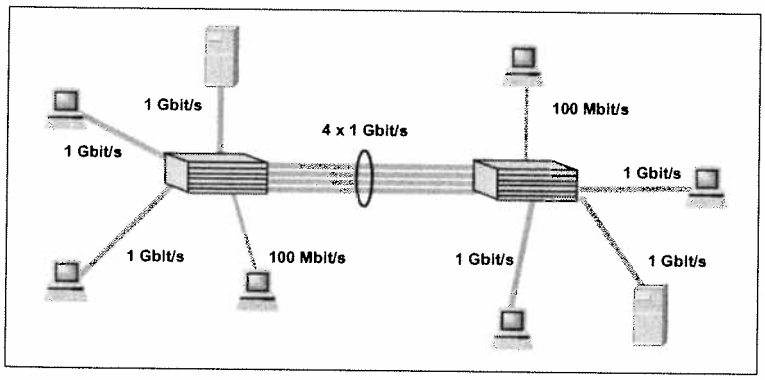


Bild: Switch-zu-Switch Verbindung

Wie der Informationsaustausch zur Steuerung zwischen den Verbindungspartnern abläuft, definiert das Link Aggregation Control Protocol (LACP). Seine Informationen liegen im Datenteil eines regulären Rahmens, sie werden regelmäßig oder nach einer Konfigurationsänderung ausgetauscht. Link Aggregation kann man entweder für Verbindungen zwischen Switches einsetzen oder für Verbindungen zwischen einem Switch und einem Server, der mehrere Netzkarten enthält. Letzteres ist derzeit allenfalls bei Fast-Ethernet (100 Mbit/s) sinnvoll, denn schon eine Gigabit-Ethernet-Karte reizt den verbreiteten PCI-Bus bis zum Anschlag aus.

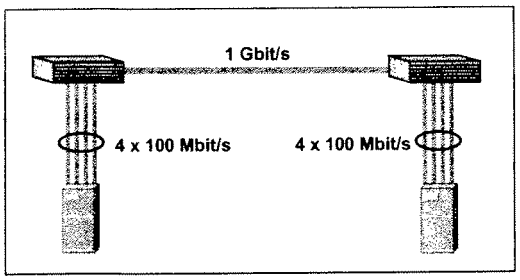


Bild: Switch-zu-Netzknoten Verbindung

Außerdem braucht man mit heute üblichen Betriebssystemen schon eine CPU der 2-GHz-Klasse, um nur einen Gigabit-Ethernet-Kanal auszulasten. Bei Verbindungen zwischen größeren Switches kann dagegen die Bündelung mehrerer Gigabit-Strecken Engpässe aufweiten.

Bei einfachen Geräten lassen sich bis zu vier Ports zusammenfassen, bei High-End-Switches gar 36 oder mehr.

- Parallelschaltung mehrerer Datenleitungen (Trunking):
  - ➢ Die Bandbreiten einzelnen Datenleitungen werden gebündelt
  - ➢ Eine redundante Verbindung wird geschaffen (Sicherheitsaspekt)
  - ➢ Lineare Erhöhung der Link-Kapazität unter Verwendung gleicher (bereits installierter) Hardware
  - ➢ Schnelle Konfigurierbarkeit
- Link Aggregation Arten:
  - ➢ Switch-zu-Switch Verbindung
  - ➢ Switch-zu-Netzknoten (Server oder Router) Verbindung
  - ➢ Netzknoten-zu-Netzknoten Verbindung

Bild: Link Aggregation (IEEE 802.3ad)

- Virtuelle Adresse:
  - Bildung einer so genannten „virtuellen" Adresse, die einer der MAC-Adressen von im „aggregated link" beteiligten DTEs/DCEs entspricht.
- Rahmenverteilung:
  - Ethernet-Rahmen, die zu einer logischen Verbindung (einer Session) gehören, dürfen nicht auf zwei physikalischen Datenleitungen übertragen werden. Das Ende einer Session wird durch eine „Marker Message" signalisiert.
- Datendurchsatz:
  - Die Link-Bündelung erhöht zwar die Kapazität einer Verbindung, der Datendurchsatz in jedem Link bleibt jedoch gleich.

Bild: Annahmen zur Link Aggregation

- Unter Verwendung von LACP (Link Aggregation Control Protocol) wird es möglich, alle aggregationsfähigen Verbindungen im System zu erkennen und zu lokalisieren.
- Von allen aggregationsfähigen Ports werden automatisch Paare gebildet (Aggregators).
- Der Aggregator mit den meisten aktiven Verbindungen wird zum aktiven Aggregator.
- Alle anderen Aggregators bleiben im „Hot Standby"-Zustand.

Bild: Link Aggregation Control Protocol

Topologien

Die Topologie eines Netzes ist der Zusammenschluss aller Stationen bzw. Netzknoten zu einem Gesamtnetz. Obwohl der Begriff Topologie abstrahiert von der verwendeten Leitungs- und Verbindungstechnik zu sehen ist, wird durch die Wahl der Verkabelung die Topologie des Netzes meistens bereits festgelegt.

Aus der Netztopologie heraus lassen sich bereits Leistungs- und Stabilitätsparameter ableiten:

- Möglichkeiten und Verhalten zur bzw. bei Skalierung des Netzes sowie hierbei anfallende Kosten,
- Netzreaktion auf den Ausfall von Stationen oder Leitungen,
- Anzahl der Leitungen, die ausfallen dürfen, ohne dass eine Station von der Netzverbindung abgeschlossen wird,
- Methoden zur Wegefindung (Routing),
- Notwendiger Protokolloverhead zur fehlerfreien Kommunikation.

Lange Zeit war man durch den Hersteller auf eine bestimmte Topologie festgelegt. Dies hat sich durch die Spezifikation von Standards grundlegend geändert. Man ist auch durch die Auswahl der Netztechnologie nicht mehr so stark an Strukturen gebunden wie das früher der Fall war. Zu nennen sind die Topologien Bus, Ring, Stern und Baum. Ein Trend ist heute in der Vermaschung zu sehen, um redundante Wege zu schaffen. Weiterhin wird aufgrund der strukturierten Verkabelung meistens eine Baum-Verkabelung im LAN und eine Stern-Verkabelung im WAN gewählt.

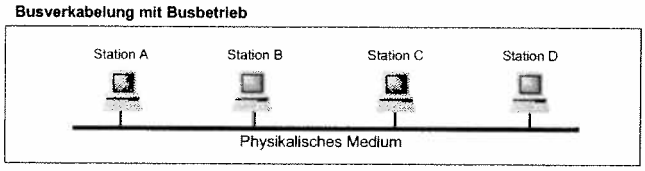


Bild: Physikalische Bus-Topologie

Bus-Topologie
Die Bus-Topologie ist trotz der Einführung von Hochgeschwindigkeitsnetzen noch die meist verbreitete Topologie. Sie ist das klassische Beispiel eines Broadcastnetzes.

Alle Stationen teilen sich dabei ein gemeinsames Medium (Shared LAN), wodurch jede Nachricht direkten Zugriff auf den Bus besitzt. Erweiterungen des Busses um weitere Endgeräte bzw. Stationen und die maximale Länge werden durch Zugriffsprotokolle und Kabel begrenzt. Außerdem ist eine Erweiterung des Netzes oder das Hinzunehmen weiterer Stationen mit einem kurzen Ausfall des LANs verbunden. Der Ausfall einzelner Stationen beeinträchtigt die Netzfunktionen nicht. Die Station ist dann nur nicht mehr für das LAN erreichbar. Eine Beschädigung des Busses oder ein Ausfall der Terminierung an einem Busende bedeutet allerdings den Kommunikationsabbruch unter allen angeschlossenen Stationen. Typische Beispiele einer Bus-Verkabelung sind das Ethernet in den Ausführungen 10Base-5 und 10Base-2. Das historische Standardmedium war das Koaxialkabel. Heute baut man die Bus-Topologie jedoch auch als Zusammenschaltung von Stern-Topologien auf.

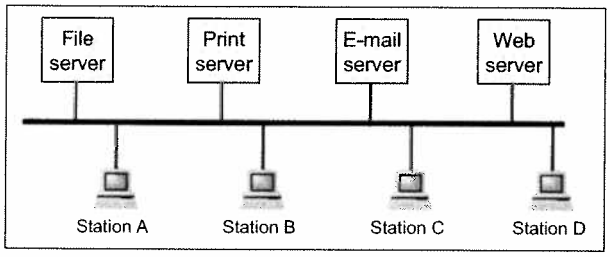


Bild: Ressourcenaufteilung

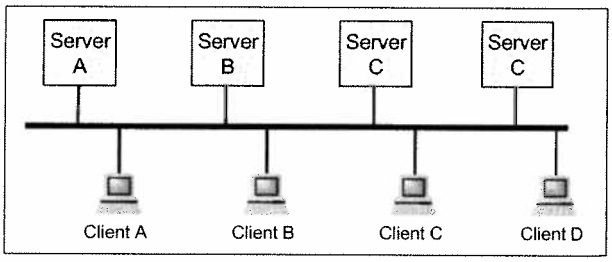


Bild: Client-Server Architektur

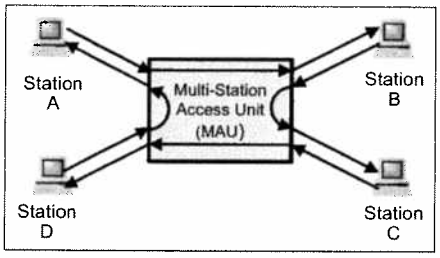


Bild: Physikalische Stern-Topologie und logische Ring-Topologie

Ring-Topologie
Die Ring-Topologie verbindet jede Station mit ihren beiden Nachbarstationen. Eine Station empfängt die im Ring übertragene Nachricht und leitet sie an die jeweilige Station weiter. Der Nachrichtenumlauf im Ring ist dabei abhängig von der Richtung. Der Ausfall eines Segments bzw. einer Station hat bereits den Ausfall des gesamten Netzes zur Folge. Aus diesem Grund wird der Ring meistens doppelt ausgelegt. Der primäre Ring wird dann für die Nachrichtenübertragung verwendet, während der sekundäre Ring als zusätzliche Redundanz zur Verfügung steht. Als Backup-Medium kann er ebenfalls sehr gut eingesetzt werden.

Wenn es doch einmal zu einem Ausfall kommt, wird das gesamte Netz rekonfiguriert. Das heißt, die beiden offenen Ringe werden zu einem primären Ring zusammengeschlossen.

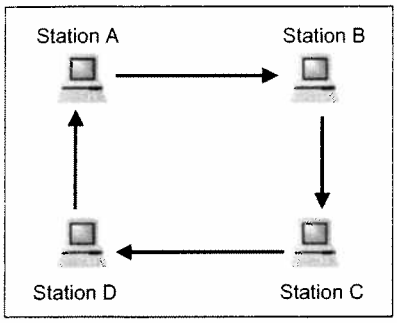


Bild: Physikalische und logische Ring-Topologie

Das Netz kann dadurch, wenn auch nur halb so leistungsfähig, aktiv bleiben. Ein weiterer Segmentausfall führt zum Zerfall des Ringes in zwei funktionsfähige Teile. Ähnlich wie bei der Bus-Topologie bei 10Base-2 bedeutet die Hinzunahme einer weiteren Station eine kurzzeitige Netzunterbrechung. Der typische Vertreter für den Doppelring ist Fiber Distributed Data Interface (FDDI). Der ursprünglich von IBM spezifizierte Token Ring folgt physikalisch gesehen eher der Stern-Topologie, da der eigentliche Ring in einem zentralisierten Verteiler nachgebildet wird. Logisch ist der Token Ring aber in jedem Fall der Ring-Topologie zuzuordnen.

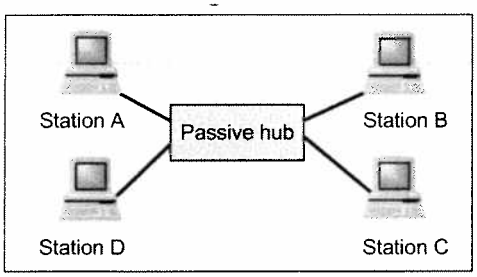


Bild: Physikalische Stern- und logische Bus-Topologie

Stern-Topologie

Bei der Stern-Topologie sind die Stationen sternförmig an einen zentralen Knoten angeschlossen. jede Station ist exklusiv mit ihm verbunden, so dass jede Kommunikation über diesen Knoten abgewickelt werden muss. Durch unterschiedliche Schaltungsarten des zentralen Knotens können verschiedene Topologien nachgebildet werden. Man unterscheidet deshalb zwischen physikalischer und logischer Topologie.

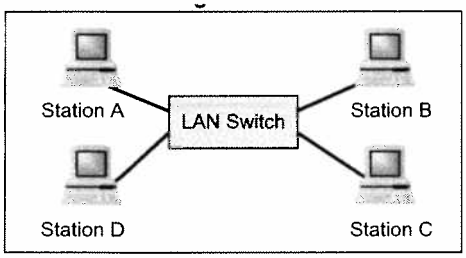


Bild: Physikalische und logische Stern-Topologie

So kann man beispielsweise durch eine physikalische Stern-Topologie durchaus eine logische Ring-, Bus- oder Baum-Topologie abbilden. Da der Verkabelungsstandard in einigen Bereichen eine physikalische Stern-Topologie vorschreibt, können nachträglich beliebige Topologien eingesetzt werden. Die Vorteile liegen somit in der einfachen Erweiterbarkeit des LANs sowie seiner Stabilität in Hinblick auf den Ausfall einzelner Segmente. Bei dem Ausfall einzelner Leiter werden nur die Stationen gestört, die mit diesem verbunden sind. Die restlichen Segmente werden nicht betroffen.

Stern-Topologien können Daten nur auf dem Umweg über den zentralen Knoten austauschen. Das ist auch gleichzeitig der fundamentale Nachteil dieser Zentralisierung, da ein Ausfall des zentralen Knotens das ganze Netz zusammenbrechen läßt. Man unterscheidet hierbei zwischen aktiven und passiven Sternsystemen. Bei aktiven Sternsystemen ist der zentrale Knoten ein Rechner, der die Weiterübermittlung der Nachrichten übernimmt. Seine Leistungsfähigkeit bestimmt die Performance des Netzes. Beispiel: Nebenstellenanlagen oder Switches. Passive Sternsysteme haben in der Mitte nur einen Knoten, der die Wege zusammenfasst. Dieser Knoten übernimmt keinerlei Vermittlungsaufgaben, sondern dient höchstens der Signalregeneration. Passive Sternsysteme können mit z.B. TDMA-, CSMA/CD- oder Token-Zugangsverfahren betrieben werden.

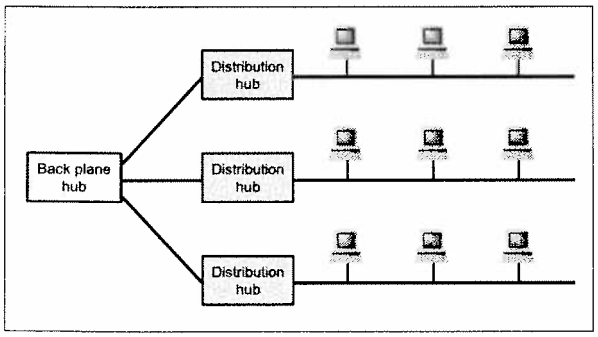


Bild: Physikalische Baum-Topologie

Baum-Topologie

Die strukturierte Verkabelung, die Anfang der neunziger Jahre in den Unternehmen aufkam, führte zwangsläufig zu einer hierarchischen Verkabelung. Diese hat die Gestalt einer Baum-Topologie. Bedingt durch die Konzentration der anfallenden Datenmengen zur Baumwurzel ist der Einsatz unterschiedlicher Technologien innerhalb des Netzes notwendig. Die Baum-Topologie ist dadurch gekennzeichnet, dass ausgehend von einer Baumwurzel eine Menge von Verzweigungen zu weiteren Knoten existiert, die bis auf die letzte Stufe wiederum die gleiche grundsätzliche Struktur mit weiteren Verzweigungen aufbauen.

Die Baum-Topologie ist wegen der strukturellen Äquivalenz ihrer Teile und der Rekursivität der Gesamtstruktur in der Informationsübertragung besonders beliebt. Es gibt sowohl logische (Spanning-Tree Algorithmus), als auch physikalische Baumstrukturen. Breitbandnetze früherer Tage nutzten einen zentralen Umsetzer[Verstärker, den man als Baumwurzel bezeichnen konnte. Heutige modular aufgebaute Netze, die aus einer strukturierten Verkabelung mit Hub-/Switch Hierarchie bestehen, haben ebenfalls eine Baumstruktur. Diese Struktur eignet sich gut für flächendeckende Verkabelung oder für Netze in mehrstöckigen Gebäuden. Die Baum-Topologie wird ebenfalls bei Breitbandnetzen (IEEE 802.4 und 802.3) verwendet.

Vermaschung

Eine vermaschte Struktur besteht im Grunde aus einer beliebigen Topologie, bei der die Knoten auf mehreren Wegen miteinander verbunden sind. Diese Struktur oder Topologie wird überwiegend in Weitverkehrsnetzen eingesetzt, um Überbelastungen abzufangen oder Leitungsunterbrechungen zwischen einzelnen Knoten durch alternative, redundante Wege zu kompensieren. Die Intelligenz der Vermittlung ist dabei in den Netzknoten angeordnet.

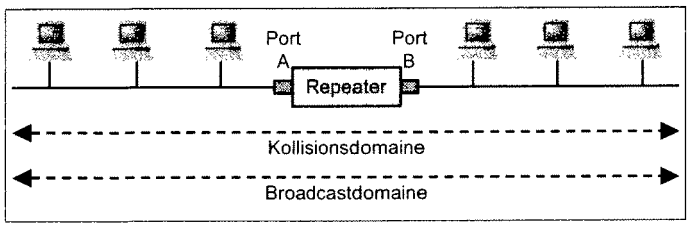
Aktive Netzkomponenten

Aktive Komponenten sind zur Kopplung einzelner LAN-Segmente oder LANs notwendig, um die Verkabelung und Topologien mit einer bestimmten Übertragungstechnologie bzw. Netztechnik auszustatten. Unterscheidungen sind dabei heute nicht mehr so einfach vorzunehmen, da die aktiven Komponenten immer mehr Funktionalität aufnehmen.

Im allgemeinen gilt die folgende Einteilung:

- Systeme der Bitübertragungsschicht: Repeater, Regeneratoren und Hubs.
- Systeme der Sicherungsschicht: Bridges und Switches.
- Systeme der Netzschicht: Router.
- Systeme höherer Schichten: Gateways.

Oftmals wird dabei nicht mehr zwischen Bridges und Routern unterschieden, da die Leistungsmerkmale von Bridges oft bereits in heutige Router integriert wurde. Denselben Trend merkt man an der unterlassenen Differenzierung von Routern und Gateways oder bei der Zusammenführung von Routern und Switches (Layer-3-Switching). Gateways besitzen aber grundsätzlich erweiterte Fähigkeiten und zeichnen sich durch die Protokollwandlung aus. Sie koppeln deshalb unterschiedliche Architekturen (z.B. IP und IPX) miteinander. Mit Layer-3-Switching ist hingegen eine neue Möglichkeit des Route-Once-Switch-Many gemeint, die in neuen Switches integriert ist. Aufgrund der Komplexität dieses Themas wird aber noch an anderer Stelle näher darauf eingegangen. Hier werden erst einmal die grundlegenden Definitionen erläutert und gegenübergestellt.



Repeater: Bitübertragungsschicht (Signalregeration)

Bild: Repeater

Repeater, Transcelver und Regeneratoren
Ein Repeater ist eine aktive Komponente, z.B. innerhalb eines Ethernet-LANs, die Regenerierungsfunktionen auf der Bitübertragungsschicht übernimmt. Ein Regenerator ist hingegen im WAN angesiedelt. Beiden gemeinsam ist die völlige Bit-Transparenz, d.h., sie sind für keine der angeschlossenen Stationen sichtbar.

Der Repeater ist im LAN für die Verbindung zweier Ethernet-Segmente zuständig, um das Netz auf größere Entfernungen auszudehnen und damit die Segmente verlängern zu können. Local Repeater verbinden zwei Kabelsegmente direkt miteinander. Neben der Verstärkung von Signalen zur Vergrößerung der Reichweite, dienen Repeater auch zur physikalischen Entkopplung von Netzsegmenten. Das heißt, wenn ein Segment aufgrund eines Kabelbruches ausfällt, arbeiten die anderen Segmente ohne Probleme weiter.

Local Repeater können eine maximale Entfernung von 100 m zwischen zwei Kabelsegmenten überbrücken. Wenn allerdings der Abstand zwischen zwei Segmenten eine Entfernung von 100 m überschreitet, werden Remote Repeater anstelle der lokalen Repeater eingesetzt. Der Repeater regeneriert den Signalverlauf sowie Pegel und Takt. Heutige Repeater verfügen über eine Testfunktion, die selbständig fehlerhafte Signale bzw. Kollisionen auf einem LAN-Segment erkennt. Bei einer Kollisionserkennung generiert der Repeater ein JAM-Signal. Dieses Störsignal wird beim Zugriffsverfahren CSMA/CD verwendet, um im Falle einer Kollision die Zeit der Kollisionserkennung möglichst kurz zu halten und das Netz schnell wieder für die Datenübertragung freizugeben. Fehlerbehaftete Signale werden dabei nicht auf das nächste Segment weitergeleitet. Dadurch kann man Fehler schnell lokalisieren.

Repeater können auch mit Zwischenpufferung arbeiten. In diesem Fall nennt man diese Systeme Buffered Repeater. Sie arbeiten im Gegensatz zum normalen Repeater auf der Sicherungsschicht. Der Buffered Repeater verarbeitet nur vollständige Rahmen und puffert diese nach dem Store-and-Forward-Prinzip ab, bevor er sie an das Netz überträgt. Neben den beschriebenen Repeatern gibt es außerdem noch den Multiport-Repeater, der mehrere Ausgänge unterstützt. Sie verbinden mehr als zwei Segmente miteinander.

Auch Fast-Ethernet enthält Repeater, die dazu dienen, die Verbindung von Fast-Ethernet Segmenten vorzunehmen. Diese Repeater werden in Repeater der Klasse I und der Klasse II unterteilt. Die Repeater der Klasse I haben Ports für unterschiedliche physikalische Medien (z.B. 100Base-TX nach 100Base-FX). Die Repeater der Klasse II unterstützen hingegen LAN-Segmente mit nur einem physikalischen Medium. Da die Repeater der Klasse I eine geringere Datenrate besitzen, als die Repeater der Klasse II, darf innerhalb einer Collision Domain (CD) nur ein Repeater der Klasse I eingesetzt werden. Man bezeichnet ein einzelnes Segment eines Ethernet-LANs als Collision Domain. Nach IEEE 802.3 befinden sich alle Endgeräte, die mit dem gleichen physikalischen Segment verbunden sind, an der gleichen Collision Domain. Repeater sind dabei nicht in der Lage, diesen Bereich zu separieren. Dies muss durch andere Systeme wie Brücken und Switches erfolgen, um die Collision Domaine möglichst klein zu halten. Zusätzlich spielt die Einschaltverzögerung eines Repeaters der Klasse 1 eine Rolle. Klasse I sieht 168 Bit pro Segment für die maximale Umlaufverzögerung vor, während Klasse II mit nur 92 Bit auskommt.

Um Netze weiter ausdehnen zu können, lassen Repeater sich mit Linksegmenten verbinden. An diese Linksegmente sind keine Rechner angeschlossen. Sie dienen ausschließlich als Verbindung von Bussegmenten, die sich in unterschiedlichen Gebäuden befinden und können Längen von bis zu 1 km annehmen. Aufgrund der endlichen Laufzeit der Signale ist die Ausdehnung eines Ethernets zwischen zwei beliebigen Stationen auf fünf Segmente mit zwei Linksegmenten begrenzt. Die Anzahl der Repeater zwischen zwei Stationen beträgt dabei höchstens vier.

Die Aus- und Einkopplung der Signale erfolgt über Transceiver. Dies kombiniert die Begriffe Transmitter (Sender) und Receiver (Empfänger) und bezeichnet eine Sende-/Empfangseinrichtung. Der Transceiver realisiert den Netzzugang einer Station und entspricht damit einer Medium Attachment Unit (MAU). In Richtung der Station wird das Access Unit Interface (AUI) angeboten, während in Richtung Übertragungsmedium das Medium Dependent Interface (MDI) sitzt. verbunden werden.

Transceiver sind ebenfalls in der Lage, Weiterleitungs-, Überwachungs-, Empfangs- und Störfunktionen (Jammer) auszuführen. Die Jammer-Funktion verhindert dabei, dass eine Station im Ethernet das Übertragungsmedium zu lange belegt. Diese Funktion ermöglicht einen Unterbrechungsmechanismus, mit dem eine MAU im Sendevorgang unterbrochen wird, wenn länger als 30 ms Daten gesendet wurden oder die definierte Rahmenlänge von 1518 Byte überschritten worden ist. Falls eine Unterbrechung vorgenommen wird, kommt es zur Sendung des Signal Quality Error (SQE) an das Endgerät, um der Station zu signalisieren, dass keine weiteren Daten gewünscht werden.

Aufgrund der Einführung der Glasfaser und der damit verbundenen optischen Signalverarbeitung sind heute ebenfalls optische Repeater im Einsatz. Der optische Repeater übernimmt hierbei die gleiche Funktion wie seine auf elektrischen Signalen basierenden Kollegen. Das elektrische Licht wird umgeformt und anschließend über eine LED oder Laserdiode in die Glasfaser eingespeist.

Der optische Regenerator dient wie der Repeater dazu, Signaldämpfungen des Lichts im WAN auszugleichen. Im Gegensatz zum Repeater enthält der Regenerator allerdings einen Entscheider. Dieser bewertet den Signalfluss. In Abhängigkeit des Schwellenwertes werden die empfangenen Lichtimpulse neu generiert. Unter mehrfachem Einsatz von Regeneratoren können Strecken unbegrenzter Länge überwunden werden. Die untere Abbildung zeigt das anliegende Eingangssignal an einem Repeater. Dieses wird über einen Entscheider regeneriert und verstärkt weitergeleitet.

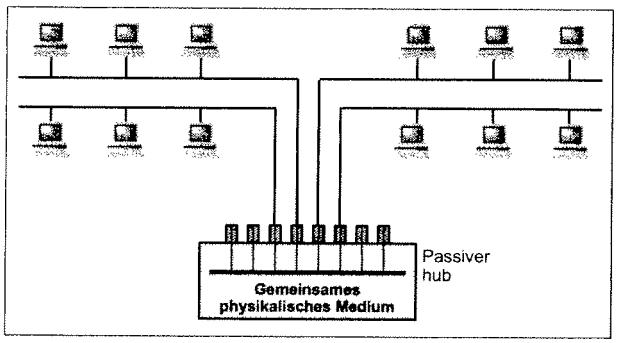
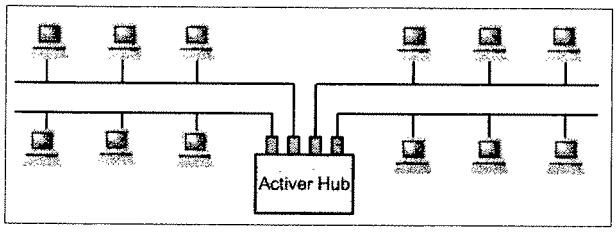


Bild: Passiver Hub

Hubs

Anders als Repeater und Regeneratoren dient der Hub nicht der Kopplung zweier Netzsegmente, sondern übernimmt die Verteilfunktion innerhalb einer Stern-Topologie. Das heißt, Hubs bilden in sich einen Bus oder Ring nach, wobei jeder Port direkt an die Stationen angeschlossen wird. Ein Hub ist dabei ein intelligentes Vermittlungssystem zwischen den Segmenten des LANs und den Endgeräten. Er stellt den Konzentrationspunkt für eine sternförmige Verkabelung dar und wird ebenfalls Sternkoppler genannt. Der Hub ist in der Lage, unterschiedliche LANs und beliebige Medien aneinander anzupassen.



Activer Hub: multi-port repeater

Bild: Aktiver Hub

Hubs können aufgrund ihrer Arbeitsgebiete unterschieden werden:

- Arbeitsgruppen Hub,
- Abteilungs-Hub,
- Unternehmens-Hub.

Die Hubs unterscheiden sich dabei in der Ausstattung und ihren Leistungsmerkmalen. Eine Unterteilung in Stackable und modular aufgebaute Hubs ist ebenfalls denkbar, die sich durch die Kosten voneinander abheben.

Der Arbeitsgruppen bzw. Workgroup Hub ist der kleinste Vertreter seiner Art. Man verwendet den Hub vorwiegend für den Anschluss von PCs oder anderen Endgeräten an den nächst höheren Abteilungs-Hub. Inzwischen sind diese Hubs als Stackable Hubs verfügbar. Bei Token Ring besitzt ein Hub über 8 bis 16 Twisted Pair (TP) Ports, während Ethernet 8 bis 24 feste UTP/STP POrts. Bei allen Stackable-Hubs werden die einzelnen Geräte über ein separates Kabel miteinander verbunden, wodurch sie nur als Repeater wirken. Dadurch können sehr viele Stationen angeschlossen werden, ohne dass Kaskadierungsprobleme auftreten. Man ist so in der Lage, bis zu zehn Hubs in einem Stack mit maximal 260 Ports zusammenzufassen. Die Hubs bzw. Ports können weiterhin in einzelne Segmente aufgeteilt werden, wodurch sich Engpässe von vornherein vermeiden lassen.

Abteilungs-Hubs unterstützen bis zu 100 Stationen und werden normalerweise mit anderen Hubs auf ihrer Ebene oder mit einem Unternehmens-Hub verbunden. Sie sind meistens modular aufgebaut, binden aber meistens die gleiche Technologie ein (z.B. nur Ethernet oder Token Ring). Unternehmens-Hubs können hingegen auch unterschiedliche LANs realisieren. Dadurch ist man in der Lage, unterschiedliche Technologien miteinander verbinden. je nach Portdichte werden einige hundert Stationen miteinander verbunden. Module sind in großer Zahl verfügbar, die als Konzentrator oder Managementsystem, aber auch als Bridge oder Router verwendbar sind.

Modulare Hubs bestehen weiterhin aus unterschiedlichen Bussystemen und Stromversorgungen. Der interne Bus wird dabei als Backplane bezeichnet. Neben der Slotanzahl für die Module bestimmt die Busarchitektur die Leistungsfähigkeit eines Hubs. Heutige Systeme arbeiten fast alle mit proprietären Bussen, wobei die Kapazitätszuordnung durch das jeweilige Zugriffsverfahren (z.B. CSMA/CD bei Ethernet) vorgenommen wird. Der Bus transportiert dabei die Datensignale wie auf einem eigenen Netz.

Vier grundsätzliche Arten einer Backplane werden unterschieden.
- Proprietärer segmentierter Bus,
- Proprietärer vielfacher Bus,
- Proprietärer gemultiplexter Bus,
- Systembus.

Die erste Möglichkeit ist in definierte Abschnitte für die Unterstützung von unterschiedlichen Technologien unterteilt. Das Modul auf dem segmentierten Bus merkt automatisch, ob ein Segment frei ist oder nicht. Wenn es nicht frei ist, ist das Modul in der Lage, auf ein anderes Segment auszuweichen oder über andere Module anderen Netzen beizutreten. Dadurch können Module für verschiedene Netztechnologien den gleichen Bus verwenden. Vielfache proprietäre Busse unterstützen hingegen nur einen Netztyp. Das heißt, es können nur zu diesem Netz passende Module eingesetzt werden. Der gemultiplexte Bus ist in mehrere virtuelle Busse aufgeteilt. Diese gehören dann jeweils zu einer Netztechnologie. Alle vorhandenen Module müssen sich an dieser virtuellen Umgebung orientieren. Systembusse haben ein einzelnes Modul, welches das Management des Busses übernimmt. Andere Module werden hierüber adressiert und beliebige Daten zugesandt. Es kann auch in manchen Fällen die Verwaltung von anderen Modulen übernommen werden, beispielsweise bei Ausfall des Kontrollmoduls.

Inzwischen ist sogar die Switching Technologie in Hubs integriert worden, wobei man in diesem Fall bereits von Switches sprechen muss. Durch den Switching-Ansatz entstehen neue Datenraten, die explizit dem Anwender zur Verfügung gestellt werden können. Die Grenzen zwischen den einzelnen Systemkomponenten verwischen deshalb zusehends.

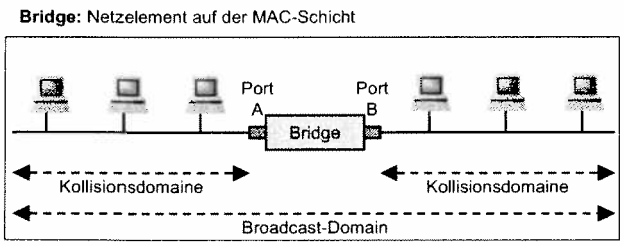


Bild: Bridge

Bridges

Bridges bzw. Brücken werden durch den Standard IEEE 802. 1) definiert und verbinden gemäß ihrer Definition Subnetze protokollmäßig auf der Schicht 2 (LLC, IEEE 802.2) oder 2a (MAC-Schicht) des OSI-Referenzmodells. Viele Bridges, speziell bei Ethernet, besitzen keine LLC-Funktionalität, sondern eine Verbindung auf der MAC-Schicht. Token Ring benötigt aber mehr Intelligenz und realisiert daher LLC-Funktionalität.

Die Hauptaufgabe einer Bridge lässt sich durch folgende zwei Punkte zusammenfassen:
- Segmentierung eines LANs: alle Segmente nutzen ein identisches Link-Layer Protokoll,
- Kopplung von LANs mit unterschiedlichen MAC-Protokollen der 802.x Protokollfamilie.

Durch den Einsatz von Bridges werden die Restriktionen des LANs für die maximale Segmentlänge und maximale Stationszahl umgangen, da jedes einzelne Subnetz die volle Stationszahl und Längenausdehnung erhalten kann. Vergleichsweise dazu stellen Repeater zwar ebenfalls eine Möglichkeit zur Verfügung, die Längenbeschränkungen eines einzelnen Segmentes zu überwinden, jedoch arbeiten sie nur auf der Ebene 1 des OSI-Referenzmodells. Ihre Aufgabe liegt ausschließlich in der Signalverstärkung.

Die Ausdehnung von LANs über deren maximale Länge (IEEE 802.3: 2,5 km bei Ethernet) stellt einen Grund für die Segmentierung dar, die den Einsatz von Bridges notwendig macht. Ein anderer Grund ist die Vermeidung von Kollisionen. Die verfügbare Datenrate in einem Shared-Medium Netz steht allen Stationen gleichermaßen zur Verfügung. Der Anschluss weiterer Stationen an das LAN, der steigende Einsatz verteilter Anwendungen und der damit weitere Anstieg von Kollisionen auf dem LAN, verringert den Datendurchsatz erheblich. Die Segmentierung des LANs durch Bridges ist eine Möglichkeit, die Anzahl der Stationen an einem Segment zu verringern und somit auch die Anzahl der Kollisionen. jedes Segment bildet dabei eine eigene CD. Zusätzlich ist die Lasttrennung für eine verbesserte Netzkapazität ausschlaggebend, da Bridges den lokalen Verkehr vom subnetzübergreifenden Verkehr trennen. Das heißt, wenn sich der Adressat im gleichen Subnetz wie der Sender befindet, dann verhindert die Brücke, dass der Datenrahmen in ein anderes Subnetz transportiert wird. Dies realisiert

sie, indem sie die Zieladresse mit der ihr bekannten Adresse vergleicht. Stimmt sie überein, dann wird es weitergeleitet, ansonsten bleibt es im lokalen Netz.

Die Aufgabe einer Bridge beschränkt sich somit bei der Segmentierung eines LANs auf das Filtern von den empfangenen Rahmen anhand der MAC-Adressen hinsichtlich des damit verbundenen Zielsegmentes. Nur die Daten, die tatsächlich für die Stationen an anderen Einheiten oder Segmenten bestimmt sind, werden von der Brücke übertragen. Allerdings können, sobald nicht zu entscheiden ist, ob es sich um lokalen oder übergreifenden Verkehr handelt, auch alle Rahmen transportiert werden. Außerdem werden auch alle Broadcast-Meldungen mit übertragen. Es handelt sich hierbei um Datenrahmen, die an alle Stationen geschickt werden. Dieses führt zu einer relativ hohen Grundlast. Brücken sind transparent bezüglich des Protokolls und unterstützen somit alle Protokolle in den LANs. Fehlerhafte Rahmen der Sicherungsschicht werden von der Brücke nicht weitergeleitet. Sie bleiben auf das Subnetz, in dem sie aufgetreten sind, begrenzt. Somit vermindern Brücken die Ausbreitung von Fehlern.

Beispielsweise wird ein Rahmen aus dem ersten Segment über die erste Bridge des zweiten Segmentes weitergeleitet, wenn keine Überbelastung vorliegt und der Rahmen nicht fehlerhaft war. Falls die Bridge den Rahmen nicht transportieren kann, kommt es zu einem Timeout. Die ursprüngliche Station hat keine Kenntnis über diese Überbelastung. Der Timeout wird deshalb nur als ein Nicht-Erreichen der Zielstation interpretiert.

Werden in den verschiedenen Segmenten auch noch unterschiedliche Link-Layer-Protokolle verwendet, können weitere Probleme auftauchen:

- Rahmenlänge,
- Prioritäten,
- Bitrate.

Die maximale Rahmenlänge (Maximum Transmission Unit, MTU) heutiger Protokolle unterscheidet sich grundsätzlich voneinander:

- IEEE 802.3 Ethernet-Standard: 1518 Byte,
- IEEE 802.5 Token-Ring: 4544 Byte,
- IEEE 802.4 Token Bus: 8191 Byte,
- FDDI: 4500 Byte,
- LAN Emulation (LANE) bei ATM: 1500-Byte, die sich aus 53-Byte-Zellen zusammensetzen,
- Fibre Channel (FC): 2148 Byte.

Da eine Fragmentierung auf dieser Schicht durch die IEEE Standards nicht vorgesehen ist, müssen beispielsweise beim Übergang Ethernet auf FDDI zu große Rahmen verworfen werden. Dies kann den Datendurchsatz und damit die Effektivität des Netzes stark verringern. Die Priorität wird bei Token Ring über den Token vergeben. Ethernet sieht keinen derartigen Mechanismus beim CSMA/CD-Verfahren vor. Zwar sind Prioritätsverfahren nach IEEE 802.1p standardisiert worden, diese haben aber nichts mit dem Token-Passing-Verfahren gemeinsam. Erschwerend kommt hinzu, dass der neue Standard wohl nur in Ethernet-Komponenten einfließen wird. Die Bridge ist hier aber in jedem Fall überfordert. Sie kann keine Prioritätsverfahren umsetzen. Ein weiteres Problem stellen die unterschiedlichen Bitraten der vorhandenen Netztechnologien dar. Dadurch kommt es trotz Einsatz von Switches zum Store-and-Forward-Betrieb. Zusätzlich könnten die Pufferspeicher einer Bridge überlaufen, so dass es zu Datenverlusten bei Überlast kommen kann. Zusammen mit den Timerwerten in den oberen Schichten können weitere Engpässe entstehen. Bei Versendung von langen Datenrahmen setzt der letzte Rahmen einen Request-Timer ein, um sich eine Bestätigung von der Gegenstelle zu besorgen, dass die Daten korrekt angekommen sind. Empfängt der Timer durch ein langsameres LAN zu spät einen Reply, nimmt die Netzschicht an, dass die gesamte Nachricht nicht oder nur teilweise angekommen ist und sendet erneut die Daten. Nach einer bestimmten Anzahl von erfolglosen Versuchen bricht sie die Übertragung völlig ab. Zusätzlich hat der ansteigende Datenverkehr das Netz weiter unnötig belastet. Aus diesem Grund werden Netze mit verschiedenen Link-Layer Protokollen meistens mit Routern auf der Schicht 3 miteinander verbunden.

Unabhängig von ihrem Einsatz zur Segmentierung des Netzes oder als Kopplung verschiedener Link-Layer Protokolle, kann man aufgrund der Funktionalität zwischen drei Arten von Bridges unterscheiden

- Transparente Bridges,
- Spanning-Tree Bridges,
- Source-Routing Bridges.

Die Transparente Bridge besitzt ihren Namen dadurch, dass sie für die Stationen der einzelnen Segmente nicht sichtbar ist. Das heißt, sie ist auch gegenüber den verwendeten Protokollen transparent. Weiterhin lernt sie selbständig, welche Station über welches Segment zu erreichen ist (Backward Learning). Sie kann somit ohne Konfiguration mit dem LAN verbunden werden. Die Segment- und Stationszuordnungen werden dabei in einer Hashtabelle festgehalten und regelmäßig aktualisiert. Bei Inbetriebnahme muss allerdings der Flooding-Algorithmus angewandt werden, da die Hashtabellen leer sind. Er lernt die Netzumgebung kennen und speichert diese in der Tabelle ab. In periodischen Abständen werden die Einträge überprüft und gegebenenfalls gelöscht, wenn diese älter als ein paar Minuten sind.

Ein weiterer Vorteil transparenter Bridges ist die Unterstützung redundanter Verbindungen. Dadurch ist die Kopplung von zwei Segmenten über mehrere Verbindungen möglich. Dies führt zu einer verminderten Störanfälligkeit. Das heißt, ein Ausfall einer Verbindung wird durch die zweite redundante Verbindung aufgefangen, ohne dass die Kommunikation zwischen Segmenten beeinflusst wird. Hier ergibt sich aber das Problem, dass Topologieschleifen durch Mehrfachübertragung eines Rahmens erzeugt werden könnten. Das heißt, durch die redundante Auslegung könnten die Bridges in Abständen das Netz fluten. Dies kann zu einer endlosen Schleife führen, da beide Bridges sich nicht gegenseitig abgleichen.

Die Lösung dieses Problems ist das Spanning-Tree-Verfahren, welches in IEEE 802.1 spezifiziert ist und transparenten Bridges erlaubt, redundante Strukturen einzuführen. Dieser Algorithmus verhindert im Grunde die Mehrfachübertragung eines Rahmens. Die Bridges kommunizieren untereinander, um die Überlagerungen der aktuellen Netztopologie mit einem überspannenden Baum zu versehen, der jedes LAN erreicht. Damit eine fiktive schleifenlose Technik aufgebaut werden kann, werden nicht alle Verbindungen innerhalb eines LANs verwendet. Wenn die Bridges sich gegenseitig übereinstimmend nach dem Spanning-Tree richten, erfolgen alle Datentransporte nach diesem Schema. Da von jedem Sender zu einem Empfänger nur eine Verbindung führt, können keine Schleifen entstehen. Um diese Konfiguration beizubehalten, sendet jede Bridge alle paar Sekunden ihre Seriennummer des Herstellers und alle ihr bekannten Bridges ins Netz. Die Basis bildet die Bridge mit der kleinsten Seriennummer. Anschließend werden Pfade über kürzeste Verbindungen zu weiteren Bridges aufgebaut. Wenn eine Bridge ausfällt bzw. aus dem Netz herausgenommen wird, fängt die Prozedur von neuem an. Der Baum wird dann nochmals neu berechnet. Das Spanning-Tree-Verfahren kann allerdings auch negative Mechanismen verursachen. Wenn man aus Sicherheitsgründen eine Firewall oder einen Server redundant an eine Bridge oder einen Switch anbinden möchte, schaltet Spanning-Tree den zweiten Pfad ab. Bei Ausfall des aktiven Pfades wird der redundante Pfad nicht mehr aktiviert. Die Verbindung ist unterbrochen. Um das zu verhindern, werden redundante Verbindungen an die gleichen Bridges bzw. Switches angeschlossen. Fällt allerdings die Bridge bzw. der Switch aus, kann keine redundante Sicherung der Daten eingeleitet werden. Heute sind Transparente Bridges mit Spanning-Tree-Verfahren hauptsächlich bei Ethernet im Einsatz. Token Bus und FDDI verwenden ebenfalls diese Art von Bridges. Sie erfordern kein Netzmanagement und besitzen eine einfache Konfiguration.

Die dritte Art von Bridges wird mit dem Begriff Source-Routing beschrieben. Anders als die Transparente Bridge erfordert dabei das Source-Routing einen erhöhten administrativen Aufwand. Das liegt daran, dass einer Station die gesamte Verbindung zu einer anderen Station eines beliebigen Segments bekannt sein muss. Die gesamte Leitweginformation wird im Header des zu übertragenden Rahmens festgehalten. Die Verwendung der Brücke ist damit nicht mehr transparent. Allerdings wurde ein gravierender Nachteil ausgeschaltet. Die Bandbreite kann durch Source-Routing besser ausgenutzt werden, da die ganze Topologie bekannt ist. Das Einsatzgebiet der Source-Routing Bridges ist heute nur noch der Token Ring.

Das Source-Routing-Verfahren basiert auf der Annahme, dass der Sender in einem LAN das Netz genau kennt und damit auch in der Lage ist, den Empfänger auszumachen. Wenn sich der Empfänger in einem anderen LAN befindet, wird das High-Order-Bit der Senderadresse auf 1 gesetzt. Zusätzlich wird der genaue Pfad angegeben, den der Rahmen zum Empfänger hin benötigt. Nur wenn das High-Order-Bit auf 1 gesetzt ist, greift das Source-Routing ein. Wird ein solcher Rahmen erkannt, wird ein Pfad nach einer bestimmten Route aufgebaut. Dies geschieht über die 12-Bit-Nummer eines LANs, die eindeutig ist und einem LAN zugeordnet wird. Enthalten sind 4 Bit für die Identifikation einer Bridge. Zwei weit auseinander liegende Bridges können die gleiche Nummer haben, während zwei Bridges im gleichen LAN sich dadurch eindeutig unterscheiden. Um den Pfad zum Empfänger zu ermitteln, wird ein Broadcast mittels eines Discovery Rahmen gesendet, der jedes LAN im Netzverbund erreicht. Wenn eine Antwort zurückkommt, fügen die Bridges ihre Kennung hinzu, wodurch der Sender den zurückgelegten Weg sehen kann und in der Lage ist, die beste Route zu ermitteln. Dieser Algorithmus verursacht damit einen exponentiellen Anstieg der Rahmen, der sich auf das gesamte Netz auswirkt. Transparente Bridges arbeiten nach einem ähnlichen Prozess, überfluten aber nur entlang eines überspannenden Baums, so dass die absolute Zahl der übertragenen Rahmen nur linear mit der Netzgröße ansteigt.

Source-Routing bietet trotzdem optimales Routing an und eignet sich sehr gut zum Einsatz parallel betriebener Bridges. Störungen werden allerdings nicht effektiv wahrgenommen. Das heißt, es wird nur gemerkt, dass die Rahmen nicht mehr bestätigt werden. Nach einem Timeout wird wieder versucht, das Ziel zu erreichen. Erst nach längerer Erfolglosigkeit wird die Taktik geändert. Die Bridge schickt einen weiteren Discovery Rahmen, der testen soll, ob das Ziel oder der Weg Probleme bereitet. Wenn eine wichtige Bridge ausgefallen ist, müssen viele Hosts den Ablauf von Timeouts abwarten und senden ebenfalls neue Discovery Rahmen aus. Dies geschieht selbst dann, wenn alternative Wege bereitstehen würden.

Damit ein Rahmen von der Bridge transportiert werden kann, muss die Bridge wissen, auf welcher Seite sich der Empfänger mit der zugehörigen MAC-Adresse befindet und über welchen Port er zu erreichen ist. Dazu sind die Bridges mit einem selbst lernenden Algorithmus und einer dynamischen Adresstabelle ausgestattet. Empfängt die Brücke an einem Port ein Rahmen, so wird zuerst die MAC-Quellenadresse des Rahmens ausgewertet und mit den Einträgen in der Adresstabelle verglichen. Ist diese Quellenadresse unbekannt, wird sie mit der entsprechenden Portnummer gespeichert. Dadurch kann ein bestimmter Sender mit einem bestimmten Port in Verbindung gebracht werden.

Anschließend findet die Transportentscheidung statt. Sie erfolgt aufgrund der im Rahmen enthaltenen MAC-Zieladresse, wodurch sich drei mögliche Wege ergeben:

- Die Zieladresse wird mit dem gleichen Port assoziiert, auf dem der Rahmen angekommen ist. Das Fazit lautet, dass es sich um lokalen Verkehr handelt und der Rahmen verworfen werden kann.
- Die Zieladresse wird durch den Eintrag in der Adresstabelle mit einem anderen Port assoziiert und der Rahmen auf diesem weitergeleitet.
- In der Adresstabelle ist kein Eintrag mit der angegebenen Zieladresse vorhanden. Ein Broadcast wird über alle aktiven Ports der Bridge initiiert, da das Subnetz des Zielrechners nicht bekannt ist.

Ein entscheidendes Kriterium für die Leistungsfähigkeit einer Bridge, ist demzufolge die Zeit, die zum Durchsuchen der Adresstabelle und zum Fällen der Transportentscheidung benötigt wird. Weitere Punkte sind Datendurchsatz und Überlastverhalten.

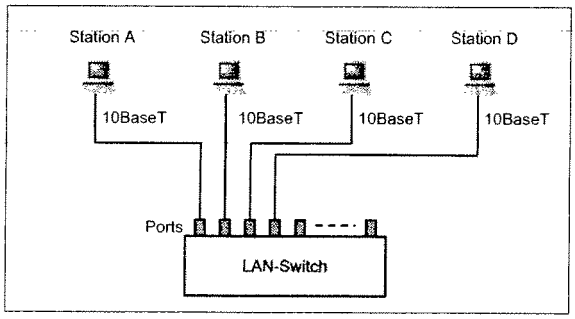


Bild: Switched LAN

Eine Segmentierung von LANs auf Basis von Bridges führt zu einer verbesserten Ausnutzung der Bandbreite des gesamten LANs. Allerdings kann die Datenrate nicht innerhalb der Segmente gesteigert werden. Zusätzlich beinhalten Bridges höhere Verzögerungen und geringeren Datendurchsatz.

Um jeder Station innerhalb eines LANs die volle Bandbreite zur Verfügung zu stellen und Latenzzeiten gering zu halten, wurden Switches entwickelt.

Switch: multi-port Bridge

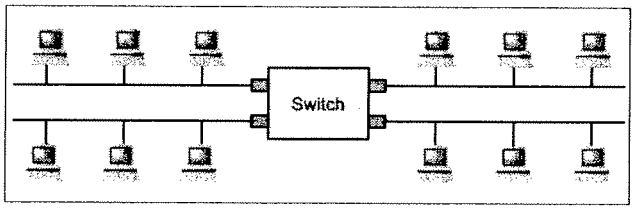


Bild: Switch

Switches sind wie Bridges Kopplungselemente der Sicherungsschicht und ermöglichen dadurch ebenfalls die Segmentierung eines LANs. Durch die Kaskadierung mehrerer Switches, als Ersatz für verwendete Hubs, ist eine Segmentierung bis hin zur direkten Kopplung einzelner Stationen über dedizierte Verbindungen möglich.

Um im Backbone gleichzeitig verschiedene aktive Verbindung zwischen zwei Switches zur Verfügung zu stellen und eine dynamische Lastverteilung abhängig vom Verkehr und dem. Übertragungsmuster zu gewährleisten, ist das Switch Meshing von einigen Herstellern eingefügt worden. Dadurch wird durch die integrierte Redundanz eine höhere Effizienz erreicht, als beispielsweise durch das Spanning Tree Verfahren. Das Spanning Tree Protocol (STP) schaltet alle redundanten Wege ab. Erst wenn eine Verbindung ausfällt, wird diese durch einen anderen Pfad ersetzt. Der Ansatz des Switch Meshing setzt hingegen alle existierende Verknüpfungen für den Datentransport ein. Dabei wird der Verkehr auf alle Pfade verteilt, indem dynamisch die entstehenden Kosten ermittelt werden und daraus der günstigste Weg berechnet wird. Dadurch kann ebenfalls eine Verdoppelung der Bandbreite zwischen den Switches umgesetzt werden. Ausfallende Verbindungen werden in ca. einer Sekunde durch andere ersetzt.

Bei der Grundstruktur vermaschter Switches wird mindestens ein Port für die Vermaschung konfiguriert, der wiederum mit einem vermaschten Switch verbunden sein muss. Es ist ebenfalls möglich mit mehreren vermaschten Switches in einer Meshing Domain angeordnet zu sein, die auch mit weiteren Switches außerhalb dieser Domäne verbunden sind. Die Meshing Domain besteht dabei aus einer Gruppe von geswitchten Ports, die über das Meshing-Protokoll Rahmen austauschen. Die Pfade zwischen den Switches können vielfach redundant ausgelegt sein, ohne Broadcast-Bursts zu verursachen. Vermaschte Verbindungen gehören zu den Punkt-zu-PunktVerbindungen. Alle vermaschten Ports eines Switches sind dabei in der selben Switch Meshing Domain angeordnet. Alle Ports müssen allerdings nicht dazugehören. Die Anordnung der Switch Meshing Domain verspricht folgende Vorteile gegenüber herkömmlichen Routing-Protokollen:

- Dynamische Werte werden bei Auswahl der Verbindung zugrunde gelegt, um die günstigste Verbindung zu erhalten. Herkömmliches Routing (OSPF oder RIP) verwendet nur konstante Größen, wie Kosten oder Anzahl der Router-Hops.
- Layer-3-Protokolle können genauso verwendet werden, wie nicht-routbare Protokolle.
- Konfiguration ist relativ einfach, da nur die Ports definiert werden müssen, die zu einer vermaschten Domäne gehören. Alles andere wird durch den Switch umgesetzt.
- Reaktionszeiten sind extrem gering und liegen bei ca. einer Sekunde.
- Verzögerungszeiten sind geringer gegenüber einem Router. Da die Rahmen nicht geändert werden, kommen keine Netzverzögerungen hinzu.

LAN Aggregation wird ebenfalls oft mit Switch-Vermaschung (Switch-Meshing) verglichen, da hier ebenfalls eine Kapazitätssteigerung und Redundanz im Netz erreicht werden kann. Hier sind mehrere Verbindungen gleichzeitig aktiv, so dass sich Schleifen bilden können. Zusätzlich ist keine dynamische Lastverteilung möglich, da dies statisch vorgenommen wird. Man darf sie nicht mit der Link Aggregation verwechseln, die zur Steigerung der Datenrate bei Punkt-zu-Punkt-Verbindungen eingesetzt wird. Das Port Trunking setzt praktisch auf dieser Technik auf, indem mehrere Ports als einzelner, schneller Port interpretiert werden. Das Ziel des Port Trunking ist dabei ausschließlich die Reduzierung des Verwaltungsaufwands durch mehrere Adapterkarten in einem Server. Trotz mehrere NICs bekommt nämlich der Server nur einen Namen und eine IP-Adresse zugewiesen. Zur Lastverteilung der parallel laufenden Verbindungen werden Einwege- und Zweiwege-Lastverteilung eingesetzt. Die erste Methode ist die einfachere, da keine Änderung am Switch erforderlich ist. Bei der Zweiwege-Methode müssen die Switches einbezogen werden, da spezielle Link Aggregation Funktionalität geleistet werden muss. Dafür ist eine hohe Transparenz bzw. einfache Verwaltung und bidirektionale Lastverteilung vorhanden.

Bei ausgereifter Technik besitzt ATM gegenüber der Technologie Gigabit-Ethernet Vorzüge, da hier die Bandbreite skaliert werden kann und eine garantierte Dienstgüte vergeben wird. Die fehlende Skallerbarkeit bei Gigabit-Ethernet führt schnell zu hohen Auslastungen, da in Zukunft auch Gigabit-Ethernet Adapterkarten in Servern Verwendung finden werden. Diese belasten dann das Netz mit der maximal möglichen Datenrate. Das heißt, die Endstationen können nicht in der Datenrate begrenzt werden. Trotzdem stellt Gigabit-Ethernet eine interessante Möglichkeit dar, da zwar das Netz bei bereits 40 % im Shared Medium Einsatz ausgelastet sein könnte (wie bei Fast-Ethernet), aber dadurch immer noch 400 Mbit/s für multimediale Anwendungen zur Verfügung stehen. Die beschriebenen Zusatzmechanismen könnten sogar für eine effizientere Auslastung sorgen.

3.1b Internet-Referenzmodell: Ethernet-Standards

Version: August 2003

Inhalt

- **Ethernet und IEEE 802.3**
  10 Mbit/s über koaxialkabel
- **Fast Ethernet**
  100 Mbit/s Ethernet
  100 Mbit/s über Twisted-Pair Kabel
- **Gigabit Ethernet (GbE)**
  1 Gbit/s Ethernet
  1 Gbit/s über Glasfaser und Twisted-Pair Kabel
- **10 Gigabit Ethernet (10 GbE)**
  10 Gbit/s über Glasfaser

Bild: Ethernet, Fast Ethernet, Gigabit Ethernet

Ethernet-version	Übertragungs-rate	Codierung	Verwendetes Kabel	Full-Duplex Operation
10Base - 2	10 Mbit/s	Manchester	Koaxialkabel 50 Ohm	nicht möglich
10Base - T	10 Mbit/s	4B/5B	Twisted-Pair	unterstützt
100Base - T4	100 Mbit/s	8B/6T	Twisted -Pair	nicht möglich
100Base - FX	100 Mbit/s	4B/5B	Glasfaser	unterstützt
1000Base -T	1.000 Mbit/s	PAM5	Twisted-Pair	unterstützt
10GBase-SR	10.000 Mbit/s	64B/66B	Glasfaser	unterstützt

Bild: Ethernet-Systeme

Ethernet (IEEE 802.3)

Anfang der siebziger Jahre entwickelte man im Palo Alto Research Center (PARC) der Xerox Corporation ein Protokoll, welches, aufgrund eines gemeinsam genutzten Übertragungsmediums, das Senderecht nach Bedarf an die angeschlossenen Stationen verteilte. Es entstand das CSMA/CD-Verfahren, welches dem heutigen Ethernet zugrunde liegt. Das Team unter der Leitung von Dr. Robert Metcalf war damals bestrebt, ca. 100 Computer über ein 1 km langes Kabel miteinander zu verbinden. Die erste Realisierung hatte nur eine Übertragungsrate von 2,94 Mbit/s in Basisbandtechnik. Dabei schickte eine Station ihr Datenrahmen, sobald es zur Übertragung bereit war, auf das gemeinsame Übertragungsmedium in Richtung Empfänger. Während der Übertragung wurde überprüft, ob das Medium frei von Kollisionen war. Das heißt, das Medium wurde nach sendenden Stationen abgehört. Wenn eine Kollision durch zwei gleichzeitig sendende Stationen registriert wurde, musste der Sender die Übertragung nach einer Zufallszeit wiederholen. Das Ethernet-Konzept wurde später von der Firmengruppe DEC, Intel und Xerox (DIX) zum heutigen Ethernet-Standard mit einer Datenrate von 10 Mbit/s weiterentwickelt. 1983 wurde diese Spezifikation bei der IEEE als Standard vorgeschlagen, von IEEE verbessert und als Norm in 802.3 aufgenommen. Gleichzeitig veröffentlichte die DIX-Gruppe die Spezifikation für Ethernet der Version 2. Die Arbeiten am Cheapernet Standard wurden aufgenommen. Durch die unterschiedliche Weiterentwicklung von Ethernet verhalten sich DIX-Ethernet und IEEE-Ethernet nicht konform miteinander. Besonders in der Vergangenheit hat das häufig zu Problemen geführt. Heute wird im Grunde nur noch der Ethernet-Standard nach IEEE 802.3 verwendet.

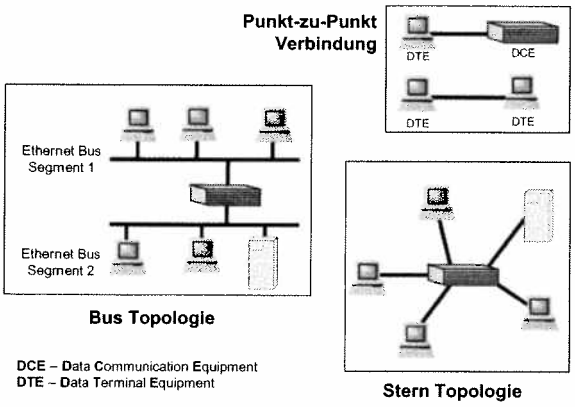


Bild: Ethernet Topologien und Strukturen

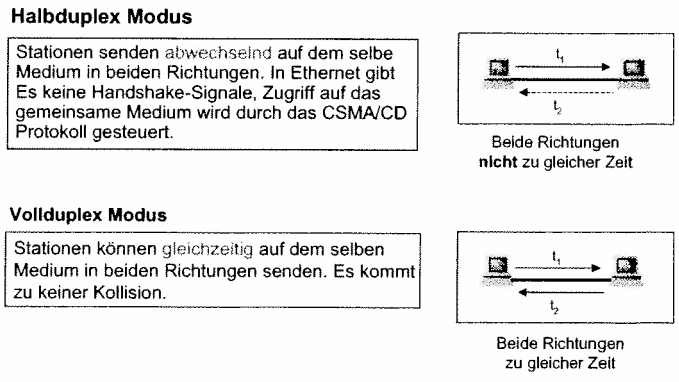


Bild: Halbduplex- und Vollduplex-Betrieb

Ethernet IEEE 802.3 Systeme

Im Standard IEEE 802.3 werden die meisten Rahmenbedingungen für Ethernet-LANs vorgegeben. Die maximale Datenübertragungsrate dieses Netzes lag anfangs bei 10 Mbit/s (Megabit pro Sekunde). Durch moderne Übertragungskomponenten und Verkabelungssysteme konnte eine Steigerung auf 100 Mbit/s erreicht werden. Dies wurde in IEEE 802.3u geregelt. Während im SOHO-Bereich (Small Office, Home Office) noch oft Netze mit 10 Mbit/s in Verwendung sind, werden im kommerziellen Unternehmensbereich bereits fast immer 100 Mbit/s-Netze eingesetzt.

Letzter Stand der Entwicklung ist eine weitere Steigerung auf 1 Gbit/s. Dies ist in IEEE 802.3z festgelegt. Bei Glasfaser gibt es bereits schon länger derartige Anwendung, während die Datenübertragung von 1 Gbit/s über Kupfer bezüglich der Verkabelungskomponenten erst Ende 1999 genormt wurde (Category 5e). Die Datenübertragungsrate von 10 Gbit/s ist heute aktuell.

Das unter IEEE 802.3 verwendete Protokoll ist CSMA/CD (Carrier Sense Multiple Access - Collision Detection). Dies spiegelt die Eigenschaften der ersten IEEE 802.3 Netze wider - ein Gerät, das Daten senden will, prüft vor dem Senden, ob die Datenleitung auch wirklich frei ist oder von anderen Geräten sich gerade Daten auf der Leitung befinden, bzw. Kollisionen werden von allen Geräten erkannt. Durch die moderne Switching-Technologie und Full duplex -Datenübertragung ist diese Charakteristik allerdings fast bereits überholt. Ab Gigabit werden überhaupt nur mehr Switches eingesetzt und die Plug-and-Play-Philisophie wird perfekt verwirklicht. Bei der letzten Gerätegeneration können Gigabit-Ports bereits mit allen Geschwindigkeiten (10/100/1000 Mbit/s) betrieben werden (sind also voll abwärtskompatibel), bzw. erkennen auch selbständig ob sie an einen anderen Switch oder an ein Endgerät angeschlossen sind.

Abhängig von den verwendeten Übertragungsmedien bzw. den erzielbaren Datenübertragungsraten spricht man auch von Thick-Wire-Ethernet (10Base-5), Thin-Wire-Ethernet (10Base-2), Twisted-Pair-Ethernet (10Base-T) sowie Fast-Ethernet (100Base-TX) und Gigabit-Ethernet (1000Base-T).

Bei der optische Datenübertragung (Fiber-optic, Glasfaser) unterscheidet man derzeit zwischen 10Base-FL (10 Mbit/s), 100Base-FX (100 Mbit/s), 1000Base-SX (1000 Mbit/s, short wave) und 1000Base-LX (1000 Mbit/s, long wave). Ein früherer Standard war auch 10Base-FOIRL (10 Mbit/s, Multimode bis 1 km). Dieser ist aber kaum mehr in Verwendung und hat somit keine praktische Bedeutung mehr.

Die verschiedenen Systeme können mittels spezieller Komponenten miteinander kombiniert werden (Transceiver, Repeater, Hub, Switch, Konverter).

Funktionsweise von Standard-Ethernet

Das Ur-Ethernet besitzt als logische und physikalische Topologie eine Busstruktur, die alle Stationen über ein gemeinsames Medium miteinander verbindet. Das heißt, alle Endstationen müssen sich den Übertragungskanal teilen (Shared Medium), da sie im gegenseitigen Wettbewerb zueinander stehen. Durch diese Situation ist ein Protokoll erforderlich, welches gewährleistet, dass immer nur eine Station sendet. Das dafür notwendige Zugriffsverfahren heißt Carrier Sense Multiple Access with Collision Detection (CSMA/CD) und ist in der Lage, einen Mehrfachzugriff durch Trägererkennung bei gleichzeitiger Kollisionserkennung durchzuführen.

- Die MAC-Subschicht regelt den Medienzugriff und die vom Zugriffsverfahren abhängige Block- oder Rahmenbildung
- Datenendgerät (Data Terminal Equipment, DTE) ist jedes an einem lokalen Netz angeschlossene Station oder Netzkomponente, die mit einer MAC-Funktion ausgestattet ist.
- Die für das Absenden eines Rahmens minimaler Länge benötigte Zeit wird als Slot Time bezeichnet.

$$Slot\ Time = \frac{Minimale\ Rahmenlänge}{Übertragungsrate}$$

Übertragungsrate	Minimale Rahmenlänge	Slot Time
10 Mbit/s	512 bit	51.2 µs
100 Mbit/s	512 bit	5.12 µs
1 Gbit/s	4096 bit	4.096 µs

Dabei lassen sich auch noch verschiedene Varianten des CSMA/CD-Protokolls unterscheiden. Diese Protokolle gehören zu den Random-Access-Verfahren, bei denen alle Stationen eines Ethernets im Prinzip zu jedem Zeitpunkt auf das Übertragungsmedium zugreifen können, um eine Datenrahmen zu senden. Die Einschränkung ist, dass eine Station nur dann senden darf, wenn das Medium auch wirklich frei ist. Die sendewillige Station muss deshalb zuerst den Übertragungskanal abhören, um Kollisionen zwischen verschiedenen Nachrichten zu vermeiden.

Bild: Medium Access Control (MAC)

Wenn eine Station ein freies Medium vorfindet, wird sofort mit der Übertragung begonnen. Während der Übertragung wird das Medium weiter abgehört, um auftretende Kollisionen sofort zu erkennen. Dies geschieht dadurch, dass die Station auf dem Medium etwas anderes hört, als das, was sie ursprünglich gesendet hat. Wird eine Kollision festgestellt, bricht die Station die Übertragung sofort ab und sendet ein Störsignal, weiches auch als JAM-Signal bekannt ist. Alle Stationen bemerken nun, dass eine Kollision stattgefunden hat und warten mit der Übertragung. Der sofortige Abbruch im Falle einer Kollision und das Senden des JAM-Signals verkürzen die überflüssige Zeit (weil die Daten nicht mehr korrekt ankommen werden) auf die Zeit der Kollisionserkennung. Bei dem JAM-Signal handelt es sich um ein 4-6 Byte langes Bitmuster, das 16 mal Bitkombinationen 1-0 beinhaltet. Die Länge des Störsignals liegt somit weit unter der Länge des kleinsten Ethernet-Rahmens von 64 Byte, wodurch dieses Signal sofort von allen erkannt wird. Nach einem zufälligen Zeitpunkt startet die Station erneut einen Sendeversuch.

- **Carrier Sense**
bedeutet, dass eine sendewillige Station das Medium erst abhört, ob schon Datenverkehr auf ihm abläuft (listen before talking).

- **Multiple Access**
bedeutet, dass alle Stationen gleichberechtigt auf das Übertragungsmedium jederzeit zugreifen können.

- **Collision Detect**
Falls zwei Stationen in einer Kollisionsdomäne gleichzeitig senden, wird der Sendevorgang sofort abgebrochen.
Nach einer durch den Backoff-Algorithmus bestimmten Wartezeit versuchen die Stationen ihre Rahmen erneut zu senden.

CSMA/CD: Carrier Sense - Multiple Access / Collision Detect

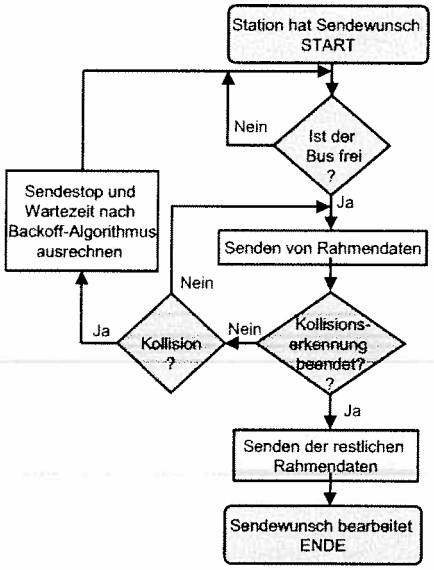


Bild: Zugriffsverfahren CSMA/CD

- Kollision tritt nur im Ethernet mit Halbduplex-Verfahren ein.
- Slot Time ist die Länge des kleinsten Rahmens (64 Byte) im Medium (512 Bitzeiten).
- Kollisionserkennung wird nur während der Übertragung der ersten 576 Bitzeiten durchgeführt (ergibt sich aus dem kleinstmöglichen Frame von 64 Byte = 512 Bit, plus einer Sperrzeit für die Kollisionserkennung).
- Daraus ergibt sich die maximale Ausdehnung einer Kollisionsdomäne (aus der halben Signallaufzeit der kleinsten Rahmengröße).
- Die erkannten Kollisionen werden durch ein Störsignal (JAM-Signal) anderen Stationen mitgeteilt.
- Das JAM-Signal besteht aus einer 32 Bit langen Folge von 1 und 0.

Bild: JAM-Signal

Falls:
- die maximale Ausdehnung des Netzes zu groß ist oder
- über Repeater/Hubs zu viele Netzsegmente gekoppelt wurden,

kann es zu einer nicht erkannten Kollision kommen.

- Solche Kollisionen nennt man „Late Collisions" und können nur von höheren Protokollebenen (z.B. Schicht 4 eines verbindungsorientierten Protokolls) erkannt und korrigiert werden.

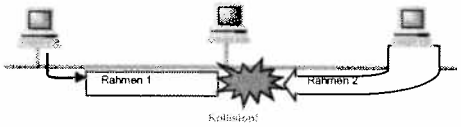


Bild: Late Collisions

Es soll vermieden werden, dass die Stationen nach dem Auftreten einer Kollision wieder gleichzeitig versuchen, ihre Rahmen auszusenden.

- Jede Station in einer Kollisionsdomäne ermittelt die ganze Zahl r nach dem Zufallsprinzip innerhalb folgendes Wertbereichs:

$$0 \leq r < 2^k ,$$

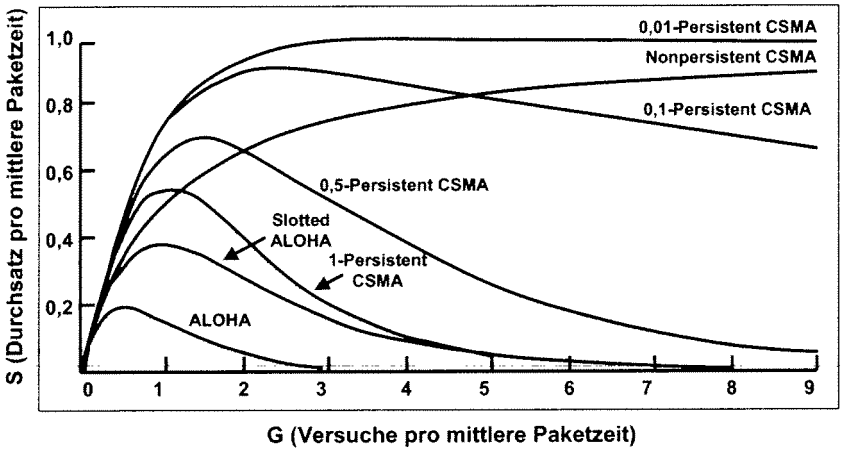
wobei $k = 1, 2, .., n$. (n stellt die Anzahl der Wiederholungsversuche dar).
- Die Wartezeit wird dann als $w = r \cdot (Slot\ Time)$ ermittelt.
- Falls es wieder zur Kollision kommt wird die Variable n um eins erhöht.
- Nach dem zehnten gescheiterten Versuch bleibt die Variable k konstant ($k = 10$).
- Bei $k = 16$ wird abgebrochen.

Bild: Backoff-Algorithmus

Diese zufällige Zeitspanne wird nach dem sogenannten Binary Exponential Backoff (BEB) Algorithmus ermittelt, der in jeder Station gleichzeitig nach Verbindungsabbruch gestartet wird.

CSMA/CD-Protokolle unterscheidet man 1-persistent, non-persistent und p-persistent. Die beiden ersten Protokolle differenzieren sich in der Redaktion auf ein belegt vorgefundenes Medium. Das Verfahren 1-persistent wartet solange, bis der Übertragungskanal wieder frei ist und versucht anschließend zu senden. Im Gegensatz dazu sendet eine Station beim non-persistent Verfahren nicht sofort. Es wird hier so verfahren, als ob eine Kollision aufgetreten wäre. Anschließend wird bei besetztem Medium eine zufällige Zeitspanne (BEB) abgewartet, bis der nächste Sendeversuch gestartet wird.

Eine Abwandlung des 1-persistent CSMA/CD ist das Protokoll p-persistent. Hier sendet die Station bei freiem Medium nicht unbedingt sofort, sondern nur mit der Wahrscheinlichkeit p. Vorteil dieses Verfahrens ist, dass versucht wird, die Wartezeit so kurz wie möglich zu halten. Nachteile entstehen bei Auftreten stärkeren Verkehrs. Dann ist die Wahrscheinlichkeit größer, dass mehr als eine Station während einer andauernden Sendung sendebereit wird und diese mit den bereits wartenden kollidiert. Das non-persistent Verfahren löst dieses Problem, indem die Übertragungsversuche aller Stationen, die während einer Sendung sendebereit werden, auf einen späteren jeweils unterschiedlichen Zeitpunkt verschoben werden. Im Standard IEEE 802.3 ist das 1-persistent CSMA/CD-Verfahren festgelegt worden.



Das Bild zeigt den Durchsatz gegenüber dem Verkehrsangebot der verschiedenen CSMA Protokolle sowie des reinen und getakteten ALOHA.

Bild: Zufallsgesteuerte Protokolle: Durchsatz

Hat erst einmal eine Kollision stattgefunden, ist es von entscheidender Wichtigkeit, dass die Stationen, die an diesem Konflikt beteiligt waren, den nächsten Übertragungsversuch nicht auch wieder zum selben Zeitpunkt einleiten. Ansonsten könnte es sehr schnell zu einer Synchronisation der Sender kommen. Das Binary Exponential Backoff (BEB) verteilt die Zugriffe der Sender auf das Medium über die Zeit. Der BEB-Algorithmus wird nach einer Kollision gestartet und es wird eine Zeitspanne nach dem Random-Access-Verfahren berechnet, nach der die Übertragung wiederholt werden darf. Die Wartezeit steigt dabei nicht weiter an, wenn ein Übertragungsversuch beim zehnten Mal durch eine Kollision beendet wird. Beim sechzehnten missglückten Versuch wird die Übertragungsanforderung abgewiesen und mit einer Fehlermeldung abgebrochen.

Möglicher Durchsatz von CSMA/CD

Nachdem das Zugriffsverfahren CSMA/CD beschrieben wurde, ist auch die Effizienz dieses Verfahrens wichtig, da es auch für Gigabit-Ethernet in Shared Medium Umgebung eingesetzt wird. Das CSMA/CD-Verfahren besitzt den Vorteil, dezentral zu funktionieren. Dies beinhaltet aber auch gleichzeitig einen Nachteil: die Signallaufzeiten unterscheiden sich. Deshalb musste man eine maximale Signallaufzeit T zwischen zwei beliebigen Stationen des gleichen Netzes festlegen, die das Signal zur physikalischen Ausbreitung auf dem gesamten Medium höchstens benötigen darf. Sie ist somit abhängig von den Übertragungseigenschaften sämtlicher Bestandteile des Signalwegs sowie seiner Länge. Für die Signallaufzeit T wurde der Wert 25,6 µs festgelegt, wodurch die Round Trip Delay (RTD) bei Ethernet 51,2 µs beträgt. Sie dient zur Kollisionsauflösung im BEB-Algorithmus.

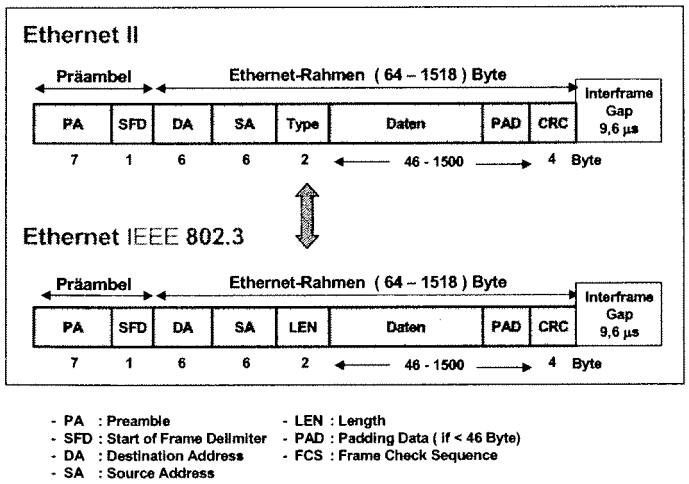


Bild: Rahmenformate in Ethernet

- Präambel: Eine Folge von 1-0-1-0-1-0-1-0 (7 Bytes), die von den mitlesenden Stationen zum Aufsynchronisieren verwendet werden kann.
- Start Frame Delimiter (SDF): Ein 1-0-1-0-1-0-1-1-Byte, welches das Ende der Präambel und den Anfang des eigentlichen Rahmens kennzeichnet.
- Zieladresse: Eine eindeutige, weltweit einmalige Hardware-Adresse vom Empfänger, die jeder Ethernet-Karte eingebrannt wurde.
- Quelladresse: Die Hardware-Adresse des Senders.

Bild: Bestandteile eines Rahmens

Rahmenaufbau

Aufgrund der verschiedenen Entwicklungen, die sich bei Ethernet parallel ereignet haben - DIX, (DEC, Intel und Xerox) und IEEE - sind zwei Rahmenformate gültig, die als Ethernet II Rahmen und IEEE 802.3 Rahmen bezeichnet werden. Der Ethernet II Rahmen ist heute nur noch geringfügig vertreten.

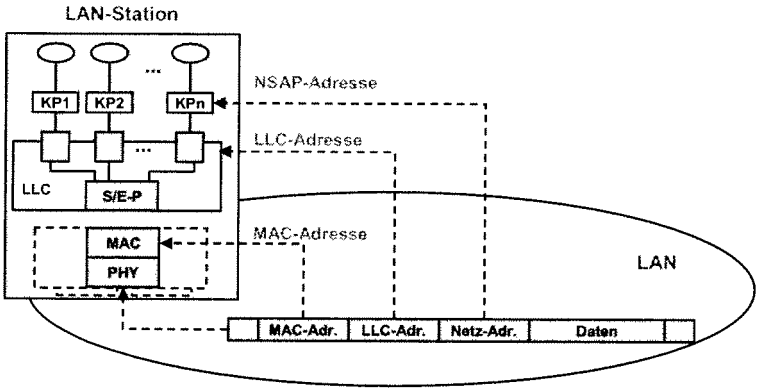
Unterschiede betreffen die Quell- und Zieladresse, die zusätzlich auch 2-Byte-Werte annehmen kann, das Address Resolution Protocol (ARP), welches die IP-Adresse (32 Bit) in die IEEE 802.3-Adresse (48 Bit) umwandelt und das Längenfeld, das die Einheitslänge für die Daten festlegt. Im IEEE-802-Standard, der sich mit der Spezifikation der unteren beiden Schichten des

OSI Referenzmodells befasst, wird die Sicherungsschicht in zwei weitere Teilschichten, die Medium Access Control (MAC) und Logical Link Control (LLC), aufgeteilt. Die MAC-Schicht beinhaltet die Zugriffssteuerung für den Übertragungskanal, während die LI-C-Schicht für einheitliche Verbindungen zwischen den unterschiedlichen MAC-Teilschichten und der Vermittlungsschicht zuständig ist. Dies wird durch den IEEE-802.3-Rahmenaufbau wiedergegeben.

- Längen-/Typ-Feld: Nach 802.3 wird in dieses Feld die Länge der eigentlichen Information geschrieben. Die Ethernet-Spezifikation sieht hier ein Typfeld, welches das darüberliegende Protokoll spezifiziert, vor. Ein Kompromiss ist die Verwendung der höchstwertigen fünf Bits als Typfeld und der verbleibenden Bits als Längenfeld.
- Datenfeld: hier werden die eigentlichen Nutzinformationen übertragen. Die Länge ist auf 1500 Byte beschränkt. Sollte die Nutzinformation eine Länge von 46 Byte unterschreiten, müssen gegebenenfalls Füllzeichen eingefügt werden.
- Cyclic Redundancy Check (CRC): Entspricht einer Quersumme für die Fehlerkorrektur. Gebildet wird sie aus dem Generator-Polynom: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$.

Bild: Bestandteile eines Rahmens

Der Ethernet Rahmen beginnt mit einer Präambel aus 10-Folgen. Sie ist 7 Byte lang und erzeugt durch die Manchester-Codierung für die Dauer von 5,6 µs eine 10-MHz-Schwingung, durch die der Taktgeber des Empfängers in der Lage ist, sich mit dem Sender zu synchronisieren. Anschließend erscheint der sogenannte Rahmenbegrenzer, der auch Starting Frame Delimiter genannt wird. Er kennzeichnet den Anfang eines Informationsrahmens und besteht aus der Bitfolge 10101011. Die letzte Eins in der Bitfolge unterscheidet zwischen Präambel und Rahmenbegrenzer. Die nächsten Felder enthalten die und Quellenadresse.



KP: Kommunikationsprotokoll

Bild: LAN-Adressierung

In IEEE-802.3 sind 2- und 6-Byte-Adressen zugelassen. Im Grunde werden aber nur 6-Byte-Adressen verwendet. Das höherwertige Bit der Zieladresse ist 0 für normale Adressen und 1 für Gruppenadressen. Durch Gruppenadressen können mehrere Stationen von einer einzigen Adresse empfangen werden.

Mit einer MAC-Adresse wird der physikalische Netzanschluss oder Netz-Zugriffspunkt eines DTEs adressiert und heißt daher auch physikalische Adresse.

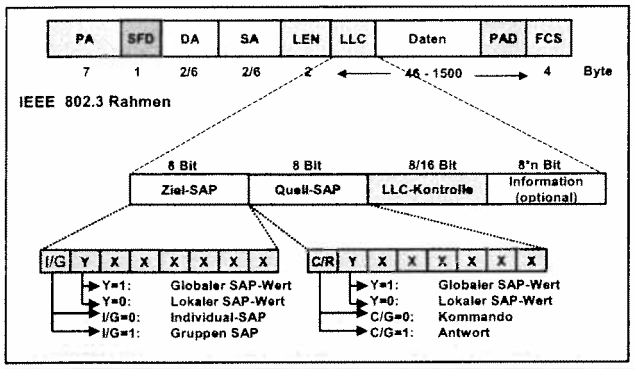
MAC-Adressformat:	1 Bit	1 Bit	22 Bit	24 Bit
	I/G	U/L	OUI	OUA

- I/G = 0: Individuelle Adresse (Unicast Address), die genau ein DTE identifiziert.
- I/G = 1: Gruppen-Adresse (Multicast Address), die eine Gruppe von DTEs identifiziert (nur als Ziel- Adresse, nicht als Quell-Adresse möglich).
- U/L = 0: universelle Adresse (weltweit eindeutig und unveränderbar).
- U/L = 1: lokale Adresse (lokal veränderbar).
- Organizationally Unique Identifier (OUI): Für die Festlegung von universellen Individual-Adressen werden von IEEE für die Bits 3 bis 24 weltweit eindeutige Werte vergeben und den Herstellern zugewiesen.
- Organizationally Unique Address (OUA): Die Werte für die restlichen Bits 25 bis 48 werden von den Herstellern vergeben.

Bild: MAC-Addressen

Ein Rahmen, der an eine Gruppenadresse geschickt wird, wird von allen Mitgliedern dieser Gruppe empfangen. Das Adressieren von Gruppenadressen wird als Multicast bezeichnet. An anderer Stelle wird noch genauer auf diesen Begriff eingegangen. Broadcasts adressieren im Gegensatz zu Multicasts alle Endgeräte, die in einem Netz erreicht werden können.

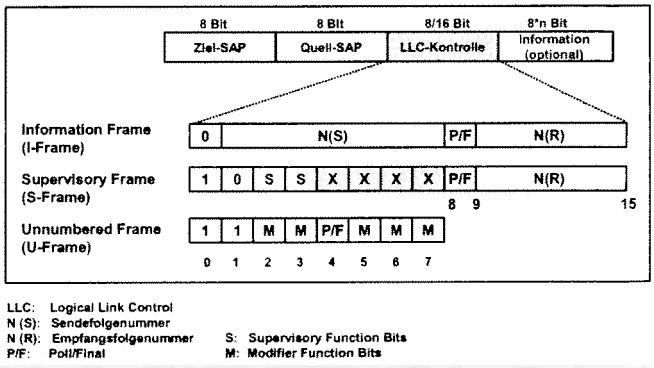
Adressen, die nur aus Einsen bestehen, sind für diese Art der Nachrichtenübermittlung gedacht. Bei den Adressen unterscheidet man außerdem zwischen lokalen und globalen Adressen. Lokale Adressen werden durch den Netzadministrator vergeben und besitzen außerhalb des Netzes keine Bedeutung. Globale Adressen werden von der IEEE vergeben, damit weltweit nur eine Station existiert, die diese Adresse besitzt.



- PA : Preamble
- SDF : Start Delimiter of Frame
- DA : Destination Address
- SA : Source Address
- L : Length
- PAD : Padding Data
- FCS : Frame Check Sequence

SAP: Service Access Point
C/R: Command/Response
XXXXXX: SAP-Angabe

Bild: LLC-Rahmen: Adressierungsfelder



LLC: Logical Link Control
N (S): Sendefolgenummer
N (R): Empfangsfolgenummer
P/F: Poll/Final
S: Supervisory Function Bits
M: Modifier Function Bits

Bild: LLC-Rahmen: Kontrollfeld

- Die LLC-Subschicht übernimmt alle von einem bestimmten Medienzugriffsverfahren unabhängigen Aufgaben der OSI-Schicht 2.
- Die zwischen Partnerinstanzen (Peer Entities) der Sicherungsschicht ausgetauschten Datenblöcke werden als LLC-Rahmen bezeichnet.

Drei Typen von Diensten:

Typ 1 (LLC1):
Unbestätigter verbindungsloser Dienst (unacknowledged connectionless Service): Der Sender verschickt unabhängige Rahmen, deren Empfang nicht bestätigt wird.

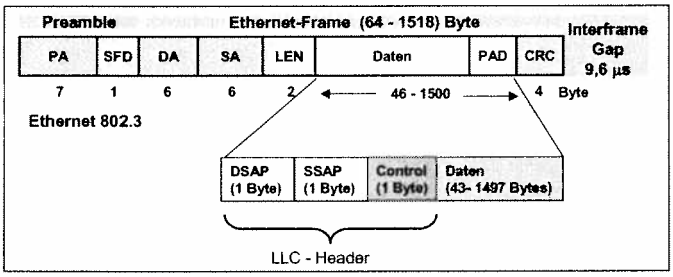
Typ 2 (LLC2):
Bestätigter verbindungsorientierter Dienst (acknowledged connection-oriented Service): Vor der Übertragung bauen Sender und Empfänger eine Verbindung auf. Jeder Rahmen wird numeriert und der Empfang der fehlerfrei und in der richtigen Reihenfolge eingetroffenen Rahmen wird bestätigt.

Typ 3 (LLC3):
Bestätigter verbindungsloser Dienst (acknowledged connectionless Service): Der Sender verschickt unabhängige Rahmen, deren Empfang individuell bestätigt wird.

Bild: Logical Link Control (LLC)

Die Netzschicht muss in der Lage sein, das Ziel festzustellen. Man unterscheidet beide Adressierungsarten durch die Benutzung von Bit 46. Somit gibt es, wenn man höherwertige Bits abrechnet, 7 x 10^{13} globale Adressen, die nur ein einziges Mal gültig sind.

Das Kontrollfeld unterscheidet die verschiedenen LLC-Rahmen. Bei verbindungsorientierten LLC-Rahmen (Class-II-Betrieb) wird dieses Feld um ein Byte auf zwei Byte vergrößert, wodurch 7-Bit-Folgenummern verwendet werden können.



DSAP: Destination Service Access Point
SSAP: Source Service Access Point

Bild: Rahmenformat von Ethernet (IEEE 802.3)

Betrachtet man den Ethernet-Rahmen weiter, so kommt als nächstes das Längenfeld. Dieses Feld gibt die Länge des nachfolgenden Datenfeldes an, welches auch den LLC-Header enthält. Dieser bietet mit der Angabe eines sogenannten Destination und Source Service Access Point (DSAP und SSAP) die Adressierungsmöglichkeit für den gleichzeitigen Betrieb mehrere Kommunikationsverbindungen auf einer Station. Die Länge kann zwischen 0 und 1500 Byte betragen, wobei 0 eher ungewöhnlich ist und Probleme aufwerfen könnte.

Das LLC-Kontroll-Feld Typ 1 wird bei einem verbindungslosen Service (unnumbered U-Format) verwendet.

- M: Codierung des U-Formats.
- P/F=1: Pool/Final Bit (fordert den Empfänger auf, umgehend eine Antwort auf ein Kommando zurückzuschicken.

Bit Nr.	1	2	3	4	5	6	7	8
Typ 1	1	1	M	M	P/F	M	M	M

Das Format des LLC Kontroll-Feldes (Typ 1):

Bitmuster (Control- Feld)	Funktion des LLC- Rahmens
11000000	UI Command Frame
11111101	XID Command Frame
11110101	XID Response Frame
11001111	TEST Command Frame
11000111	TEST Response Frame

Bild: LLC Control-Feld

Das heißt, bei Entdeckung einer Kollision durch den Transceiver wird der aktuelle Rahmen abgeschnitten, wodurch Bruchstücke des Rahmens auf dem Übertragungsmedium vorhanden sind. Da die Mindestrahmenlänge 64 Byte beträgt, um korrekte Rahmen von unkorrekten unterschieden zu können, würde es bei dem Wert 0 zu einer Verwerfung des Rahmens kommen. Falls Datenteile kleiner als 46 Byte sind, füllt deshalb das PAD-Feld den Rahmen bis zum Minimum auf.

Im allgemeinen werden mit der LLC-Adresse die Dienste der über der LLC-Subschicht liegenden Vermittlungsschicht identifiziert.

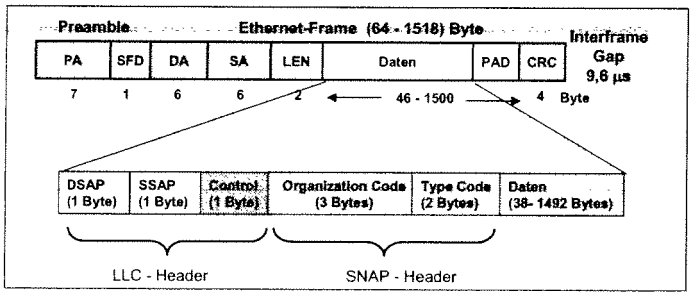
LLC-Adressformat:

1 Bit	7 Bit	1 Bit	7 Bit
I/G	DSAP	C/R	SSAP

- I/G = 0: Individuelle Adresse, die genau einen DSAP identifiziert.
- I/G = 1: Gruppen-Adresse, die eine Gruppe von DSAPs oder alle DSAPs identifiziert.
- C/R = 0: Command Frame.
- C/R = 1: Response Frame.

Bild: LLC-Adressformat (IEEE 802.2)

Ethernet 802.3 SNAP



SNAP: SubNet Access Protocol

Bild: Ethernet Rahmenformat (IEEE 802.3 SNAP)

Durch die Einführung des SNAP wird die Möglichkeit geschaffen, die Protokolltyp-Nummern des DIX Ethernet auch bei allen anderen Rahmenformaten zu verwenden.

Das Format eines SNAP-Headers:

3 Bytes	2 Bytes
Organization Code	Type

- Type: Identifikation eines in der OSI-Schicht 3 angesiedelten Protokolls (identisch mit der Protokolltyp-Nummer des DIX Ethernet).
- Organization Code: Erweiterung des Type-Feldes, um neue Protokolltypen kennzeichnen bzw. abgrenzen zu können

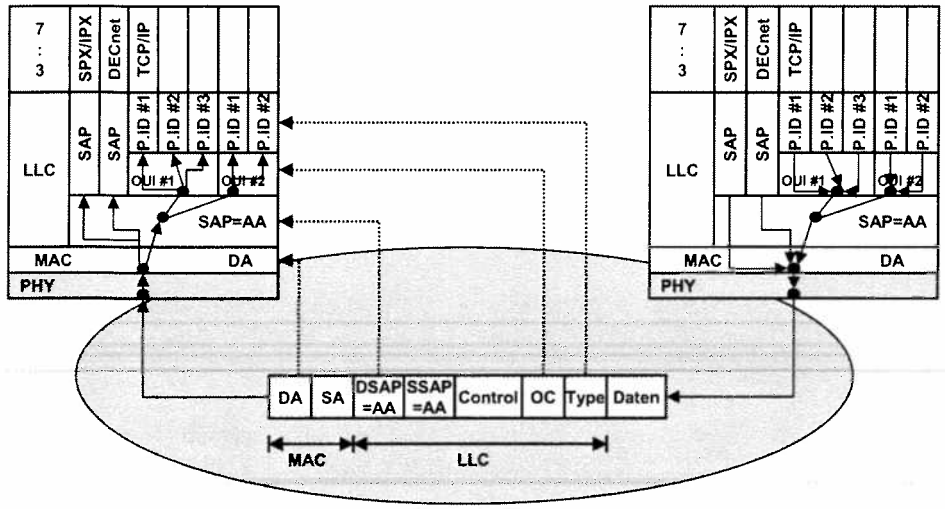
Bild: Sub-Network Access Protocol (SNAP)

• Einige Beispiele für die LLC-Adressen:

LLC- Adresse				Protokoll
DSAP (Binär)	SSAP (Binär)	HEX	DEZ	
00000000	00000000	00	0	LLC-Instanzen
10000000	00000001	01	1	(In Verbindung mit XID- und TEST-Rahmen)
01000000	00000010	02	2	Management der LLC Subschicht
11000000	00000011	03	3	
01100000	00000110	06	6	Internet Protocol (IP)
01000010	01000010	42	66	Spanning Tree Protocol
01010101	10101010	AA	170	Subnetwork Access Protocol (SNAP)
00000111	11100000	E0	224	Novell - Protokoll IPX
00001111	11110000	F0	240	Microsoft - Protokoll NetBEUI
01111111	11111110	FE	254	OSI - Protokolle der Netzschicht
11111111	11111111	FF	255	

Bild: LLC-Adressen

Nach der LLC-PDU folgt das Sub-Network Access Protocol (SNAP), welches die Felder Protokoll-ID oder Organisationscode sowie den Ethernet-Typ enthält. Anschließend folgt das zu übertragende Schicht-3-Protokoll, welches in dieser Abbildung IP ist. Das letzte Feld des Ethernet-Rahmens besteht aus der Frame Check Sequence (FCS), die einen 4 Byte langen Hashcode der Daten enthält. Bei Verfälschung einiger Daten im Rahmen kann man von einer falschen Prüfsumme ausgehen, die den Fehler aufdeckt, aber nicht beheben kann. Der verwendete Algorithmus ist eine zyklische Redundanzprüfung.



DA: Destination Address
SA: Source Address
DSAP: Destination Service Access Point
SSAP: Source Service Access Point
OC: Organization Code

Bild: Aufgaben des Protokolls SNAP

Oft verwenden die Herstellerfirmen ihre eigenen Bezeichnungen für verschiedene Ethernet-Teilbereiche. Ein Beispiel dafür liefert die folgende Tabelle:

Ethernet Rahmenaufbau	Standardisierungs-grundlage	Bezeichnung der Fa. Novell	Bezeichnung der Fa. Cisco
Version II	DIX	Ethernet II	ARPA
802.3 mit 802.2	IEEE/ISO	Ethernet 802.2	LLC
802.3 mit 802.2 und SNAP	IEEE/ISOC	Ethernet SNAP	SNAP
802.3 "raw"	Novell- proprietär	Ethernet 802.3	Novell

Bild: Unterschiedliche Bezeichnungen

- Basisband Übertragung
  - das gesamte technisch nutzbare Frequenzband steht auf einem Übertragungsmedium für einen einzigen Kanal zur Verfügung und beginnt bei Null.
- Breitband Übertragung
  - das gesamte Frequenzband (nicht notwendigerweise bei Null beginnend) steht für mehrere Kanäle zur Verfügung und ist daher in mehrere nebeneinanderliegende Bänder aufgeteilt.
- Codierungsverfahren im Ethernet:
  - Manchester Code
  - 4B/5B
  - 8B/6T
  - PAM 5
  - 8B/10B
  - 64B/66B

Bild: Codierungsverfahren

Codierung

Die Manchester-Codierung ist die Leitungscodierung für das 10Mbit/s-Ethernet. Durch die Darstellung eines einzelnen Bits durch ein 2-Bit-Wort (1B2B) besteht der Leitungscode aus insgesamt vier Codewörtern. Die logische Null wird hierbei als 10, die logische 1 als 01 codiert. Die Codewörter 00 und 11 sind redundant und werden nicht genutzt. Unter Verwendung symmetrischer Spannungspegel von -0,85 V und +0,85 V sowie einer Verdopplung der Signalfrequenz, wird neben der Gleichstromfreiheit des Signals eine einfache senderseitige Taktrückgewinnung ermöglicht. Durch die starken Anforderungen, die eine Verdopplung der Signalfrequenz bzw. der Baudrate (Schrittgeschwindigkeit) bei höheren Übertragungsraten an die genutzten Leitungen stellen würden, ist die Manchester-Codierung bei Fast- oder Gigabit-Ethernet nicht mehr im Einsatz.

Standard Ethernet (10 Mbit/s)

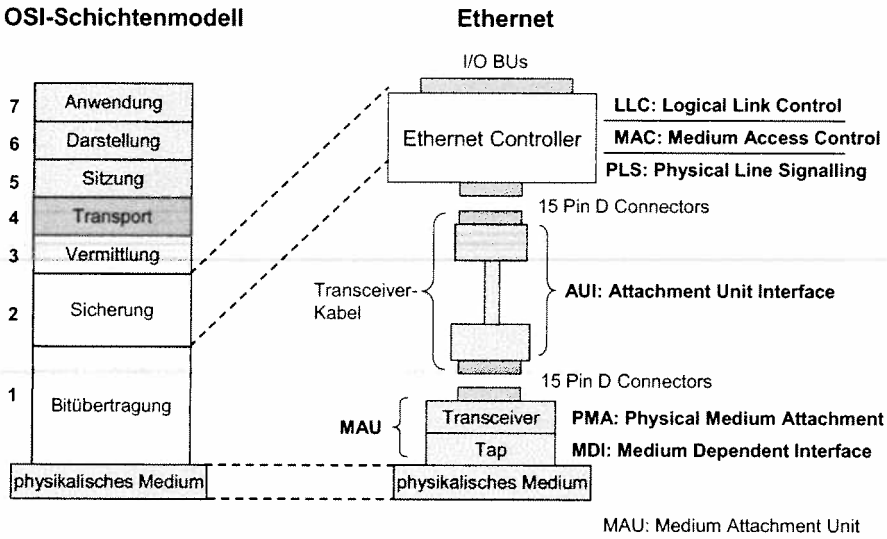


Bild: Der Ethernet-Standard

Teilbereiche der Bitübertragungsschicht:

- MAU - Media Access Unit (Transceiver)
  - Die Medium-Anschlusseinheit (MAU) stellt eine mediumspezifische Anbindung an das Medium dar.
- AUI - Attachment Unit Interface
  - Stellt die Verbindung zwischen Transceiver und Endgerät dar.
- PLS - Physical Line Signalling
  - dient zum Austausch von Daten zwischen zwei MAC-Schichten und wird zur Steuerung des Medienzugriffs (CSMA/CD) benutzt (für die Signalisierung spezieller Zuständen des physikalischen Mediums, wie »Medium belegt«, »Medium frei« oder »Kollision auf dem Medium«.

Bild: Bitübertragungsschicht

- Funktionen der MAU:
  - Übermittlung der Signale auf das Medium
  - Empfang der Signale vom Medium
  - Feststellen der Signalfreiheit auf dem Medium
  - Überwachen der Daten auf Kollision
- Bestandteile:
  - **MDI - Medium Dependant Interface** (physikalische Schnittstelle zum Medium)
  - **PMA - Physical Medium Attachment** (funktionale Schnittstelle zum Medium)

Bild: Media Access Unit (MAU)

- Transmit-Funktion: Senden von seriellen Daten-Bitströmen auf das Medium
- Receive-Funktion: Empfangen serieller Datenströme vom Medium
- Kollisions-Funktion: Die PMA kann zwei verschiedene Zustände des Mediums an das Endgerät weiterleiten:
  - Ein ungültiges Signal wurde auf dem Medium empfangen
  - Eine Kollision wurde erkannt
- Jabber-Funktion: Diese Funktion stellt einen Unterbrechungsmechanismus zur Verfügung, welcher garantiert, dass keine MAU länger als 30 ms hintereinander Daten auf das Medium sendet. Hardware unterbricht den Datentransfer und signalisiert dem Endgerät einen Sendeabbruch.
- Monitor-Funktion (optional): Die Monitorfunktion ermöglicht das Abschalten der Sendefunktion unter Beibehaltung der Kollisions- und Empfangsfunktion.

Bild: Physical Medium Attachment (PMA)

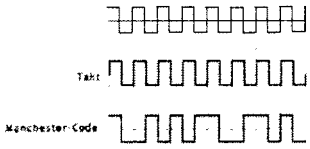
- Das PLS liefert der MAC-Schicht Informationen zur Steuerung des Medienzugriffs nach dem CSMA/CD-Verfahren.
- Es signalisiert die verschiedenen Zustände des Mediums:
  - belegt,
  - frei,
  - Kollision aufgetreten.
- Baulich ist das PLS im MAC-Controller integriert.

Bild: Physical Line Signalling (PLS)

- Der physikalische Anschluss an das Übertragungsmedium.
- Durch das MDI werden physikalische Anpassung des Datensignals (z. B. Signalpegel) und die mechanische Anbindung (z. B. der Aufbau eines Steckers) definiert.

Bild: Medium Dependant Interface (MDI)

- **Manchester Codierung:**



0 → 10 (Pegel positiv und Polaritätswechsel von positiv nach negativ in der Taktmitte)

1 → 01 (Pegel negativ und Polaritätswechsel von negativ nach positiv in der Taktmitte)

- Der Manchester Code wird in folgenden Ethernet Systemen verwendet:
  - 10Base5
  - 10Base2
  - 10Base-T
  - 10Base-F

Bild: Manchester Code

Netzstruktur

Die Struktur eines 10-Mbit/s-Ethernet kann abhängig vom verwendeten Ethernet-Standard durch unterschiedliche Übertragungsmedien realisiert werden.

10Base-5

Das klassische Ethernet 10Base-5 besitzt ein gelbes Koaxialkabel (Yellow Cable) mit einer Impedanz von 50 Ohm. Die Dämpfung beträgt bei 10 MHz 17 dB/km mit einer Signalausbreitungsgeschwindigkeit von etwa 5 µs/km oder 0,77-facher Lichtgeschwindigkeit. Als Topologie wird die Busstruktur verwendet. Als andere Bezeichnung für 10Base-5 ist auch Thick Ethernet üblich. Werden mehrere Endstationen an einem Kabel angeschlossen, so entsteht ein Netzsegment.

Ethernet kann dabei aus mehreren Netzsegmenten bestehen, die mit Repeatern verbunden werden. Die maximale Länge eines Bussegments ist auf 500 m festgelegt. Jedes Bussegment ist dabei in der Lage, bis zu 100 Stationen in einem Abstand von minimal 2,5 m anzuschließen. Für dieses resultierende Netz, das häufig auch als Subnetz bezeichnet wird, gelten aufgrund der Signallaufzeitbeschränkungen bestimmte topologische Grenzen, die unabhängig vom verwendeten Ethernet-Standard sind:

- Zwischen zwei beliebigen Stationen eines Ethernets dürfen sich nicht mehr als vier Repeater befinden.
- Die maximale Anzahl von Netzsegmenten zwischen zwei Stationen ist auf fünf begrenzt, von denen nur drei zum Anschluss von Endstationen genutzt werden können. Die restlichen zwei Anschlüsse dürfen nur zur Überwindung größerer Distanzen eingesetzt werden und heißen Linksegmente.
- Linksegmente können bis zu 1000 m (2000 m) lang sein und enden jeweils in einem Remote Repeater.

Diese genannten Restriktionen kann man nur durch andere Koppelelemente, wie beispielsweise Router und Bridges, umgehen. Der Bereich innerhalb eines Subnetzes wird auch als Collision Domain bezeichnet, da innerhalb seiner Grenzen zur Zeit immer nur eine Station senden darf.

10Base-2

Dieser Standard beinhaltet ebenfalls eine Busstruktur. Als Medium wird ein Koaxialkabel nach RG-58 verwendet, welches wesentlich flexibler in der Handhabung ist und somit auch kostengünstiger als 10Base-5. Deshalb hat man 10Base-2 auch als Cheapernet oder Thin-Ethernet bezeichnet. Aufgrund der höheren Dämpfungswerte liegt die erzielbare Länge pro Bussegment bei 185 m. Die Anzahl der Stationen pro Segment ist auf 30 limitiert. Zum Anschluss an das Übertragungsmedium werden BNC-Stecker verwendet. Dies ist ein weiterer Unterschied zu 10Base-5, da jede einzelne Station mittels einer BNC-Buchse, die sich auf der Controller-Karte befindet, direkt an das Übertragungsmedium angeschlossen wird. Da sich Änderungen im Cheapernet ausschließlich auf die Bitübertragungsschicht beziehen, können 10Base-5 und 10Base-2 über geeignete Repeater oder Adapter miteinander verbunden werden.

10Base-T

Dieses System besteht aus einer Stern-Topologie. Durch diesen Standard wurde es erstmals ermöglicht, auch Ethernet mit einer sternförmigen Topologie zu versehen. Unshielded Twisted Pair (UTP) ist dafür die typische Verkabelung, basierend auf Kategorie 3, 4 und 5. Es werden zwei Adernpaare des UTP-Kabels für Receive und Transmit verwendet. Zusätzlich ist der Vollduplex-Betrieb hinzugekommen. Der Bus wird hierbei innerhalb des verwendeten Hubs (Sternkoppler) betrieben. Die Verbindung zwischen Hub und den Stationen darf 100 m nicht überschreiten. Wie bei Twisted Pair üblich, erfolgt die Verbindung zum Hub bzw. zur Station über RJ-45 Stecker, ein achtpoliges Miniaturstecksystem als Schnittstelle von der Datendose des Verkabelungssystems zu den Endgeräten.

Drei weitere Standards zur optischen Übertragung wurden für Ethernet spezifiziert:

- 10Base-FL
- 10Base-FP
- 10Base-FB

10Base-FL ist eine Fiber Optic Inter Repeater Link (FOIRL) Erweiterung und abwärtskompatibel zu dieser Spezifikation. Es beschreibt alle Funktionen zur Datenübertragung von einer Medium Access Unit (MAU) bzw. einem Transceiver zu einem aktiven Hub bzw. Sternkoppler und Verbindungen zwischen Hubs. Daten werden asynchron übertragen und entsprechen im wesentlichen der Spezifikation FOIRL. Dabei sind maximal fünf Repeater für eine 3-stufige Netzhierarchie erforderlich. Die maximale Länge beträgt 2000 m. Signale von 10Base-FL sind nicht kompatibel zu 10Base-FP und 10Base-FB. Wenn eine Kopplung erfolgen soll, funktioniert das nur über Repeater oder Bridges. Der Endgeräteanschluss an die Glasfaser erfolgt direkt auf dem Controller mit Hilfe eines integrierten Transceivers oder über ein Access Unit Interface (AUI). Der externe Transceiver auf der Glasfaser ermöglicht dabei die Anpassung an den Standard 10Base-FL.

10Base-FP
Der Standard 10Base-FP definiert alle Funktionen zur Datenübertragung von einer MAU zu einem passiven Hub über die Glasfaser. Der passive Hub unterscheidet sich vom aktiven dadurch, dass Übertragungsmedium und Hub vollständig passiv sind, keine Abstrahlung und keine Stromversorgung haben. Die maximale Länge beträgt im Gegensatz zu 10Base-FL nur 500 in. Eine Unterstützung von 1024 Stationen in einem einzelnen Netz ist möglich. Die Anzahl der Ports pro Hub ist nicht festgehalten worden. Die Zahl sollte aber zwischen 2 und 33 betragen. 10Base-FP ist eine preiswerte Konfiguration und für kleine Installationen oder LANs geeignet. Eine eigene Stromversorgung entfällt, da die Endgeräte die optische Energie liefern.

10Base-FB
10Base-FB ist entgegen 10Base-FP für den Backbone ausgelegt worden. Der Standard definiert alle Funktionen zur Datenübertragung zwischen aktiven Hubs (Repeater oder Bridges). Verbindungen zwischen Transciever und Hub werden ebenfalls berücksichtigt. 10-Base-FB überträgt die Daten synchron. Dadurch werden keine Repeater mehr benötigt, da auch die Signalaufbereitung in den Koppelelementen vorgenommen wird. Die maximale Segmentlänge eines Linksegments beträgt 2000 m. Redundante Verbindungen sind durch die synchrone optische Übertragungstechnik möglich. Verbindungsausfälle können so durch automatisches Umschalten aufgefangen werden. Eine Fehlererkennung wird durch ein 1,67-MHz-Signal ermöglicht, welches durch den empfangenen Transceiver an den Sender zurück gesendet wird.

10Broad-36
Zusätzlich sei noch der Breitband-Ethernet-Standard erwähnt, der die Bezeichnung 10Broad-36 besitzt. Hier wird ein 75-Ohm-Koaxialkabel nach CAT-5 eingesetzt. Die Topologie ist ein unregelmäßiger Baum. Die maximale Entfernung zwischen zwei Stationen beträgt 3600 m. Die AUI-Schnittstelle wird unverändert eingesetzt. Durch die Breitbandtechnik wird ein Hin- und Rückkanal benötigt. Die Bandbreite beträgt dadurch 36 MHz (18 MHz in jeder Richtung). 14 MHz werden dabei für die Datenübertragung verwendet, während 4 MHz für die Signalisierung reserviert sind. Neben Einkabelsystemen mit einem Frequenz-Offset von 156,25 MHz oder 192,25 MHz, werden auch Zweikanalsysteme verwendet. Das Nutzsignal wird auf eine Trägerfrequenz aufmoduliert, wobei die Bandbreite des Kabels in unabhängig nutzbare Frequenzbänder aufgeteilt wird. Dadurch kann man das verwendete Medium mehrfach ausnutzen. Ein Frequenzband wird dadurch ähnlich wie ein Basisband genutzt. Der Verbreitungsgrad ist allerdings als sehr gering zu bezeichnen.

Isochrones Ethernet
Aufgrund der schlechten isochronen Eigenschaften von Ethernet, di man zur Unterstützung multimedialer Anwendungen benötigt, wurde das Isochrone Ethernet entwickelt. Das Isochrone Ethernet besitzt eine hybride Netzarchitektur, die den Standard IEEE 802.3 mit dem ISDN verbindet. Aus diesem Grund hat man zusätzlich 6,144 Mbit/s dem 10-Mbit/s-Ethernet hinzugefügt. Die erweiterte Kapazität wird dabei in 95 B-Kanäle und einen D-Kanal mit jeweils 64 kbit/s aufgeteilt. Die Signalisierung des D-Kanals wird wie beim ISDN mittels des Q.931-Protokolls vorgenommen. Die B-Kanäle können einzeln oder gebündelt verwendet werden. Der reine Datenverkehr findet weiterhin über 10-Mbit/s-Ethernet über das CSMA/CD-Verfahren statt und bleibt von den isochronen Datenströmen unbeeinflusst. Unterstützt wird UTP und STP nach Kategorie 3, 4 und 5. Die Manchester-Codierung wird nicht mehr eingesetzt, die 4B/5B-Codierung hat diesen Platz eingenommen. Um ein Isochrone Ethernet aufbauen zu können, sind bestimmte NICs notwendig sowie Hubs und Switches. Aus diesem Grund konnte sich dieser Ethernet-Standard auch nie durchsetzen.

Netzanschlüsse und Teilschichten
Der Ethernet-Anschluss einer Station setzt sich aus einem Transceiver, dem dazugehörigen Kabel und einem Controller zusammen. Der Transceiver, ein Kunstwort zwischen Transmitter und Receiver, wird dabei auch als Medium Access Unit (MAU) bezeichnet. Er besitzt die Fähigkeit, Signale auf das Übertragungsmedium zu senden oder es von ihm zu empfangen. Die MAU besteht aus dem Tap und dem eigentlichen Transceiver, einer Basisband Sende- und Empfangseinheit. Der Tap ist das Bauteil, an dem der Transceiver an das Kabel angeschlossen wird. Der Transceiver ist verantwortlich für die Signal- und Kollisionserkennung. Er sendet auch das JAM-Signal bei Erkennung einer Kollision.

Das Transeciver-Kabel wird als Attachment Unit Interface (AUI) bezeichnet. Es verbindet den Transceiver mit dem zur Station gehörenden Controller. Das Kabel kann bis zu 50 m Länge aufweisen und besitzt eine vom Bus abweichende Spezifikation. Es enthält fünf einzeln abgeschirmte, miteinander verdrillte Doppeladern (STP), von denen jeweils zwei für die Dateneingabe und Datenausgabe sowie für die Kontrolldaten zuständig sind. Das verbleibende Paar ist für die Stromversorgung des

Transceivers vorhanden. Dem Controller wird dadurch eine definierte Schnittstelle zur Verfügung gestellt (AUI), die von den spezifischen Eigenschaften des Übertragungsmediums und der eingesetzten Übertragungstechnik unabhängig ist.

Der Ethernet Controller realisiert Schicht-2-Funktionalität. Somit generiert er den Ethernet-Rahmen und den Cyclic Redundancy Check (CRC), der für die Fehlererkennung genutzt wird. Auf der Netzseite stellt der Controller noch die Physical Layer Signalling (PLS) zur Verfügung, die die Signalaufbereitung vornimmt. Das heißt, hier wird die Codierung bzw. Decodierung der Daten durchgeführt.

Signalverzögerung des Übertragungspfads

Die Signalverzögerung aller an der Übertragung beteiligten Komponenten im Ethernet stellt eine besondere Problematik dar. Die Summe aller Verzögerungen in einem Übertragungspfad darf nicht größer werden, als die Zeit, die benötigt wird, um einen Rahmen mit minimaler Größe (64 Byte) erfolgreich zu transportieren. Bei Anwendung verschiedener Ethernet-Standards in einem Netz, wird die Übersicht deutlich erschwert. Es kann dann nicht mehr so einfach abgeschätzt werden, ob die untere Grenze der Signalverzögerung nicht doch überschritten wird. Eine Möglichkeit für eine Überprüfung ergibt sich aus der Berechnung der Gesamtverzögerung eines Übertragungspfades (Path Delay Calculation). Diese wird in Abhängigkeit der verwendeten Übertragungsstrecke über die maximale Signallaufzeit bzw. den Round Trip Delay (RTD) ermittelt. Dafür muss man ein Übertragungsmodell zugrunde legen, das aus zwei Stationen sowie einem linken, mittleren und rechten Segment besteht.

Übertragungsmodell

Zur Berechnung der Signallaufzeit dient ein Übertragungsmodell. Dabei wird vorausgesetzt, dass auch bei einer Minimalkonfiguration (4 Repeater und 5 Segmente, inkl. 2 Linksegmente) eine Erkennung von Kollisionen sichergestellt werden muss. Das heißt, die Station A sendet solange, bis die durch Station B verursachte Kollision von Station A erkannt wurde. Um nun feststellen zu können, ob die zulässige Signallaufzeit überschritten wurde, muss der längste Pfad zwischen zwei Stationen ermittelt werden. Das ist der Übertragungspfad, welcher die größte Signallaufzeit aufweist. Genaue Kenntnisse über die physikalische Netzstruktur sind dafür unerlässlich. Anschließend kann man entscheiden, welche der einzelnen Segmente wie in das Übertragungsmodell passen. Eine Aufteilung in linkes, mittleres und rechtes Segment muss also vorgenommen werden. Der Standard des Ethernets ist dabei auch entscheidend. Weiterhin sollte die Länge der einzelnen Segmente bekannt sein, damit die Segmentverzögerungszeit (SVZ) möglichst genau ermittelt werden kann. Folgende Gleichung wird dafür verwendet:

SVZ = Basiswert + Länge – RTD/Meter

Falls die Länge eines Segments nicht bekannt ist, muss der Maximalwert angenommen werden, um in jedem Fall einen sicheren Wert zu erhalten. Dabei wird vorausgesetzt, dass die zulässige Länge des Segments nicht überschritten wird. Der Wert der Übertragungsverzögerung ergibt sich nun aus der Summe aller SVZ mit Hinzunahme der Verzögerungen aller im Übertragungsweg enthaltenen AUI-Kabel, die länger als 2 m sind. Das AUI-Kabel der sendenden Station A wird dabei nicht berücksichtigt. Ist der berechnete Gesamtwert größer als 57,6 µs, wird es in diesem Ethernet zu Problemen kommen.

Fast-Ethernet (IEEE 802.3u)

Ein Local Area Network (LAN) besitzt gegenüber anderen Netzen eine hohe Anschlussdichte und wird üblicherweise von der Organisation seiner Benutzer betrieben. Ethernet stellt in diesem Bereich die wichtigste Netztopologie dar, da praktisch an die 80-90% der weltweit verkauften Network Interface Cards auf dieser Technologie beruhen.

- **IEEE 802.12**
  - neues MAC-Verfahren
- **IEEE 802.3u (100 BASE-T)**
  - Unterschiede zu 802.3
    - Media Independent Interface (MII) ersetzt AUI
      - Transmit Clock, Transmit Data, Transmit Enable, Transmit Error
      - Receive Clock, Receive Data, Receive Data Valid, Receive Error
      - Carrier Sense, Collision
      - Management Data Clock, Management Data Input Output
    - Ersetzen der Manchester-Codierung bei AUI durch NRZ bei MII
    - Dual-Speed 10/100 Mbit/s Operation mit Auto-Negotiation
    - Vollduplex-Operation
    - Punkt-zu-Punkt-Verkabelung

Bild: Fast-Ethernet

Charakteristika

- Übertragungsraten von 10-100 Mbit/s
- Medienzugriffsverfahren und Format der Dateneinheiten wie CSMA/CD

- Flexibles Verkabelungskonzept (Hierarchie von Hubs)
- Kompatibilität zum existierenden Ethernet-Standard
- Einfache Migration
- Autonegotiation (Protokoll zur automatischen Festlegung der Übertragungsrate)

- Multiport-Bridges (Switches) möglich
- Vollduplex-Betrieb (keine CSMA/CD mehr erforderlich)
- Senderate kann durch PAUSE-Dateneinheiten gedrosselt werden

Bild: Fast-Ethernet

Variante	Medium
100BASE-T4	Kabel mit 4 Doppeladern, UTP, Kat. 3, 4, 5, Leitungscode 8B6T, NRZ, nicht Vollduplex
100BASE-FX	Zwei Multimode-Glasfasern (62,5/125 µm), Leitungscode 4B5B, NRZI, Vollduplex
100BASE-TX	Kabel mit 2 Doppeladern, UTP, Kat. 5 alternativ 2 Doppeladern, STP, 150 m, Leitungscode MLT-3 Vollduplex
100BASE-T2	Kabel mit 2 Doppeladern, UTP Kat.3, 4, 5, Leitungscode PAM5 Vollduplex

Datenrate 100 Mbit/s, Segmentlänge einheitlich 100 m, Netzausdehnung 200 m (bei 10BASE-FX bis 400 m)

S Short range fiber
L Long range fiber
C Copper

T IEEE 802.3 access protocol and Format
X FDDI optical Interface
UTP Unshielded Twisted Pair
FDDI Fiber Distributed Data Interface

Bild: Fast-Ethernet

Dadurch lässt sich der Erfolg von Fast-Ethernet und die Prognosen für Gigabit-Ethernet leicht nachvollziehen, da Administratoren bei Netzerweiterungen gerne auf die gleiche Technik setzen, um den Schulungs- und Kostenaufwand in Grenzen zu halten: Lediglich der zentrale Hub oder die Switching Komponente muss ausgetauscht werden. Vorhandene Verkabelung, falls sie auf Kupferkabel der Kategorie 5 basiert, lässt sich ebenfalls weiter verwenden. Falls keine strukturierte Verkabelung vorhanden ist bzw. alte Kabelmedien (z.B. Koaxialkabel) eingesetzt werden, ist hingegen mit wesentlich höheren Kosten zu rechnen.

Netzanschlüsse und Teilschichten

Aufgrund erhöhter Anforderung an die Bandbreite durch zunehmende Rechenleistung der Clients, neue Anwendungen und höherer Benutzerzahlen musste sich Ethernet zwangsläufig weiterentwickeln. Zur Anpassung des aus dem Jahre 1983 stammenden Standards IEEE 802.3 lagen zwei Standardisierungsvorschläge vor: IEEE 802.12 (100BaseVGAnyLAN) und IEEE 802.3 (100Base-X). Während sich IEEE 802.12 von dem Zugriffsverfahren CSMA/CD löste, basierte der Standardisierungsvorschlag IEEE 802.3 weiterhin auf diesem Verfahren. 1995 wurde dann IEEE 802.3 als IEEE 802.3u (100Base-T) veröffentlicht und allgemein als Fast-Ethernet bezeichnet.

Die Datenübertragungsrate von Ethernet-LANs ist maßgeblich durch den CSMA/CD-Algorithmus begrenzt, wenn man Fast-Ethernet als Shared Medium einsetzt.

Der Engpass von Ethernet-LANs mit dem CSMA/CD-Verfahren ist demnach das Zugriffsverfahren selber, da nur Netzlasten bis 50% noch effektiv sind. Netzlasten über 40% verursachen bereits dermaßen viele Kollisionen, dass der Durchsatz rapide absinkt. So sind Datenraten von 25-30 Mbit/s als Richtwert in einem Shared-Media-100-Mbit/s-Fast-Ethernet anzusehen. Im Full-Duplex-Modus mit dedizierten Switch-Anschlüssen lassen sich hingegen 50-80 Mbit/s durchaus erreichen.

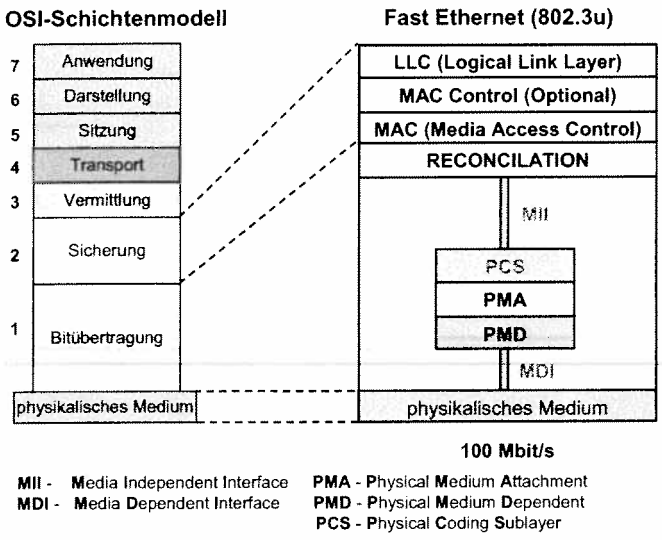


Bild: Fast-Ethernet Schichtenmodell

Als Grundlage für eine Fast-Ethernet-Verkabelung dient, im Gegensatz zum Vorläufer, ausschließlich die Stern-Topologie. Neben der von 10BaseT bekannten UTP-Verkabelung der Kategorien 3, 4 und 5 kommt STP-Verkabelung und Glasfaser zum Einsatz. Fast-Ethernet legt deshalb neben dem MAC-Sublayer drei auf das jeweilige Übertragungsmedium angepasste Bitübertragungsschichten (100Base-T4, 100Base-TX und 100Base-FX) fest.

Diese beinhalten neben einem mediumunabhängigen Physical Medium-Dependent Sublayer (PMD) auch die Teilschichten Physical Medium Attachment (PMA) und Physical Coding Sublayer (PCS). Die PMD Sublayer bildet die unterste Teilschicht der Bitübertragungsschicht. Sie definiert die physikalischen Eigenschaften und Übertragungsparameter (z.B. Datenrate) sowie die korrekte Signalisierung. PMD ist bei Fast-Ethernet als auch für FDDI spezifiziert.

- Reconciliation Subschicht
- Media Independent Interface (MII)
- Codierungsverfahren:
  - 4B/5B (100Base-TX und 100Base-FX)
  - 8B/6T (100Base-T4)
  - PAM5 (100Base-T2)
- Übertragungsmedium:
  - 4 Paar UTP Kat. 3,5 (100Base-T4)
  - 4 Paar TP Kat. 5 (100Base-TX)
  - Glasfaser (100Base-FX)

Bild: Besonderheiten des Fast-Ethernet

PMA ist der physikalische Medienzugang und ist die funktionale Schnittstelle zum Übertragungsmedium. Diese Schnittstelle übernimmt eine Reihe von Übertragungs- und Steuerfunktionen (Reset, Transmit, Receive, Carrier Sense, Link Integrität, Jammer, Align und Clock Recovery).

Mit der Funktion Transmit werden serielle Bitströme über das Übertragungsmedium gesendet. Dabei dürfen nur maximal zwei volle Bitzeiten fehlerhaft sein. Bereits jedes zweite Bit sollte den spezifischen Anforderungen bezüglich Zeit- und Pegelverhalten entsprechen. Verzögerungen zwischen der MAU-Schnittstelle und dem Kabel dürfen nicht mehr als eine halbe Bitzeit betragen. Die Funktion Receive dient dem Empfang serieller Bitströme. Bei der Datenübertragung dürfen nur maximal fünf Bitzyklen verloren gehen oder fehlerhaft übertragen werden. Auch hier muss jedes zweite Bit den Spezifikationen entsprechen. Eine zusätzliche Funktion ist für die Erkennung von Kollisionen zuständig.

- Die Funktion des PLS wird durch die Reconciliation Subschicht ersetzt.
- Die Reconciliation Subschicht stellt eine logische Schnittstelle zwischen MAC-Schicht Bitübertragungsschicht dar.
- Die Aufgabe der Reconciliation Subschicht ist die Wandlung der MAC/Physical Line Signalling-Primitive in MII-Signale und umgekehrt.
- Baulich ist die Reconciliation Subschicht als Bestandteil der 100 Mbit/s MAC-Einheit implementiert.

Bild: Reconciliation Subschicht

Bei auftretenden Kollisionen wird von der MAU der Signal Quality Error (SQE) übermittelt. Das Signal wird durch eine Frequenz mit der halben Bitrate dargestellt, um sie sofort erkennen zu können. Die optionale Monitorfunktion ist für das Ausschalten der Sendefunktion zuständig. Die Kollisions- und Empfangsfunktion bleiben dabei unverändert erhalten.

Zusätzlich ist die Tellschicht Reconcilation Sublayer (RS) spezifiziert worden. Der RS und das MII bilden beim Fast-Ethernet zusammen den Zugang zur physikalischen Schicht. Diese Teilschicht kommuniziert über eine SAP-ähnliche Schnittstelle mit der Bitübertragungsschicht, die sich Media Independent Interface (MII) nennt. Der RS übernimmt hierbei die logische Schnittstelle zwischen MAC-Schicht und MII und wandelt die MAC/PLS-Primitives in MII-Signale um. Das MII kann zusätzlich zur Abtrennung der mediumabhängigen Teilschicht in Form eines Transceivers genutzt werden. Die IEEE spezifiziert dazu eine 40-polige Steckverbindung. Entsprechend der neueren 802.3-Standards wird auch bei Fast-Ethernet auf externe Transceiver zugunsten einer integrierten Lösung verzichtet. Die maximale Ausdehnung wird durch das Zugriffsverfahren begrenzt. Da die Übertragungsrate auf einen zehnmal höheren theoretischen Wert angehoben wurde, bedeutet das eine automatische Reduzierung der maximal zulässigen Kabellänge um den Faktor 10. Die maximale Kabellänge zwischen zwei Stationen ist daher auf 205 m festgelegt. Der Abstand jeder einzelnen Station zum Hub kann die Hälfte betragen.

- Das MII ersetzt das im 10 Mbit/s Ethernet verwendete AUI.
- die Funktion des MII ist dieselbe wie vom AUI, nämlich die Trennung der MAC-Schicht und unterschiedlichen Übertragungsschichten.
- Genauso wie beim AUI kann das MII entweder intern platziert sein oder als Schnittstelle für die unteren Teilschichten nach außen über eine 40-polige MMI Buchse geführt werden.
- **MII Signale:**
  - 4-Bit breite Datenleitungen
  - Taktrate 25 MHz
  - Management Signale
  - Collision (COL)
  - Carrier Sense (CRS)
  - Data Valid
  - Error und Enable

Bild: MII (Media Independent Interface)

Die Teilschicht PCS ist die obere Teilschicht der Bitübertragungsschicht. Sie stellt die direkte Verbindung zum Media Independent Interface (MII) dar und nimmt die Codierung bzw. Decodierung der Daten vor. Zusätzlich werden die Signale Carrier Sense und Collision Detect hier generiert. Je nach Fast-Ethernet-Standard, wird das Signal bei 100Base-T4 in 8B/6B, bei 100Base-FX in 4B/5B-Codierung als Non-Return-to-Zero-Inverted (NRZ-I), bei 100Base-TX ebenfalls in 4B/5B-Codierung, allerdings in Multi Level Transmission (MLT-3), und bei 100Base-T2 im PAM5-Verfahren übertragen.

100Base-T wird von fast allen Herstellern unterstützt. In der sogenannten Fast-Ethernet-Allianz sind über 60 Hersteller vertreten. 100Base-T verspricht einen schnellen und kostengünstigen Umstieg auf eine höhere Bandbreite. Für die Vernetzung im Backbone-Bereich ist zusätzlich der Einsatz der Switching Technologie notwendig. Dies hängt mit den Eigenschaften von 100Base-T zusammen, d.h. mit den redundanten Strukturen, der Entfernungsbeschränkung (Cat-5-UTP: 100 m; Multimode: 2 km; Monomode: 20 km) und der Protokolleffizienz.

Die Anpassung an unterschiedliche Übertragungsraten wird automatisch durchgeführt. Das heißt, beim Hochfahren des Systems oder in Ruhephasen wird ein Fast Link Pulse (FLP) ausgesendet, der die Integrität der Verbindung überprüft und für die Auto-Negotiation genutzt wird. Der FLP findet in Ethernet-/Fast-Ethernet-Adaptern, Ethernet -Dual-Repeatern und Ethernet-Switches Verwendung und unterstützt Datenraten von 10 bis 200 Mbit/s (Vollduplex). Dabei kommt automatisch die höchste Übertragungsrate zum Einsatz.

Codierung

Aufgrund der unterschiedlichen Medien bei Fast-Ethernet und der 10fachen Datenrate werden auch verschiedene Codierungsverfahren eingesetzt, um den jeweiligen Anforderungen gerecht zu werden. Zu nennen sind dabei die Verfahren:

- 4B/513
- MLT-3
- 8B/6T
- PAM5

Die physikalische Codierung 4B/5B unterteilt alle Daten in 4-Bit-Einheiten (Nibble) und wandelt sie nach einer Tabelle in 5-Bit-Einheiten (Symbole) um. Diese Codiertabelle ist so aufgebaut, dass unabhängig von den Eingangsdaten nie Symbole mit mehr als drei Nullen in Folge auftreten. Der Vorteil dieser Methode liegt darin, dass man die NRZI-Codierung nutzen kann, ohne dass bei langen Nullsequenzen die Synchronisation verloren geht. Allerdings entsteht ein Overhead von 25%, wodurch die Datenrate auf 125 Mbit/s erhöht werden muss, wenn man 100 Mbit/s erreichen will. Von den 32 verschiedenen Zeichen, die mit dem 4B/5B-Code erzeugt werden, werden 16 zur Nutzdatenübertragung benötigt, die restlichen 16 für Steuerzwecke.

Die MLT-3-Codierung ist hingegen ein reines Ternärverfahren. Es werden die Signalpegel -1V, 0V und +1V eingesetzt. Damit unterscheidet es sich grundsätzlich zu NRZ und NRZ-1, die zu den Zweipegelverfahren gehören. MLT-3 ist in der Lage, die Basisfrequenz zu reduzieren, wodurch geringere Abstrahlmöglichkeiten vorhanden sind. Zusätzlich kann man auf geringere Verkabelungsqualität (UTP, Kategorie 5) zurückgreifen. Durch die geringere Basisfrequenz wird außerdem noch eine geringere Bitfehlerrate erreicht. MLT-3 wird bei FDDI verwendet und bei 100Base-TX.

Die 8B/6T-Codierung wandelt hingegen 8-Bit-Wörter in 6stellige Ternärsymbole um. Auch hierbei handelt es sich um eine physikalische Codierung. Die dreiwertigen Symbole unterscheidet man wie beim MLT-3-Verfahren. Das 8B/6T-Verfahren findet ausschließlich Verwendung im Standard 100Base-T4.

Das letzte Verfahren ist kein Codierverfahren im eigentlichen Sinn. PAM5 ist eine Puls Amplitude Modulation (PAM), die mit fünf unterschiedlichen Pegeln in zwei Ebenen arbeitet. Die Werte der Pegel werden unterteilt in -2V, -1V, 0V und +1V, +2V. Dadurch können bis zu 25 Signalpegel dargestellt werden. Der Standard 100Base-T2 verwendet dieses Verfahren, um die Bitrate reduzieren zu können. Da 100Base-T2 auch UTP der Kategorie 3 unterstützt, ist das auch zwingend notwendig.

- 8B/6T Codierung:

Bitfolge	8B6T- Code
	+ - 0 0 + -
0000 0001	0 + - + - 0
.....	.....
0000 1110	- + 0 – 0 +
.....	.....
1111 1110	- + 0 + 0 0
1111 1111	+ 0 - + 0 0

- Bei der 8B/6T-Codierung (8 binary 6 ternary) wird ein Byte in einen 6T-Code umgewandelt. Jeder 6T-Code besteht aus 6 "Tri-State-Symbolen", die als "?", "0" und "+" notiert werden.
- Der 8B/6T Code wird in folgenden Ethernet Systemen verwendet:
  - 100Base-T4

Bild: 8B/6T Code

- Beim Codierungsverfahren 5-Level Pulse Amplitude Modulation (PAM5) wird pro Takt ein Symbol übermittelt, das einen von fünf verschiedenen Zuständen (?2, ?1, 0, +1, +2) darstellt.
- Mit jedem Symbol werden zwei Bits übertragen. Da es vier verschiedene 2-Bit-Gruppen ("00", "01", "10" und "11") gibt, bleibt noch ein Symbol übrig, das für die Fehlerbehandlung eingesetzt wird.

- Der PAM5 Code wird in folgenden Ethernet Systemen verwendet:
  - 100Base-T2
  - 1000BASE-T

Bild: PAM5 Code

- Bei der 8B/10B-Codierung werden 8-Bit lange binäre Sequenzen in 10-Bit Codegruppen umgewandelt. Dadurch erreicht man eine Gleichspannungsfreiheit und eine ausreichende Anzahl der Pegelwechsel für die Taktrückgewinnung.
- Der 8B/10B Code wird in folgenden Ethernet Systemen verwendet:
  - 1000Base-LX,
  - 1000Base-SX,
  - 1000Base-CX und
  - 10GBASE-LX4

Bild: 8B/10B Code

- Die 64B/66B-Codierung wird in 10GBASE Systemen verwendet. Bei der Umwandlung von 64 Bits zu 66 Bits erhält jede Gruppe eine Präambel, über die eine ständige Synchronisierung auf das ankommende Datenstrom sichergestellt ist. Dadurch werden Distanzen von bis zu 40 km ermöglicht.
- Der 64B/66B Code wird in folgenden Ethernet Systemen verwendet:
  - 10GBase-R
  - 10GBase-W

Bild: 64B/66BCode

Netzstruktur

Der Standard 100Base-T wurde für den Einsatz von Fast-Ethernet über eine Verkabelung mit UTP der Kategorien 3, 4 und 5 mit jeweils vier Adernpaaren definiert. Die Berücksichtigung von UTP Kategorie 3 nimmt besonders auf eine vorhandene 10-Base-T-Verkabelung Rücksicht. 100Base-T kennt mehrere Varianten mit verschiedenen Übertragungsmedien:

- 100Base-TX,
- 100Base-T2,
- 100Base-T4,
- 100Base-FX.

Alle 100Base-T-Netze haben gemeinsam, dass sie sternförmig aufgebaut sind und an einen zentralen Hub bzw. Switch angeschlossen werden. Aufgrund der Beibehaltung des CSMA/CD-Verfahrens sind nur geringe Entfernungen zulässig sowie Echtzeitanwendungen nicht anwendbar. Twisted-Pair-Kabel können Entfernungen bis maximal 100 m bei einer Dämpfung von 13 dB bei 12,5 MHz zulassen, während bei Glasfaser die Entfernungsbeschränkung 400 in beträgt. Die Limitierung bei TP (Twisted Pair) kann durch Hinzunahme eines Repeaters auf 200 m erweitert werden. Deshalb ist Fast-Ethernet auch nicht als Backbone-Technologie geeignet.

Für typische Arbeitsgruppenanwendungen sind bei 100Base-T zwei Repeaterarten definiert.

- Class-I-Repeater, der die Kopplung unterschiedlicher Physical-Layer-Implementierungen gestattet,
- Class-II-Repeater, der ausschließlich gleiche Verkabelungsarten erlaubt.

Unterstützt werden von 100Base-T drei Physical-Layer-Standards. 100Base-TX ist ein Verfahren, welches zwei UTP-Kabel (100 Ohm) der Kategorie 5 oder STP-Kabel (150 Ohm) nach Typ 1 benötigt. Die Daten werden mit einer MLT-Codierung (MLT-3, 4B/5B) codiert, wodurch sich die Datenrate auf 33,3 MHz reduziert. Dadurch können amerikanische und europäische EMV-Richtlinien eingehalten werden. Bei UTP wird der RJ-45-Stecker eingesetzt, während STP Sub-D9-Stecker verwendet, wodurch die Anschlusstechnik unabhängig von dem Übertragungsmedium ist. Die Basis bildet die PMD von FDDI mit CSMA/CD-Verfahren. Zusammenfassend kann man folgende Eigenschaften des 100Base-TX-Standards hervorheben:

- Basisband Signalisierung,
- Codierung: 4B/5B-Verfahren,
- Stern-Topologie,
- 4-adriges 100/150-Ohm-TP-Kabel der Kategorie 5,
- Maximale Segmentlänge: 100 m,
- Steckertyp: 8-poliger RJ-45-Stecker,
- Idle-Signale überwachen die Verbindungsstrecke.

100Base-T2
Das System basiert aus einfachem UTP-Kabel der Kategorie 3. Die Übertragung findet über zwei Leitungspaare statt, wodurch die Bezeichnung T2 entstand. Um die Aufgabe, 100 Mbit/s über Kategorie 3 auf zwei Leiterpaaren zu übertragen, lösen zu können, wurde die Puls Amplituden Modulation (PAM) eingesetzt. Das PAM5-Verfahren verwendet fünf verschiedene Pegel. Durch gleichzeitige Übertragung auf beiden Leitungen können bis zu 25 verschiedene Signalpunkte übertragen werden.

100Base-T4
Diese Version von Fast-Ethernet wurde für die Verkabelung mit UTP der Kategorie 3, 4 und 5 mit jeweils vier Adernpaaren definiert. Die Berücksichtigung von UTP der Kategorie 3 ist aufgrund der vorhandenen 10Base-T-Verkabelung in den Standard eingeflossen. Die Limitierung der oberen Grenzfrequenz auf 16 MHz bei UTP der Kategorie 3 macht eine Leitungscodierung nach Manchester unmöglich. Statt dessen beruht die Übertragung nach 100Base-T4 auf der Aufteilung der Gesamtübertragungsrate auf drei Level. Zur weiteren Reduzierung der auf den Leitern auftretenden höchsten Frequenz, kommt ein ternärer Leitungscode 8B6T zum Einsatz. Durch die Darstellung von je acht Bits durch sechs ternäre Symbole wird eine Reduzierung der Übertragungsfrequenz von 33,33 MHz auf 25 MHz je Adernpaar erreicht. Durch die 6-stellige ternäre Darstellung ergibt sich ein Alphabet mit insgesamt $3^6 = 729$ Codewörtern. Der eigentliche Transport der Daten benötigt aber nur $2^8 = 256$ Codewörter, wodurch eine Optimierung möglich wird. Das heißt, es wird eine Gleichstromfreiheit erreicht und empfangsseitig eine Rückgewinnung des Taktes. Hierzu erhält jedes Codewort mindestens zwei Flankenwechsel. Jedes einzelne ternäre Symbol innerhalb eines Codewortes repräsentiert dabei einen der Zustände -V, 0 oder +V. Das vierte Adernpaar dient für die Collision Detection. Die Datenübertragung erfolgt ausschließlich unidirektional. Eine Station kann entweder senden oder empfangen.

Die wichtigsten Eigenschaften des 100Base-T4-Standards lassen sich wie folgt zusammenfassen:
- Basisband Signalisierung,
- Codierung: 8B/6T-Verfahren,
- Stern-Topologie,
- 8-adriges 100 Ohm TP-Kabel der Kategorie 3, 4, 5,
- Maximale Segmentlänge: 100 m,
- Steckertyp: 8-poliger RJ-45-Stecker,
- Idle-Signale überwachen die Verbindungsstrecke.

100-Base-FX
Diese Version von Fast-Ethernet basiert auf den Spezifikationen der Technologie FDDI. Zur Übertragung kommen zwei Gradientenindex-Profilfasern mit 62,5/125 µm oder 50/125 µm zum Einsatz. Die Aufteilung der Fasern sowie die Codierung auf ihnen, entsprechen den von 100BaseTX bekannten Definitionen. Bauformen der Anschlussstecker sind ST-, SC- oder MIC-Stecker. Der SC-Stecker kann als Duplex-Stecker eingesetzt werden, wodurch eine maximale Faserlänge von 2000 m zwischen zwei Switches oder Bridges zulässig ist. Der MIC-Stecker wird dabei immer als M-Verbinder codiert. Normalerweise beträgt die Länge eines Glasfasersegments 400 m mit 570 ns Laufzeit. Als Codierung wird 4B/5B-Codierung und NRZ-I verwendet.

Folgende Eigenschaften lassen sich festhalten:
- Basisband Signalisierung,
- Codierung: 8B/6T-Verfahren,
- Stern-Topologie,
- 2-adriges LWL-Kabel mit 62,5/125 µm oder 50/125 µm,
- Maximale Segmentlänge: 2000 m,
- Steckertyp: ST-, SC- oder MIC-Stecker,
- Idle-Signale überwachen die Verbindungsstrecke.

Wie bereits angedeutet, ist das CSMA/CD-Verfahren nicht für die Übertragung isochroner Signale geeignet. Die Sendung einzelner Datenrahmen kann unterschiedlich verzögert werden. Daher garantiert CSMA/CD keine sichere Mindestbandbreite. Aus diesem Grund ist diese LAN-Variante nicht für zukünftige Systeme einsetzbar. Allerdings ist 100Base-T abwärtskompatibel zu älteren 10Base-T-LANs, so dass es sich einfach in bestehende Strukturen integrieren lässt. Außerdem kann statt eines einfachen Hubs bei 100Base-TX und 100-Base-FX auch ein Switch eingesetzt werden. Dadurch werden dedizierte Verbindungen zu den Endstationen ermöglicht. Weiterhin kann das zweite Adernpaar zum Vollduplexbetrieb eingesetzt werden, wodurch auf die Kollisionskontrolle verzichtet werden kann. In Kombination mit einem High-Performance-Switch ist 100Base-T deshalb ein erster Schritt in Richtung High-Speed-LAN.

Gigabit-Ethernet (IEEE 802.3z)

Durch neue Applikationen und wachsende Teilnehmerzahlen steigt kontinuierlich der Bedarf an Bandbreite im lokalen Netz. Aus diesem Grund hat man bei der IEEE die Arbeitsgruppe 802.3z Gigabit Task Force gegründet, die den neuen Standard Gigabit-Ethernet spezifiziert haben. Das Ziel dieses Standards ist es, Vollduplexübertragung mit 1 Gbit/s zu ermöglichen und dabei dieselben Datenrahmen wie Ethernet 10Base-T und Fast-Ethernet 100Base-T zu verwenden. Die Empfehlung enthält Protokoll Media Access Controller (MAC), Repeater und Physical Layer sowie komplexere Themen wie Flusskontrolle, Repeater-Architektur sowie Übertragung über UTP-Kupferleitung.

Gigabit Ethernet (auch 1000Base-X genannt)

Grundgedanke
- ursprüngliches Ziel: Beibehaltung des CSMA/CD-Verfahrens
- Ideale Ergänzung zu (Fast-) Ethernet zur Fortführung im Backbone-Bereich
- Einsatz auf Glasfaser und Kupferadern

Probleme
- Minimale Länge der Dateneinheiten von (Fast-) Ethernet zu klein, um Kollisionen zu erkennen.

Kompatibilität zu (Fast-) Ethernet möglich?
- Strenge Normen zur elektromagnetischen Verträglichkeit
- Elektro-physikalische Eigenschaften der Kupferadern: Übersprechen, Dämpfung, ...

Bild: Gigabit Ethernet

Dabei ist man bei der Standardisierung von 1000Base-T einen großen Schritt weitergekommen. Somit wird es wahrscheinlich zukünftig die Möglichkeit geben, Gigabit-Ethernet auch über Kupferkabel anzubieten. Dabei ist allerdings eine verbesserte Version von Kategorie 5 (bis 100 MHz) einzusetzen, da die Performance von CAT-5 nicht mehr als ausreichend bezeichnet werden darf. Zwei weitere Vorschläge wurden deshalb diskutiert, die Kupferkabel nach Kategorie 6 (bis 200 MHz) und Kategorie 7 (bis 600 MHz) betreffen.

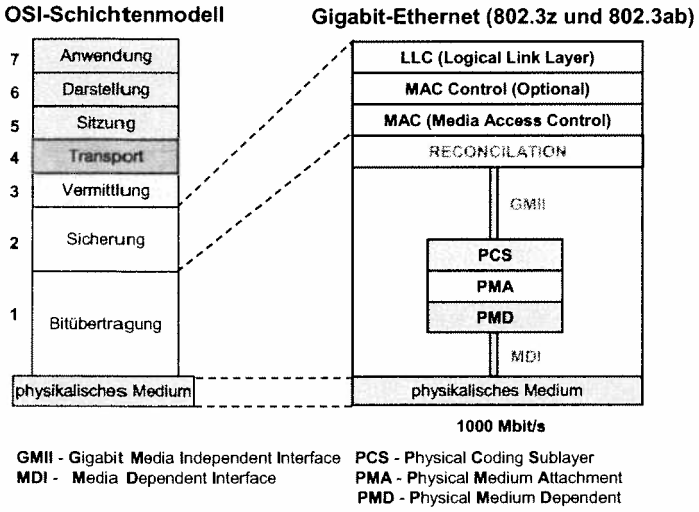


Bild: Gigabit-Ethernet Schichtenmodell

Weiterhin können mit dieser neuen Ethernet-Technik 1024 Stationen angeschlossen werden, wobei auch hier aufgrund der Abwärtskompatibilität immer noch die CSMA/CD-Technik verwendet wird. Die Komplexität ist gegenüber ATM gering, wodurch eine Integration in existierende Ethernet-Infrastrukturen mit Unterstützung aller Anwendungen und Betriebssysteme ohne höheren Einarbeitungsaufwand möglich ist. Durch Fast-Ethernet-Switches können zusätzlich Engpässe vermieden werden. Gigabit-Ethernet wird somit einen Migrationspfad zu vorhandenen Ethernet-Netzen eröffnen, so dass es sich als Backbone-Technik zur Kopplung von Fast-Ethernet-Switches eignet. Jedoch gibt es Beschränkungen bezüglich der Kabellänge. Gigabit-Ethernet überbrückt bisher maximal 100 m. Durch das CSMA/CD-Verfahren kann sich der Aktionsradius sogar auf 25 m verringern. Das ist für eine Backbone-Verkabelung entschieden zu wenig.

- Gigabit Media Independent Interface (GMII)
- Codierungsverfahren:
  - 8B/10B (1000Base-LX, 1000Base-SX, und 1000Base-CX)
  - PAM5 (1000Base-T)
- Übertragungsmedium:
  - Twinax (1000Base-CX)
  - 4 Paar TP Kat. 5 (1000Base-T)
  - Glasfaser – kurze Wellenlängen (1000Base-SX)
  - Glasfaser – lange Wellenlängen (1000Base-LX)

Bild: Besonderheiten des Gigabit-Ethernet

Aus diesem Grund wurde eine Erweiterung implementiert, die sich nicht konform zu Ethernet verhält. Dabei werden die Datenrahmen mit Hilfe einer Carrier Extension verlängert, die einer Veränderung der MAC-Spezifikation gleichkommt. Wenn die Rahmen in ein Low-Bandwidth-Ethernet gelangen, muss diese Protokollerweiterung wieder entfernt werden.

- Das GMII (Gigabit Media Independent Interface) stellt eine Erweiterung des im 100 Mbit/s Ethernet verwendeten MII dar.
- 8 Datenleitungen
- 125 MHz Taktfrequenz
- **Weitere Steuerungssignale:**
  - Kollisionssignal (COL)
  - Carrier Sense Signal (CRS)
  - Transmit Enable (TX_EN)
  - Transmit Error (TX_ER)
  - Receive Data Valid (RX_DV)
  - Receive Error (RX_ER)

Bild: Gigabit-Ethernet: GMII

Rahmenaufbau

Gigabit-Ethernet verwendet weiterhin den gleichen Rahmenaufbau und das gleiche Zugriffsverfahren wie seine Vorgänger. Der Gigabit Media Access Frame (GMAF) ist identisch mit der Struktur des klassischen Ethernets. Die sieben Byte der Präambel mit dem Bitmuster 1010 ... 10 dienen der Taktsynchronisation des Empfängers. Anschließend kommt das Feld Start Frame Delimiter (SFD), welches den Rahmenbegrenzer darstellt. Er besitzt ein unverwechselbares Bitmuster (10101011). Die MAC-Quellen- und -Zieladresse schließt sich an. Sie werden wiederum durch das Feld Längetyp begrenzt. Anschließend kommen die Nutzdaten, die auch hier 1500 Byte betragen können. Das Padding-Byte dient wie immer zum Auffüllen der Daten auf eine gerade Anzahl von Bytes und muss nicht immer vorhanden sein. Abschließend wird noch eine Prüfsumme aus dem Datenfeld gebildet, um Fehler erkennen zu können. Aufgrund von Problemen bei der Erkennung von Kollisionen und Effizienz der Bandbreite, mussten aber doch folgende Erweiterungen eingefügt werden:

- Carrier Extension,
- Packet-Bursting,
- Multiple Link Segments,
- Buffer Distribution,
- Jumbo-Rahmen.

Bei Gigabit-Ethernet werden die Wettbewerbsintervalle von 512 Bit (64 Byte) auf 512 Byte heraufgesetzt, um Kollisionen noch erkennen zu können. Die minimale Rahmenlänge $LF_{min} = v \times T$ ergibt sich daraus, dass eine Station beim CSMA/CD-Verfahren nach erfolgter Kollision solange senden muss, bis das Medium mit einem Rahmen voll ausgefüllt ist. Bei Ethernet ergibt einer Datenrate von 10 Mbit/s und einer Signallaufzeit T = 51,2 µs eine minimale Rahmenlänge von 64 Byte. Die Merkmalswerte sind austauschbar, sie müssen aber entsprechend angepasst werden. Deshalb hat man die Carrier Extension spezifiziert, die das Ethernet-Rahmen auf diese Mindestlänge von 512 Byte auffüllt. Dies beinhaltet jedoch einen Nachteil bei kleineren Rahmengrößen, da ein Carrier-Signal die Leitung während des gesamten.Wettbewerbsintervalls blockiert, obwohl die Datenrahmen bereits in einer geringeren Zeit übertragen wurden. Bei einer Rahmengröße von z.B. 64 Byte werden 448 Byte verschwendet, wodurch die Leistung des Netzes im schlimmsten Fall auf Fast-Ethernet-Niveau sinken könnte. Eigentlich wären 640 Byte notwendig gewesen, um Kollisionen in jedem Fall erkennen zu können. Aufgrund des damit verbundenen Leistungsabfalls hat man aber auf diese Rahmengröße verzichtet. Bisherige Untersuchungen zeigen jedoch, dass die minimale Rahmengröße zwischen 200 < 640 Byte liegt, so dass dieser Effekt nicht so stark ins Gewicht fällt. Diese Rahmenverlängerung hat jedoch einen Engpass zur Folge, da je nach Rahmenlänge nur von einer Leistungssteigerung mit dem Faktor 2 bis 9 gegenüber Fast-Ethernet ausgegangen werden kann. Ob dieses Verfahren in der Praxis noch weitere Probleme verursachen wird, bleibt abzuwarten. Auf jeden Fall wird sich die Verzögerungszeit weiter erhöhen.

Frame-Bursting

Ein weiteres Verfahren bei Gigabit-Ethernet ist das Frame-Bursting, bei dem der Nutzungsgrad der Bandbreite durch die Carrier Extension weiter erhöht wird. Durch die Sendung von zusätzlichen Rahmen nach Ablauf des störungsfreien Wettbewerbsintervalles soll der Datendurchsatz bei Minimumrahmen auf 300 Mbit/s oder bei typischen Arbeitsgruppenanwendungen auf 700 Mbit/s gesteigert werden. Zusätzlich sorgt die Virtual Collision für eine effiziente Ausnutzung der Bandbreite. Da jede Kollision verlorene Bandbreite bedeutet, soll bei einer auftretenden Kollision der Repeater der erste Rahmen retten und unbeschädigt verbreiten, die anderen beteiligten Rahmen jedoch unterdrücken. Ein Gigabit Buffer Repeater geht noch einen Schritt weiter. In diesem Vorschlag geht man davon aus, dass Vollduplex-Verbindungen vorhanden und die Repeater mit Zwischenpuffern ausgestattet sind. Dadurch wird CSMA/CD nur als Zugangskontrolle für die jeweilige Verbindung dienen und nicht für das gesamte Netz. Dies ist bisher allerdings nur ein Vorschlag, der noch nicht in die Praxis umgesetzt wurde.

Trunking

Verbindungen zusammen zu schalten. Diese können dann parallel genutzt werden. So eine Funktion ist bei großen Netzen notwendig und wird dabei auch als Trunking bezeichnet. Der Begriff ist aber bei IEEE 802.3 bereits belegt, so dass man einfach diesen neuen Begriff geschaffen hat. Bisher gab es bei der Parallelschaltung mehrerer Ethernet-Verbindungen Probleme, da der Spanning-Tree-Algorithmus bis auf eine Leitung alle anderen deaktiviert hat. Spanning Tree ist demnach nicht für eine Skalierung des Bandbreitenbedarfs geeignet. Die Arbeitsgruppe IEEE 802.3ab hat sich des Trunking Problems nun ange-

nommen und wird sie nach seiner Lösung verabschieden. Multiple Link Segments wird dabei zwischen den höheren Ebenen bzw. der LLC-Schicht eingefügt und sorgt dafür, dass über mehrere MAC-Layer parallel gearbeitet werden kann. Das heißt, man bekommt einerseits Leitungsredundanz, so dass nach Ausfall einer Verbindung mit geringerer Kapazität weitergearbeitet werden kann. Anderseits kann eine Art Skalierbarkeit erreicht werden. Diese ist allerdings nicht fein regulierbar, sondern nur grob im Vollduplex-Modus einstellbar (z.B. von 100 auf 200 Mbit/s). Bisher sind nur proprietäre Ansätze realisiert.

Buffer Distribution
Dieses Verfahren soll das Problem der Einschränkung von Bandbreite durch die Repeater kompensieren. Dafür setzt man einen sogenannten Buffer Distributor ein, der ähnlich wie ein Switch in der Lage ist, Rahmen zu puffern, aber im Gegensatz zum Switch keine Rahmenanalyse durchführen kann. Buffered Repeater waren auch im normalen Ethernet vorhanden, bis man in der Lage war, Bridges herzustellen. Aus diesem Grund ist dies auch nur als ein kurzer Übergang zu verstehen, bis man aus wirtschaftlichen Gründen sofort zum Switch wechselt.

Jumbo-Rahmen
Die sogenannten Jumbo-Rahmen sind für die Skalierbarkeit von Gigabit-Ethernet eingeführt worden. Das heißt, es können unterschiedliche Übertragungsraten durch den Einsatz von 10/100/1000 Mbit/s-Rahmen eingestellt werden. Man kann dadurch Gigabit-Ethernet nicht fein skalieren. Zusätzlich können große Ethernet-Rahmen natürlich auch echtzeit Datenströme wie IP-Telephonie und IP-Multimedia im Netz behindern, da sich diese Rahmen nicht fragmentieren lassen.

Netzanschlüsse und Teilschichten
In der Architektur von Gigabit-Ethernet lassen sich unterschiedliche physikalische Schnittstellen ausmachen. Da die ersten drei Möglichkeiten auf dem Standard Fibre Channel (FC) aufbauen, dieser aber nicht die Verwendung von CAT-5-Verkabelung vorsieht, müssen bei UTP und STP erst die Definitionen neu erarbeitet werden, was erhebliche Verzögerungen nach sich zieht. Deshalb ist ein weiterer Arbeitskreis mit dem Namen IEEE 802.3ab gegründet worden, der sich mit effektiveren Codierungsverfahren auseinandersetzt, da die normale digitale Übertragung nicht mehr ausreicht. Die Bitübertragungsschicht besteht aus der Schnittstelle Medium Dependent Interface (MDI) mit den darüberliegenden Modulen Physical Medium Dependent (PMD), dem Physical Medium Attachment (PMA) und dem Physical Coding Sublayer (PCS), die die Anpassung der physikalischen Schnittstellen vornehmen. Die Schnittstelle zwischen MAC- und Bitübertragungsschicht heißt Gigabit Media Independent Interface (GMII) und ist medienunabhängig. Beide Schichten werden normalerweise durch unterschiedliche elektronische Bausteine umgesetzt, so dass sich Anpassungen an die verschiedenen Übertragungsmedien vornehmen lassen.

GMII-Schnittstelle
Ähnlich wie die MII-Schnittstelle bei Fast-Ethernet dient die GMII-Schnittstelle neben dem Anschluss verschiedener Medien auch zur automatischen Erkennung des Mediums und dem Austausch von Daten über Zustand und Eigenschaften der aktuellen Verbindung sowie Statistiken über den Datenverkehr zwischen MAC und Bitübertragungsschicht. Dazu wurde die Auto-Negotiation entwickelt, die anders funktioniert als bei Fast-Ethernet. Bei Gigabit-Ethernet dient sie dem Informationsaustausch über die verwendeten Medien und dem Aushandeln der Link-Eigenschaften für den Betrieb mit dieser Datenrate. Diese Daten werden mit speziellen Code-Gruppen zwischen zwei Link-Partnern kommuniziert. Ein weiteres Modul (RC = Reconciliation) nimmt die endgültige Umsetzung zur MAC-Schicht vor. Die MAC-Schicht regelt bekanntlich den Zugriff auf das Übertragungsmedium und legt die Rahmenformate fest. Die Spezifikation wurde bis auf die Carrier Extension und das Packet Bursting von den Vorgängern mit 10/100 Mbit/s übernommen.

Ein Problem bei Gigabit-Ethernet ist die Einbeziehung von bestehender Kabelinfrastruktur. Neben der fragwürdigen Unterstützung von Kupferkabeln können Fehlübertragungen auch bei Übertragungen über Multimode-Fasern existieren. Der Grund ist, dass bei einigen Multimode-Fasern bestimmte Modengruppen dominieren, wenn als Lichtquelle Laser verwendet wird. Dadurch sind Fehlinterpretationen des Empfangssignals zu befürchten. Zehn Prozent der installierten Multimode-Infrastruktur sollen immerhin dieses Problem besitzen. Als Problemlösung wird das Conditional Launching angeboten, welches die ausgeglichene Anregung aller Moden ermöglichen soll. Dafür sind allerdings zusätzliche Kompensationsmodule erforderlich, die sich im Transceiver realisieren lassen.

VLAN
Weiterhin die VLAN-Spezifikation IEEE 802.1q im Standard Gigabit-Ethernet fest integriert. Der Standard beschreibt u.a. ein Tagging-Verfahren, das dem Datenrahmen eine 4 Byte lange VLAN-Kennzeichnung hinzufügt. Dadurch vergrößert sich die maximale Rahmenlänge bei Ethernet von 1518 Byte auf 1522 Byte. Das heißt, dass eine Erweiterung der MAC-Spezifikationen vorgenommen werden musste. VLAN-Datenrahmen lassen sich durch das bisherige Längen-/Typenfeld identifizieren, welches die definierte hexadezimale Typ-Identifikation von 8100 enthält. Die beiden nachfolgenden Bytes beinhalten die VLAN-Identifizierung und enthalten zusätzlich drei Bits zur Prioritätsvergabe. Diese Priorität ist nach der Spezifikation IEEE 802.1p bereits festgelegt und ermöglicht den Datenverkehr in Prioritätsklassen einzuteilen, wodurch eine Art Dienstgüte unterstützt wird (Class-of-Service). Das ursprüngliche Längen-Typenfeld befindet sich inklusive des regulären Datenrahmen hinter der eingeschobenen VLAN-Kennzeichnung. Das letzte Feld Prüfsumme überprüft das gesamte Datenrahmen auf Fehler. Diese VLAN-Spezifikation ist unabhängig von der Datenrate und wird ebenso für 10-Mbit/s-Ethernet eingesetzt werden.

Netzstruktur

Frequenzen an der Grenze zum Mikrowellenspektrum lassen sich schwer auf vorhandene Kupferverkabelung übertragen. Zu großen Abstrahlungen sind die größten Probleme, die nicht nur die elektromagnetische Verträglichkeit betreffen, sondern auch zu Übertragungsfehlern führen. Dämpfung und Frequenzverzerrung tun ein übliches, um die Rückgewinnung des Sendesignals zu erschweren oder gar unmöglich werden zu lassen. Ethernet verwendet ein Basisbandverfahren, welches nicht für die Übertragung auf Kupferbasis mit 1 Gbit/s geeignet ist. Bei 1000Base-CX wird zwar die Datenrate mit 1250 MBaud transportiert; das setzt aber eine sehr aufwendig abgeschirmte Verkabelung voraus, die als maximale Entfernung auch nur 25 m zulässt.

Variante	Medium
1000BASE-T	4 Paare UTP, Kategorie 5, Distanz bis 100 m. Leitungscodierung 4D-PAM5
1000BASE-CX	Twinax-Kupferkabel bis 25 m. Je ein Paar pro Richting, Leitungscodierung 8B10B
1000BASE-SX	Multimode-Glasfaser (770-860 nm), bei Kerndurchmesser 50 µm: Distanz 440 m, bei Kerndurchmesser 62,5 µm: Distanz 260 m, Leitungscodierung 8B10B
1000BASE-LX	Multimode-Glasfaser (1270-1355 nm), Distanz bis 550 m bei Kerndurchmesser 50 oder 62.5 µm, Distanz bis 5 km bei Singlemode-Glasfaser Kerndurchmesser 10 µm, Leitungscodierung 8B10B

T IEEE 802.3 access protocol and Format
X FDDI optical Interface
S Short range fiber
L Long range fiber
C Copper

FDDI Fiber Distributed Data Interface

Bild: Gigabit Ethernet-Verkabelung

Für Gigabit-Ethernet sind die folgenden Verkabelungen spezifiziert worden:

- 1000Base-T,
- 1000Base-CX,
- 1000Base-SX,
- 1000Base-LX.

Um die Probleme der hohen Bandbreite bei Basisbandübertragung im Kupferbereich in den Griff zu bekommen, hat sich die Arbeitsgruppe IEEE 802.3ab zusammengefunden. Erreichen will man eine Entfernungsbeschränkung von 100 in mit normalem UTP-Kabel nach Kategorie 5. Um dies zu erreichen, sendet man nicht mit der vollen, sondern nur mit der nötigen Baudrate und verwendet zusätzlich komplexe Codier- und Übertragungsverfahren. Aus Sicht der Informationstheorie ist die Kapazität des Übertragungskanals durch das Basisbandverfahren noch nicht ausgeschöpft. Heutige analoge Modems, die Datenraten von über 56 kbit/s über eine Telefonleitung, die ursprünglich für 2400 Hz ausgelegt war, transportieren, belegen dies. Durch Phasen- und Amplitudenmodulation, die mit Hilfe von Digital Signal Processors (DSPs) mittels Algorithmen in Echtzeit durchgeführt wird, kann die Übertragung hoher Datenraten ermöglicht werden. Der 1000Base-T-Transceiver ähnelt einem Modem in dieser Hinsicht, wobei aber wesentlich höhere Datenraten zu beherrschen sind.

1000Base-CX

Der Kupferstandard 1000Base-CX ist eher ein konventioneller Ansatz, wenn man ihn mit dem eben beschriebenen vergleicht. Hier nimmt man ein sehr aufwendiges 150-Ohm-Twinax-Kabel, dass eine Entfernungsbeschränkung von 25 m besitzt. Dabei wird keine zusätzliche Codierung vorgenommen. Die Daten werden direkt über einen entsprechenden Pulstransformator auf ein Kabelpärchen getrieben. Dies geschieht mit einem Pegel von ± 1V. Zwei Steckertypen sind bislang ausgesucht worden. Der DB9 mit Anschlussbelegung nach Token Ring und einen Stecker für 1000Base-CX ausgewählt werden. Nachteilig sind an der STP-Verkabelung die unhandlichen Twinax-Kabel, die eine große Verbreitung nicht sicherstellen werden. Nischenbereiche wird es aber in räumlich begrenzten Rechenzentren oder Collapsed Backbones geben.

1000Base-SX

Der Standard 1000Base-SX ist hingegen für die Übertragung über Short-Wavelength-Duplex-Multimode-Glasfaser vorgesehen. Hier wird mit einer Wellenlänge von 850 nm (nahes Infrarot) operiert. Geräte mit unterschiedlichen Wellenlängen können sich nicht gegenseitig sehen. Bewegliche Transceiver hinsichtlich der Wellenlänge sind zwar denkbar, aber ungünstig bezüglich der Kosten. Deshalb wird es inkompatible Glasfaseranschlüsse geben, die einmal auf 850 nm und ebenfalls auf 1300 nm basieren. Moderne Multimode Fiber besitzt ein Bandbreitenlängenprodukt = 600 - 1000 MHz x Kilometer. Wenn man die nicht ganz richtige Annahme, dass die Baudrate gleich der Übertragungsfrequenz ist, in Betracht zieht, genügen hochwertige Multimode-Glasfasern für Reichweiten bis zu 800 in. Dämpfung spielt hierbei eine eher untergeordnete Rolle. Die Dispersionseffekte sind entscheidend und verantwortlich für Verzerrungen der Pulsform auf dem Weg zum Empfänger, die durch unterschiedliche Wellenlängen und Reflexionspfade des Lichtes in der Faser entstehen.

1000Base-LX

1000Base-LX wird auf einer Wellenlänge von 1300 nm basieren und auf eine maximale Entfernung von bis zu 3 km aufweisen. Trotz der erheblich geringeren Verzerrungen ist auch bei der Long-Wavelength-Duplex Multi-/Monomode-Glasfaser keine größere Entfernung möglich. Bei Multimode sind es sogar nur noch maximal 550 m. Als Steckertypen kommt hauptsächlich der Duplex-SC-Stecker in Frage, da er sich gegenüber seiner Konkurrenz durchgesetzt hat. Dabei relativiert sich die beschriebene Entfernungsbeschränkung ein wenig. Aufgrund der vorhandenen Fibre-Channel-Basisspezifikationen von Gi-

gabit-Ethernet sind inzwischen auch proprietär 50 km Entfernungen Punkt-zu-Punkt einsetzbar. Der Standard enthält aber die dargestellten Werte, so dass nur bei Einsatz eines einzelnen Herstellers diese Möglichkeit gegeben wäre.

Übertragungseigenschaften
Die Übertragung dieser hohen Frequenzen hat ebenfalls auf die Glasfaser einen Einfluss. Die verwendeten Laserdioden bei Gigabit-Ethernet regen mehrere Moden in einer Faser an, die unterschiedlichen Brechungen folgen. Diese Pfade können verschiedene Längen besitzen, die sich auf die Verzögerungszeit auswirken. Im schlechtesten Fall kann aus einem Lichtimpuls zwei unabhängige Impulse resultieren. Man nimmt als Ursache eine konvexe Eingangsform des Kabels an. Monomode-Fasern und Kupferleitungen besitzen dieses Problem nicht. Bei 1000Base-SX hat man durch den Einsatz einer konkaven Linse vor dem Laserausgang diesen Effekt kompensiert. Bei 1000Base-LX wird der Einsatz eines zusätzlichen Patch-Kabels empfohlen. Zusätzliche Messungen sollten auf jeden Fall bei Verwendung von Multimode Glasfaser erfolgen, wenn man störende Dispersion (Streuung) vermeiden möchte.

Um Bandbreite im Gigabitbereich letztendlich aber auch bei Kupferverkabelung zur Verfügung stellen zu können, sind folgende Verfahren zusätzlich notwendig geworden:

- Parallelverarbeitung,
- Echo Cancellation,
- Master-Slave-Verfahren,
- Multi-Level-Codierung,
- Trellis-Codierung,
- Scrambling,
- Adaptive Equalization,
- Next Cancellation,
- Baseline-Wander-Korrektur.

Gigabit-Ethernet verwendet alle Adernpaare eines UTP-Kabels, wodurch vier simultane Übertragungskanäle vorhanden sind. Jeder Übertragungskanal wird mit 250 Mbit/s belastet und muss parallel bearbeitet werden, um eine Gesamtübertragungsrate von 1 Gbit/s zu erreichen.

Echo Cancellation löst hingegen das Problem des Übersprechens bzw. der Wiedergewinnung des Signals. Dadurch, dass alle Leitungen gleichzeitig und bidirektional genutzt werden, mischen sich die Signale beider Richtungen. Es gibt vier Adernpaare und die Transceiver (T) und Receiver (R) an beiden Kabelenden, die die gleichen Übertragungsleitung benutzen. Durch Echo Cancellation ist der Receiver in der Lage, sein eigenes Signal aus dem Gesamtsignal zu subtrahieren, wodurch die Signale wieder getrennt werden.

Ein weiteres Problem ergibt sich aus der Taktung der Empfänger, da eine bidirektionale Übertragung vorliegt. Die unterschiedlichen Takte würden sich auf einer einzelnen Leitung überlagern. Aus diesem Grund hat man bei Gigabit-Ethernet das Master-SlaveVerfahren verwendet. Der Repeater oder Switch muss dabei die Aufgabe des Masters übernehmen und den Takt vorgeben. Alle angeschlossenen Stationen sind der Slave. Sie übernehmen den gesendeten Takt für die Rückgewinnung des Sendesignals und verwenden ihn erneut für das eigene Sendesignal. Dadurch sind alle Slave-Stationen sende- und empfangssynchron. Bei Verbindungen zwischen Repeatern und Switches muss allerdings vorher vereinbart werden, wer die Rolle des Masters übernimmt. Durch das Protokoll Auto-Negotiation wird dies beim Verbindungsaufbau festgelegt. Wer welche Rolle übernimmt, wird zufällig bestimmt.

Scrambling löst das wichtigste Problem der hohen Frequenzen: das Abstrahlen. Durch Verwürfelung werden die Frequenzspitzen geglättet, wodurch die Abstrahlung der Leitung wesentlich verringert wird. Der dafür ausgewählte Algorithmus arbeitet auf der Bitübertragungsschicht, wodurch der gesendete unverwürfelte Datenstrom mit einer binären Datensequenz Modulo 2 addiert wird. Es wird somit eine Art Verschlüsselung angewandt, die einen 2047-Bit Schlüssel besitzt und nach folgender Funktion berechnet werden kann:

$gm(x) = 1 + x^{13} + x^{33}$ Master

$gS(X) = 1 + X^{20} + X^{33}$ Slave

Die Endstationen bekommen von diesem Verfahren nichts mit, da das Scrambling praktisch im Hintergrund läuft. Die Synchronisation der gesendeten Signale verhält sich natürlich umständlicher. Durch Start-/Stop-Sequenzen will man das aber in den Griff bekommen.

Die Adaptive Equalization bzw. Entzerrung ist zur Kleinhaltung der Dämpfung entscheidend. Auf der Übertragungsleitung, die aufgrund ihrer Frequenzabhängigkeit wie ein Tiefpass funktioniert, werden die tiefen Frequenzen weniger gedämpft als die hohen. Dadurch verliert ein Signal zunehmend seine ursprüngliche Form. Um die geringste Abstrahlung und Dämpfung zu erreichen, wird der Ausgang eines 1000Base-CX-Transcelvers einer speziellen Filterung unterzogen. Durch den Partial

Response Filter (PRF) werden drei Viertel des neuen Symbols und ein Viertel des vorangegangenen Symbols addiert. Danach wird das NEXT aus dem Signal herausgefiltert. Durch die Trellis-Codierung kann dann wiederum keine Gleichspannung auf dem Kabel entstehen. Innerhalb einer Codegruppe kann aber trotzdem eine kleine Verschiebung des Bezugspegels auftreten. Durch die Baseline Wander Correction wird dies über DSPs verhindert.

Codierung

Gigabit-Ethernet (IEEE 802.3z) muss aufgrund der hohen Datenrate ein effizientes Codierverfahren verwenden, wenn es die Nutzdaten ohne Fehler übertragen möchte. Um die Baudrate weiter zu senken, wurde deshalb bei 1000Base-T die Multi-Level-Codierung eingeführt. Bereits Fast-Ethernet führte mit MLT-3 drei Zustände zur Codierung ein, um die Übertragungsfrequenzen auf dem Kabel zu halbieren. 1000Base-T verwendet hingegen fünf Level (-2V, -1V, 0V, +1V, +2V). Dadurch könnte man theoretisch bis zu 2322 binäre Bits pro Takt senden. Zusammengefasst ergeben sich daraus 9288 Bits, die wiederum aus 625 verschiedenen Codes pro Takt resultieren (5 4 = 625). Durch das PAM5-Verfahren werden nun über doppelt soviel Codes zur Verfügung gestellt, wie letztendlich notwendig wäre, da man 8 Bits pro Takt sendet und entsprechend 2^8 = 256 Codes daraus resultieren. Das ist aber beabsichtigt, da diese Redundanz zur Steigerung der Übertragungsqualität notwendig ist und für die Trellis-Codierung verwendet wird.

Die Trellis-Codierung benutzt 8 Datenbits, die mit einem Parity Bit versehen sind. Diese werden auf die fünf Level der vier Übertragungsleitungen verteilt. Es entsteht ein vierdimensionaler Leitungscode mit acht Zuständen, welcher folgende Eigenschaften besitzt:

- Codesequenzen auf der Leitung sind so gewählt, dass die daraus resultierenden Pegelwechsel immer größer sind als bei einer nur zufällig erfolgten Auswahl.
- Das Signal-/Rauschverhältnis verbessert sich dadurch um 5,2 dB.
- Das zusätzliche Parity Bit wird zur Fehlererkennung verwendet.

Außerdem wird ein ähnliches Verfahren wie bei FDDI und Fast-Ethernet für die Codierung genutzt. Es handelt sich um die 8B/10B Codierung, die noch effizienter vorgeht als der Vorgänger 4B/5B. Um sich das Verfahren vor Augen zu führen, kann man sich eine längere Taktsequenz vorstellen, die ohne Taktinformation auskommt. Daraus kann kein Empfänger einen notwendigen Takt rekonstruieren, geschweige denn Anfang oder Ende eines Symbols oder Rahmens ermitteln. Das 8B/10B-Verfahren codiert nun jeweils 1-ByteDaten geschickt auf 10 Bits um, so dass niemals weniger als vier und mehr als sieben Wechsel pro Symbol auftreten können. Da man auf diese Weise die Lauflängen der Nullen und Einsen limitiert, wird ein solcher Code auch Run Length Limited genannt. Die maximale Baudrate dieses Verfahrens liegt bei den erwähnten 1250 MBaud.

Der gewonnene Coderaum wird nun genutzt, um weitere Code-Gruppen zu definieren. Diese speziellen Symbole können nicht mit Datensymbolen verwechselt werden, da sie alleine oder in Kombination mit ein bis drei Datensymbolen zu sogenannten Ordered Sets zusammengefasst werden. Diese werden als Marken für Carrier-Extension, Idle, Start-/End-of-Packet und Konfiguration verwendet. Alle anderen Bitkombinationen, für die es keine eindeutige Definition als Daten oder Special-Code-Gruppen gibt, zählen als Fehler. Da solche Symbole eigentlich nicht auftreten dürfen, kann man an der Häufigkeit feststellen, wie die Qualität der Verbindung bzw. des Netzes ist.

Funktionsweise

Es bleibt zu beachten, dass auch hier das CSMA/CD-Verfahren verwendet wird, wodurch Datenkollisionen im Halbduplexbetrieb entstehen können. Eine wirkliche Alternative stellt nur der Vollduplexbetrieb dar, der die Bandbreite verdoppelt und Kollisionen durch dedizierte Verbindungen vermeidet. Allerdings kommt kann es dann zu Verarbeitungsproblemen im Switch kommen, da hier alle Verbindungen zusammenlaufen. Es findet somit eine Verlagerung statt. Verwendet werden hier dann Punkt-zu-Punkt-Verbindungen, die über die Switches gesteuert werden. Zusammen mit der Vollduplex-Technologie wird auch ein einfaches Flusssteuerungsverfahren angeboten, welches auf einem Pause-Mechanismus basiert. Das heißt, die empfangene Station kann den Sender durch Aussenden eines XOFF-Rahmens beeinflussen. Das heißt, kurz vor Überlastung eines Eingangs-/Ausgangspuffers im Switch wird eine Nachricht an den Sender abgeschickt. Der Sender verzögert oder stoppt anschließend den Datentransport für den im Rahmen angegebenen Zeitraum. XOFF-Rahmen der Dauer 0 können längere Pausen aufheben. Dieses wird in dem Standard IEEE 802.1x festgehalten. Damit diese einfache Flussregelung zum Einsatz kommen kann, müssen natürlich alle Switches diesen Standard unterstützen.

10Gigabit-Ethernet (10GbE)

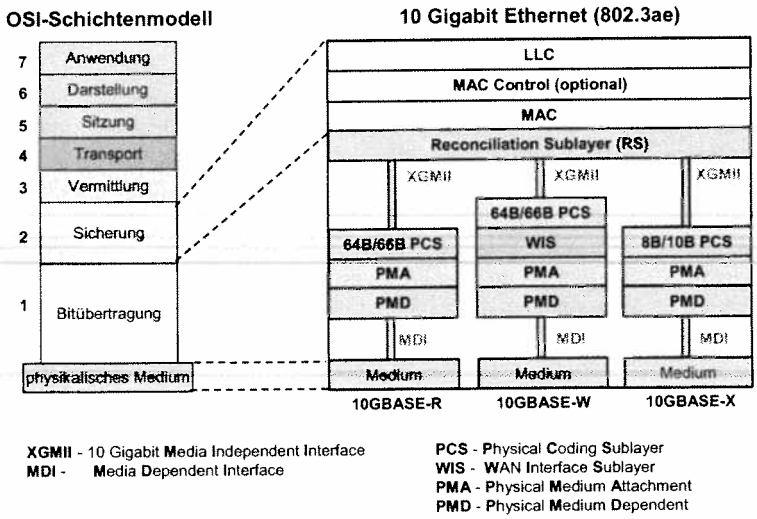


Bild: 10 Gigabit-Ethernet Schichtenmodell

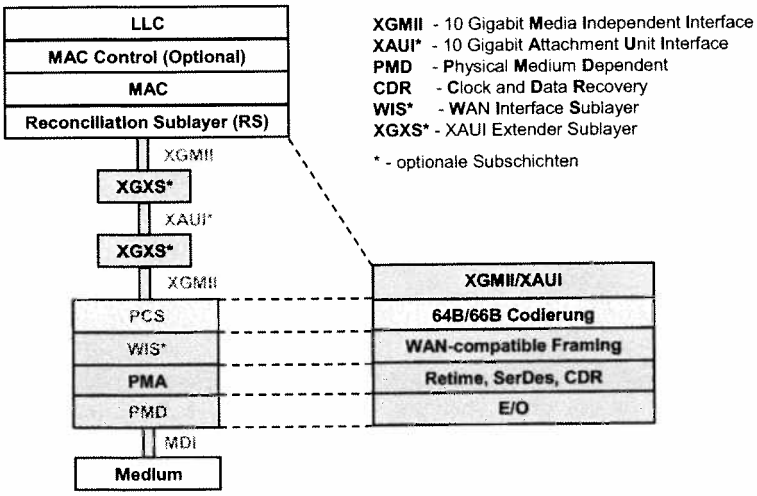


Bild: 10Gigabit-Ethernet: XAUI

IEEE 802.3ae:

- MAC: einfach nur Ethernet
  - 802.3 Rahmenformat und -Größe wird beibehalten
  - Nur Vollduplex Modus
  - Steigerung der Datenrate auf 10 Gbit/s für LAN PHY oder 9.58464 Gbit/s für WAN PHY
- PHY: LAN und WAN PHYs
  - LAN-PHY verwendet nur einfache Codierungsverfahren (8B/10B und 64B/66B) für die Datenübertragung.
  - Im WAN-PHY wird ein SONET Framing-Subschicht eingefügt.
- PMD: nur optisches Medium möglich:
  - 850 nm auf MMF to 65m
  - 1310 nm (4 Wellenlängen WDM bis 300 m auf MMF oder bis 10 km auf SMF)
  - 1310 nm auf SMF bis 10 km
  - 1550 nm auf SMF bis 40 km

Bild: 10Gigabit-Ethernet: Besonderheiten

- XGMII (10 Gigabit Media Independent Interface) ist eine 74-Bit breite Schnittstelle mit 32-Bit-Datenpfad zum jeweiligen Senden und Empfangen der Daten (Informationsaustausch) zwischen der MAC- und Bitübertragungsschicht.
- XGMII kann eine maximale physikalische Länge von nur 7 cm überbrücken. Eine Verlängerung erfolgt über das XAUI.
- XAUI (10 Gigabit Attachment Unit Interface) ist eine vereinfachte Erweiterung der XGMII-Schnittstelle die mit nur 16 Leitungen aus kommt.
- XAUI wird von vier seriellen, selbsttaktenden Verbindungen mit einer Bandbreite von je 2,5 Gbit/s realisiert.

Bild: 10Gigabit-Ethernet: XGMII

Variante	Medium
10GBASE-LX	Singlemode-Glasfaser, 1310 nm, Codierung 64B66B für LAN, bis 10 km
10GBASE-EX	Singlemode-Glasfaser, 1550 nm, Codierung 64B66B für LAN, bis 40 km
10GBASE-LW	Singlemode-Glasfaser, 1310 nm, Codierung 64B66B für WAN, bis 10 km
10GBASE-EW	Singlemode-Glasfaser, 1550 nm, Codierung 64B66B für WAN, bis 40 km

X FDDI optical Interface
W Wide Area
L Long range fiber
E Extended range fiber

FDDI Fiber Distributed Data Interface

Bild: 10 Gigabit-Ethernet-Anschlüsse

Glasfasertyp / 10 GbE System	MMF 62.5		MMF 50			SMF
SR/SW - 850 nm	28 m	35 m	69 m	86 m	300 m	-
LR/LW - 1310 nm	-	-	-	-	-	10 km
ER/EW - 1550 nm	-	-	-	-	-	40 km
LX4 - 1310 nm	300 m	300 m	240 m	300 m	-	10 km

Bild: 10GbE: Maximale Übertragungsstrecke

Der 10-Gigabit-Ethernet-Standard sollte kompatibel zu einer Vielzahl anderer Standards bleiben, darunter:

- IEEE802.3-Teilstandards wie 802.1p (Multicast), 802.3q (VLAN) und 802.3ad (Link Aggregation).
- IETF-Standards wie Simple Network Management Protocol (SNMP), Multi-Protocol Label Switching (MPLS) und Remote Monitoring for Ethernet (RMON).
- Standards aus dem OSI-Umfeld (Open Systems Interconnection).
- Kompatibilität mit SONET/SDH Netzen im WAN-Bereich.

Bild: 10Gigabit-Ethernet: Kompatibilität

1 Gigabit Ethernet (802.3z)

- CSMA/CD + Vollduplex
- Glasfaser oder Kupfer
- Einfluss von Fibre Channel PMDs
- 8B/10B Codierung
- Unterstütz werden lokale Netze bis 5 km

10 Gigabit Ethernet (802.3ae)

- Nur Vollduplex
- Nur Glasfaser
- Neue PMDs
- Neue Codierungsverfahren
- Unterstützt werden lokale Netze bis 40 km
- Verwendung von SONET/SDH

PMD - Physical Medium Dependent

Bild: Vergleich zwischen GbE und 10GbE

10 GbE in Service Provider Zentren und unternehmens-internen LANs
- Switch zu Switch
- Switch zu Server
- Datenzentren
- Zwischen Gebäuden

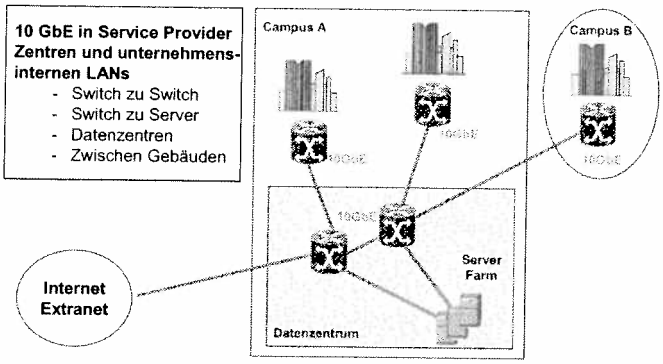


Bild: 10 GbE: LAN-Anwendungen

Unternehmen:
- 10 GbE ermöglicht eine Verbindung von Gebäuden mit einer direkten Fernwartung und -Backup.

Service Providers:
- 10 GbE bietet sehr hohe Datenrate um wenig Geld (ein gutes Preis/Leistungsverhältniss vom Low-Cost 10 GbE).

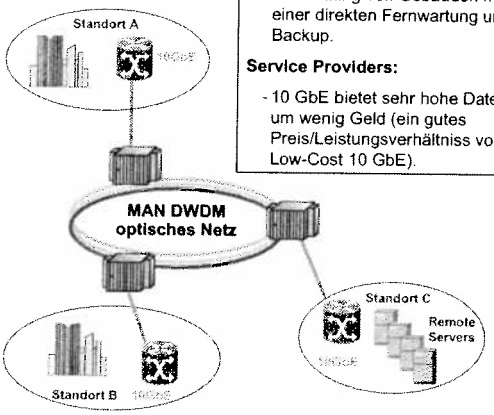


Bild: 10 GbE: MAN-Anwendungen

- Eine direkte Anbindung an das optische Backbone
- Kompatibilität mit bereits installierten SONET/SDH Netzen

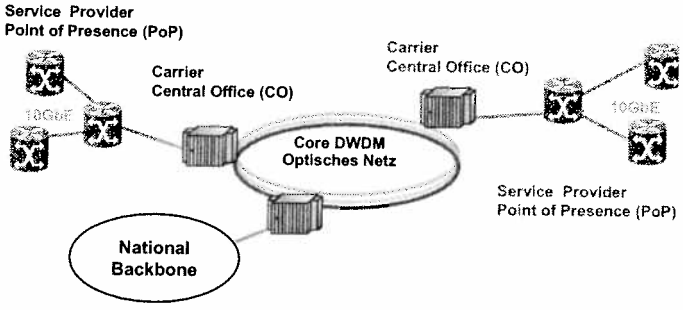


Bild: 10 GbE: WAN-Anwendungen

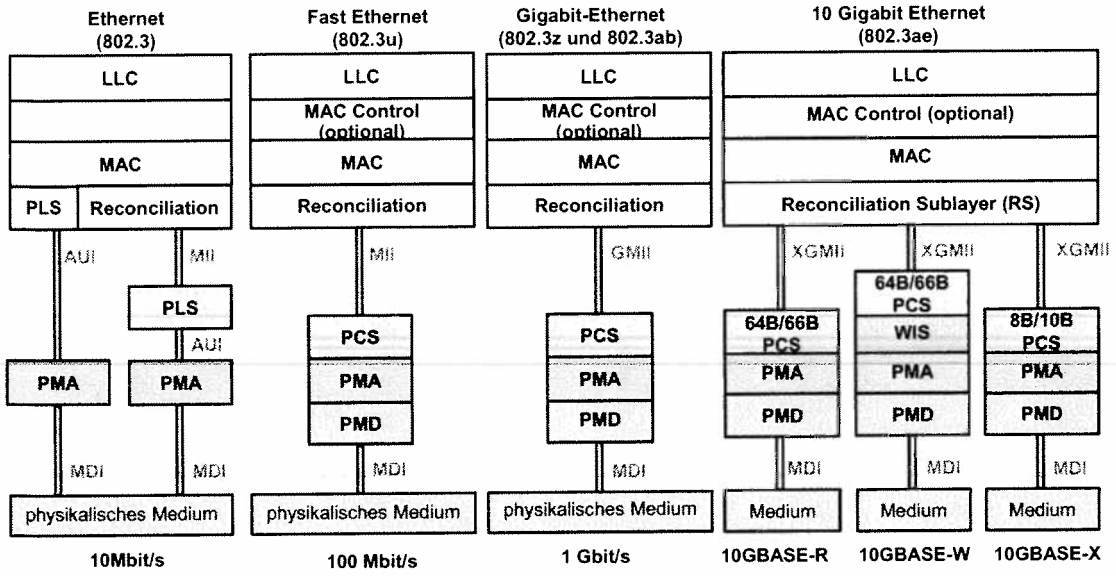


Bild: Ethernet-Schichtenmodelle

Vorteile von Ethernet:

- Bekanntheitsgrad und bewährter Einsatz der Ethernet-Technik. Dadurch sind Treiber für jede gängige Plattform vorhanden.
- Weitere Standardisierung findet statt, besonders im Bereich IP über Ethernet und Hochgeschwindigkeitsnetze..
- Niedrige Preise für Switch-Systeme und Schnittstellenkarten bei Ethernet und Fast-Ethernet: geringe Einstiegskosten.
- Unterstützung von Halb- und Full-Duplex-Verkehr.
- Unterstützung bestehender Infrastrukturen: Investitionssicherheit und geringerer Schulungsaufwand.
- Durch maximale Rahmengröße von 1518 Byte ist geringer Overhead vorhanden.
- Redundante Broadcastpfade werden durch Spanning-Tree-Verfahren ausgeschaltet.
- Switching-Komponenten ermöglichen verbesserte Ausnutzung der Bandbreite sowie weniger Netzkollisionen.
- Verfügbarkeit von Anwendungen: breites Marktangebot zu günstigen Preisen.
- Standardisierte Schnittstellen, um solche Anwendungen effizient zu nutzen: Produktivität und Herstellerunabhängigkeit. Standard-WAN-Schnittstelle ermöglicht beliebigen Übergang ins Weitverkehrsnetz.
- Stand der Normierung und Verfügbarkeit normierter Systemlösungen: Investitionssicherheit und Herstellerunabhängigkeit.
- Funktionen und Prozeduren, um die entsprechende Technologie so einfach wie möglich zu implementieren und zu betreiben: geringe Einführungs- und Betriebskosten.

Nachteile von Ethernet:

- CSMA/CD-Verfahren im Half-Duplex-Modus, wodurch Kollisionen bei hoher Netzauslastung entstehen.
- VLAN-Umsetzung basiert bislang auf proprietären Lösungen der Netzhersteller. Wird allerdings zunehmend durch den Standard IEEE 802.1q abgelöst.
- Keine Skalierbarkeit der Bandbreite auf Shared-Medium-Netze (10, 100 Mbit/s, oder 1 Gbit/s einsetzbar).
- Eingeschränkte Leistungsmerkmale, da Ethernet für reine Datenübertragung konzeptioniert wurde.
- Spanning-Tree-Verfahren aktiviert nach Ausfall einer Verbindung nicht mehr die abgeschalteten Links.
- Quality-of-Service lässt sich nicht ausnutzen, da nur Prioritätsklassen nach IEEE 802.1p definiert wurden.
- Ethernet ist als WAN-Technologie ungeeignet, wodurch Umsetzungen zu anderen Netzen erfolgen müssen. Gigabit-Ethernet will dieses durch Ausnutzung der Fibre-Channel Implementierung ändern. Bisher sind allerdings nur proprietäre Ansätze vorhanden.
- Echtzeitapplikationen lassen sich nur bedingt in Ethernet Umgebung einsetzen.
- Entfernungsbeschränkung.

3.2a Internet-Referenzmodell: Internetschicht - Protokolle

Version: Dez. 2003

Bemerkung vorab: Abgesehen von den beiden IP-Header-Formaten dienen die genauen Formate der verschiedenen Protokolle zur ergänzenden Information. Der Zweck der verschiedenen Formatfelder soll jedoch erklärt werden können.

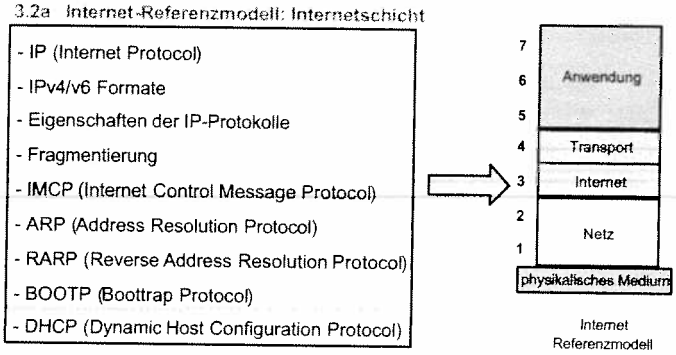


Bild: Übersicht

Aufgaben und Eigenschaften von IP

IP (Internet Protocol) ist ein Protokoll der Vermittlungsschicht, das Datagramme (datagrams) vom absendenden Endsystem (Quelle, source) zum empfangenden Endsystem (Ziel, destination) überträgt. IP leistet eine verbindungslose und unzuverlässige Übertragung. Verbindungslos bedeutet, dass die einzelnen Datagramme unabhängig von anderen Datagrammen behandelt werden, unzuverlässig drückt aus, dass Datagramme verloren gehen können, sowie in falscher Reihenfolge oder mehrfach beim Empfänger eintreffen können.

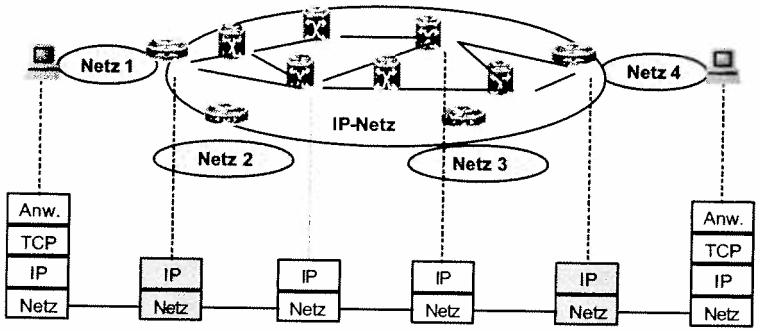


Bild: IP-Vernetzung

IP leistet einen Best-Effort Dienst, d. h., Datagramme werden so gut und rasch wie möglich übertragen, es gibt dafür aber keinerlei Garantien. IP verwendet IP-Adressen, die ein Endsystem global eindeutig kennzeichnen. IP-Adressen besitzen eine innere Struktur, die Netz-ID und Host-ID unterscheidet. Für das Routing ist jedoch nur die Netz-ID von Bedeutung.

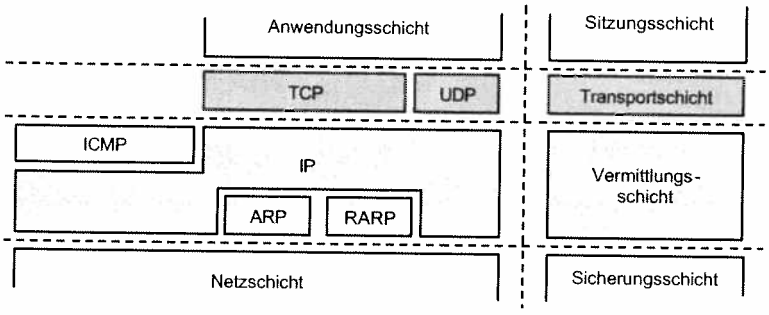
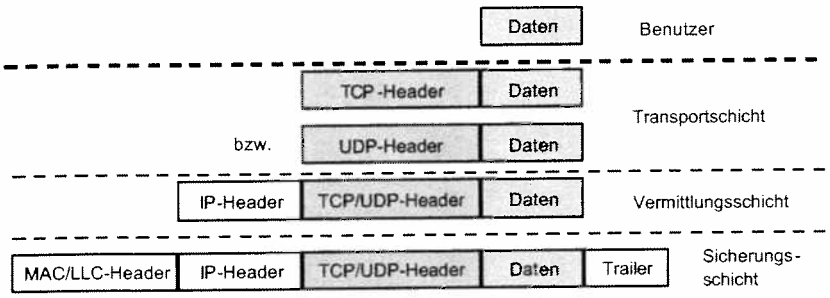


Bild: Basis TCP/IP-Protokollfamilie

TCP: Transmission Control Protocol
UDP: User Data Protocol
IP: Internet Protocol
ICMP: Internet Control Message Protocol
ARP: Address Resolution Protocol
RARP: Reverse ARP

Die Bezeichnung TCP/IP wird häufig als Synonym für die gesamte Protokollfamilie verwendet.

Obwohl ICMP den IP-Dienst nutzt, wird es dennoch der Vermittlungsschicht zugeordnet.



Die Verschachtelung von PDUs (Protocol Data Units) geschieht auf der gleichen Weise wie bei OSI-Protokollen.

Bild: PDU Verschachtelung

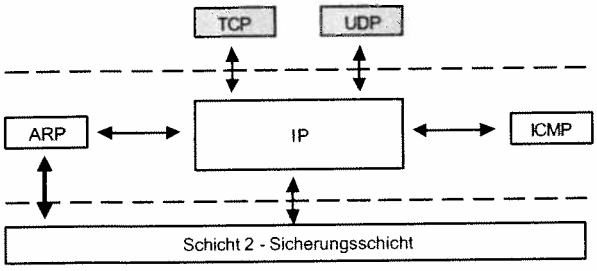


Bild: Zusammenspiel der Protokollinstanzen

Senden:
- Die TCP- bzw. UDP-Instanz übergibt Daten mit der IP-Adresse des Empfängers zur Übertragung an die IP-Instanz.
- IP-Instanz beauftragt ARP-Instanz mit Ermittlung der entsprechenden Schicht-2-Adresse.
- IP-Instanz übergibt Pakete (PDUs) mit der ermittelten Schicht-2 Adresse an die Instanz der Sicherungsschicht.

Empfangen: IP-Instanz reicht empfangene Daten an die TCP- bzw. UDP-Instanzen weiter.

Kontrolle: Im Falle von Problemen während der Übermittlung können diese den Partnerinstanzen über ICMP mitgeteilt werden (wobei ICMP zur Übertragung der Meldungen IP benutzt).

Das Internet Protocol (IP) ist verbindungslos, d.h. jedes Datenpaket ist selbständig und enthält alle notwendigen Informationen, es von einem Endsystem zu einem anderen Endsystem zuzustellen. Die aktuelle Version ist Version 4, Version 6 ist ebenfalls spezifiziert, hat aber noch keine weite Verbreitung gefunden.

Die beiden Hauptfunktionen des Internet Protokolls sind
- Adressierung und Routing,
- Fragmentierung.

TCP (Transmission Control Protocol)
- Stellt verbindungsorientierten, zuverlässigen und bytestromorientierten Transportdienst bereit

UDP (User Datagram Protocol)
- Stellt verbindungslosen, unzuverlässigen und nachrichtenorientiertenTransportdienst bereit

IP (Internet Protocol)
- Sorgt für Wegewahl und unzuverlässige Übertragung einzelner Dateneinheiten

ICMP (Internet Control Message Protocol)
- Unterstützt Austausch von Kontrollinformationen innerhalb der Vermittlungsschicht

ARP (Address Resolution Protocol)
- Zuordnung von IP-Adressen zu den entsprechenden Adressen der Sicherungsschicht

RARP (Reverse Address Resolution Protocol)
-Stellt die Umkehrfunktion von ARP zur Verfügung

Bild: Funktionen einiger Protokolle

Die IP-Schicht erhält ein Datenpaket von der unterliegenden Schicht und leitet es dann nach oben an die entsprechende höhere Protokoll-Schicht weiter. Sind nun mehrere Abnehmer, d.h. höhere Protokoll-Schichten vorhanden, dann muss eine Art Adressierung vorhanden sein. Die Information dazu steckt im Protokollnummer-Feld des IP-Protokoll-Header. Mit seiner Hilfe wird das entsprechende Transport-Protokoll adressiert.

Zusatzprotokolle im Zusammenhang mit IP

Direkt im IP residieren drei Zusatzprotokolle, die dem Betrieb des Internet Protokolls selbst dienen. Dies sind:

- Internet Control Message Protocol (ICMP),
- Address Resolution Protocol (ARP),
- Reverse Address Resolution Protocol (RARP).

Internet Control Message Protocol (ICMP)

Dieses Protokoll residiert innerhalb IP und wird in IP-Paketen transportiert, die den Protokoll-Wert 1 haben. In 8 Bytes werden die ICMP-spezifischen Informationen gesendet. Mit den Feldern Type und Code werden die Nachrichten unterschieden. CRC ist die Prüfsumme über die ICMP-Nachricht. Im Datenfeld werden die für die spezifische Nachricht notwendigen Daten ausgetauscht. Wenn eine Fehlermeldung gesendet wird, dann wird immer der IP-Protokollkopf und die ersten 8 Byte des

Datenfeldes von demjenigen Paket zurückgesendet, das den Fehler verursacht hat. Damit kann der Empfänger eine genaue Analyse des Fehlerfalls durchführen. Für IPv6 wurde ein eigenes ICMP spezifiziert, das sehr viel weniger Nachrichten beinhaltet.

Address Resolution Protocol (ARP)
Die IP-Datenpakete beinhalten eine IP-Adresse-, die MAC-Schicht (z. B. Ethernet) arbeitet aber mit einer eigenen Adresse, der MAC-Adresse. Für die Abbildung aufeinander wurde eine eigene Prozedur geschaffen, das Address Resolution Protocol (ARP).

Reverse Address Resolution Protocol (RARP)
Es gibt auch den umgekehrten Vorgang: Die Suche nach einer IP-Adresse bei gegebener MAC-Adresse. Das Protokoll dazu ist das Reverse Address Resolution Protocol (RARP). Da die MAC-Adresse fest auf der Netzkarte einprogrammiert ist, die IP-Adresse normalerweise konfiguriert wird, bietet RARP einem Host, der seine IP-Adresse nicht speichern kann (z. B. Diskless-Workstation), die Möglichkeit, diese Adresse aus einem Server abzufragen. Allerdings wurde für diesen Anwendungsfall ein mächtigeres Protokoll entwickelt: das Dynamic Host Configuration Protocol (DHCP).

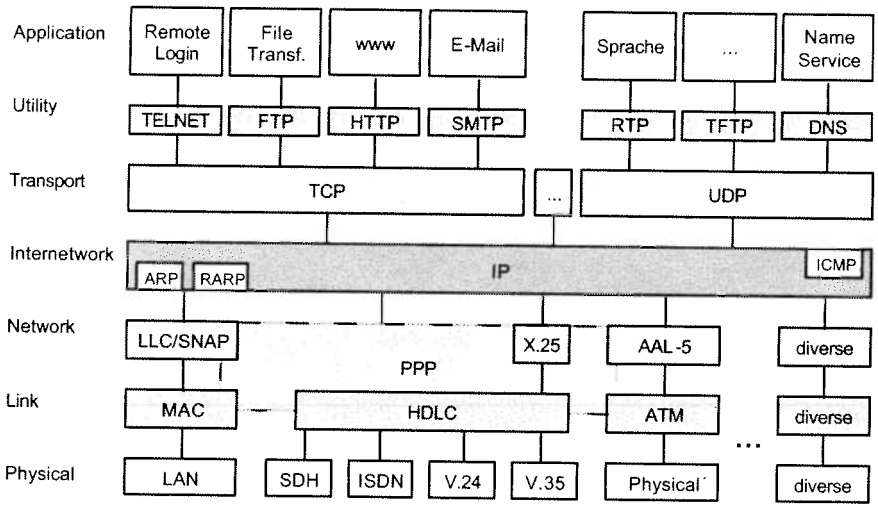


Bild: Protokolle im Internet

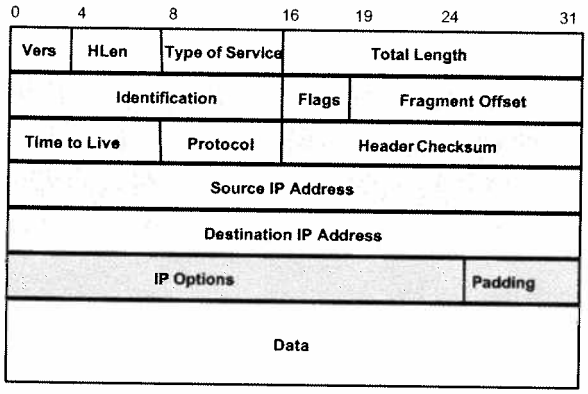


Bild: IPv4 Header-Format

IP-Header (IPv4)

- **Version:** Wert 4.
- **Hlen (Header Length):** Länge des Headers in 32-Bit-Einheiten. Die Mindestlänge beträgt 20 Byte, sie wird bei der Verwendung von Optionen um jeweils 4 Byte verg)rößert.
- **TOS (Type of Service):** 3 Bit-langes Precedence Feld; einzelne Bits D (low delay), T (high throughput), R (high reliability); zwei nicht benutzte Bits. Wegen der Best-Effort-Eigenschaft von IP wurde dieses Feld praktisch nicht benutzt. Es ist jedoch im Zusammenhang mit Quality-of-Service neu definiert worden und damit extrem wichtig geworden.
- **Gesamtlänge:** Datagrammlänge (Header und Daten) in Byte.
- **Identification:** alle Fragmente eines Datagramms enthalten hier denselben Wert, der ihre Zusammengehörigkeit dokumentiert. Die **Fragmentierung** (fragmentation) eines Datagramms kann notwendig werden, wenn ein zwischen Quelle und Ziel liegendes Netz dieses nicht übertragen kann, weil seine Länge größer ist als die für das Netz gültige MTU (Maximum Transmission Unit). Die MTU wird von der Hardware des jeweiligen Netzes bestimmt. Bei der Fragmentierung wird ein Datagramm in Fragmente zerlegt, jedes Fragment erhält einen IP-Header. Die Header zusammengehöriger Fragmente unterscheiden sich nur in den Feldern, die mit der Fragmentierung im Zusammenhang stehen. Der Empfänger macht die Fragmentierung im Prozess der **Reassemblierung** (reassembly) wieder rückgängig.

- **Fragmentierungsflags:** besteht aus drei Bits: ein nicht definiertes Bit, , D-Bit (Do not fragment) weist eine Zwischenstation an, nicht zu fragmentieren, M-Bit (More fragments) kündigt weitere Fragmente an.
- **Fragment Offset:** gibt die laufende Nummer des ersten Byte eines Fragments relativ zum ersten Byte des gesamten Datagramms an. Enthält den Wert null, wenn keine Fragmentierung verwendet wird.
- **TTL (Time-To-Live):** Zähler, der beim Senden des Datagramms auf einen Anfangswert gesetzt und von jedem Zwischensystem (bei jedem Hop) dekrementiert wird. Beim Erreichen des Wertes null wird das Datagramm vernichtet. Dadurch wird eine Überlastung des Netzes durch nicht zustellbare Datagramme vermieden.
- **Protocol:** Nummer des Protokolls, das oberhalb der Vermittlungsschicht verwendet wird (ICMP = 1, IGMP = 2, TCP = 6, UDP =17, SIP = 41, RSVP = 46, OSPF = 89).
- **Header-Prüfsumme:** schützt verfälschte Datagramme gegen Zustellung an die falsche Adresse. Zur Berechnung werden 16-Bit-Werte in Einerkomplement-Arithmetik addiert. Die Prüfsumme ergibt sich als Einerkomplement der berechneten Summe.
- **Quellen- und Zieladresse:** IP-Adressen der Länge 4 Byte.
- **Optionen:** Die Optionen liefern zusätzliche Möglichkeiten zur Steuerung und Überwachung der IP-Übermittlung. Die maximale Länge der Optionen beträgt 44 Byte.
- **Padding:** Nicht benutzte Bits werden mit Füllbits (padding bits) aufgefüllt.
- **Data:** Variables Daten Feld. Daten + header < 64 Kbyte (= 65 535 Byte).

Identifikation eines Paketes
Falls mehrere Versionen eines Protokolls im Einsatz sind, ist eine eindeutige Kennzeichnung nötig. Beim Internet Protokoll ist diese in den ersten 4 Bit des Paketes untergebracht und erlaubt es so, unterschiedliche im Einsatz befindliche Protokoll-Versionen zu unterscheiden. Bei Paketen variabler Länge muss eine Information über die Größe des Paketes vorhanden sein. Naheliegend ist die Angabe einer Gesamtlänge. Ist zusätzlich der Paketkopf variabel, muss auch dessen Länge angegeben werden, damit die Nutzinformation korrekt an die höhere Protokollschicht übergeben werden kann.

Schutz des Paketes
Prüfsummen lassen einen Rückschluss über die Qualität der Übertragung zu. Je nach Aufwand kann es sich um reines Erkennen von Bitfehlern bis zur Korrektur mehrfacher Fehler reichen. Hier ist eine Abwägung zwischen Aufwand und Nutzen notwendig. Im vorliegenden Fall wird eine einfache Prüfsumme nur für den Protokollkopf vorgesehen. Ist auch eine Schutz des Informationsinhaltes des Paketes notwendig, so ist dies auf einer höheren Protokoll-Schicht anzusiedeln.

IP Routing
Wichtig sind die beiden Adressen für Quelle und Senke-, es sind bei IPv4 32-Bit-Adressen. In dynamischen Routing Protokollen kann es - trotz aller Vorkehrungen - zu Schleifen kommen. Das kann dann dazu führen, dass ein Datenpaket ständig zwischen einigen Routern herumgereicht wird. Um zu verhindern, dass dieser Vorgang unendlich lange anhält, wird mit jedem Durchlauf durch einen Router der Inhalt eines speziellen Feldes im Paketkopf (Time-to-Live), das einen Zähler darstellt, vermindert. Eigentlich sollte eine reale Zeit eingetragen werden. In der Realität wird das Feld aber als hop count verwendet, also bei jedem Router-Durchlauf um eins vermindert. Wenn der Zählerstand null erreicht ist, wird das Paket verworfen.

Quality-of-Service (Qos)
Im klassischen Internet ist jeder Verkehr gleich viel Wert, es gibt keine Bevorzugung. Allerdings hatten schon die Erfinder des Internet Protokolls ein Feld vorgesehen, mit dem Qualitätsstufen unterschieden werden können. Dieses Type of Service genannte Feld hat eine Struktur, die es erlaubt, Prioritätsstufen zu unterscheiden und gewisse Kriterien an den Transport des Paketes zu legen.

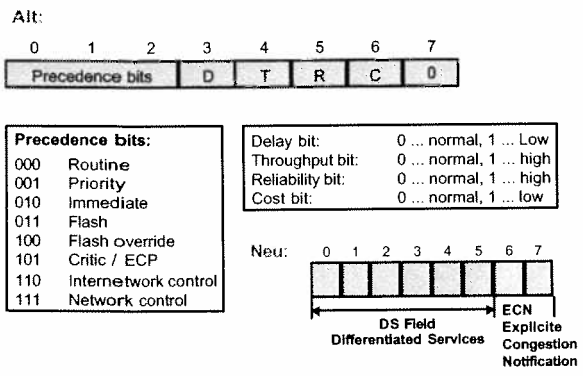
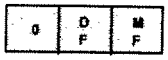


Bild: Type of Service (TOS)

Die Bezeichnungen der acht Prioritätsstufen, mit Precedence bezeichnet, wurden direkt der Originaldokumentation entnommen. Allgemein gilt, je höher der Wert, desto wichtiger ist das Paket. Es wurden Regeln entwickelt, welche Werte bei den Standarddiensten wie telnet und ftp einzusetzen sind. Die Kriterien für den Transport tragen die Bezeichnungen Zuverlässigkeit (Reliabilty), Durchsatz (Throughput), Verzögerung (Delay) und Kosten. Sie sind abstrakt und relativ zu sehen. Es wurden nie Regeln aufgestellt, wie diese Werte in einem Netz zu behandeln sind.
Im Zusammenhang mit neuen Richtungen für QoS wurde das Feld neu eingeteilt, um Mechanismen wie IntServ (Integrated Services), DiffServ (Differential Services) und MPLS (Multiprotocol Label Switching zu unterstützen.

Fragmentierung

IP-Pakete dürfen eine maximale Größe von 65.535 Bytes haben. Die darunter liegenden Schicht-2-Protokolle haben aber in der Regel kleinere Paketgrößen, die von 576 Bytes bei X.25, über 1500 Bytes bei Ethernet bis zu 4500 Bytes bei FDDI reichen. Dieser Wert wird Maximum Transmission Unit (MTU) genannt. Die minimale MTU eines Paketes beträgt bei IPv4 576 Bytes, dabei bedeutet dies nicht, dass es keine kleineren Pakete geben dürfte, sondern dass jede Implementierung Pakete dieser Größe ohne Fragmentierung verarbeiten können muss.



Flags (3 bit):
DF (do not fragment) flag
- 0 Fragmentation is allowed by the router
- 1 Fragmentation is forbidden

MF (more fragments) flag
- 0 last fragment or single fragment
- 1 more fragments will follow

Bild: Flags im IP Datagram Header

Ist jetzt ein IP-Paket zu übertragen, das größer ist als die Schicht 2 transportieren kann, dann fragmentiert IP das Paket, teilt es also in eine Reihe kleinerer Pakete auf. Dieser Vorgang kann im Ursprung oder bei IPv4 auch in einem beliebigen Zwischenknoten (Router) stattfinden. Ein einmal fragmentiertes Paket wird erst am Ziel wieder zusammengesetzt. Wichtig ist, dass das IP-Fragment wieder wie ein normales IP-Paket aussieht. Daher ist auch das Längenfeld entsprechend anzupassen und die Prüfsumme neu zu berechnen. Wenn ein Fragment verloren geht, dann ist das ganze IP-Paket unbrauchbar, es findet keine Sicherung statt.

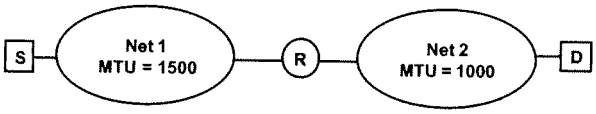


Bild: Maximum Transmission Unit (MTU)

- Jedes Subnetz hat eine maximale IP-Paketlänge: **MTU (Maximum Transmission Unit).**
  - Ethernet: 1518 Byte,
  - FDDI: 4500 Byte,
  - Token Ring: 2 to 4 Kbyte.
- Übertragungseinheit = IP Paket (data + header)

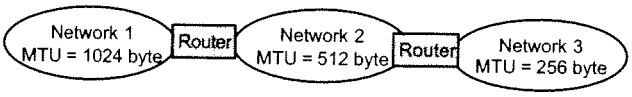


Bild: Fragmentierung

- Datagramme länger als MTU (Maximum Transmission Unit) werden fragmentiert.
- Der Originalheader wird als Kopie in jedem Fragment mitgeführt.
- Bei jedem Fragment wird der Fragmentteil modifiziert (fragment flag, fragment offset, length).
- Auch einige Optionsfelder werden kopiert.

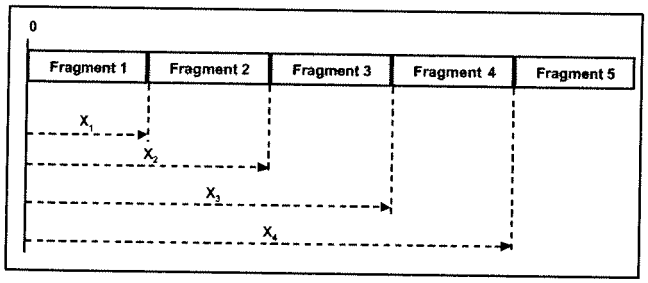


Bild: Payload-Fragmentierung und Fragment Offset

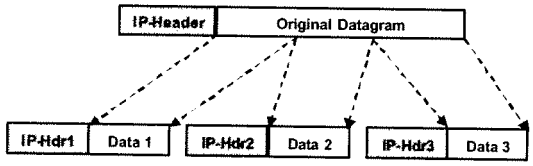


Bild: IP Fragment Offset

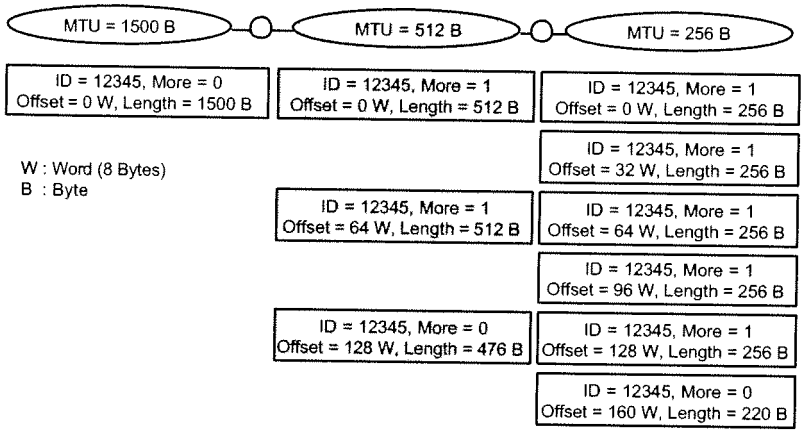


Bild: Beispiel einer Fragmentierung

- Die Reassemblierung der Fragmente findet nur am Ziel statt.
- Teilweise reassemblierte Datagramme werden nach einem Timeout verworfen.
- Fragmente können auf dem Weg zum Ziel weiter fragmentiert werden.
- Die Subfragmente haben das gleiches Format wie die Fragmente.
- Es ist nicht möglich festzustellen, wie oft fragmentiert wurde.
- Die minimale MTU auf dem Weg zum Ziel ist der Pfad MTU.

Im Internet geschieht die Adressierung der Anwendungsprozesse über die Kette:

- IP-Adresse,
- Port-Nummer (Well-known Ports oder anwenderspezifizierte Ports).

Dabei kann ein Port über verschiedene Protokolle erreicht werden. Ein Protokoll ist durch eine Protokollnummer spezifiziert. Die Protokollnummer befindet sich bei IPv4 direkt im Header. Bei IPv6 ist das Protokoll im Header-Extention vorhanden. Die Portnummer ist im TCP oder UDP-Header zu finden.

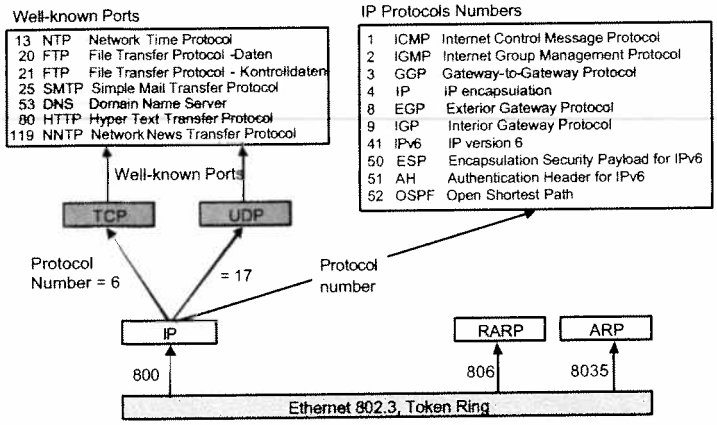


Bild: Adressierung von Internetanwendungen

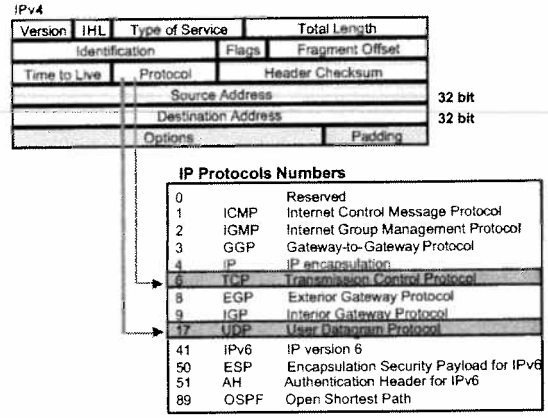


Bild: IPv4 Header und Protokollnummern

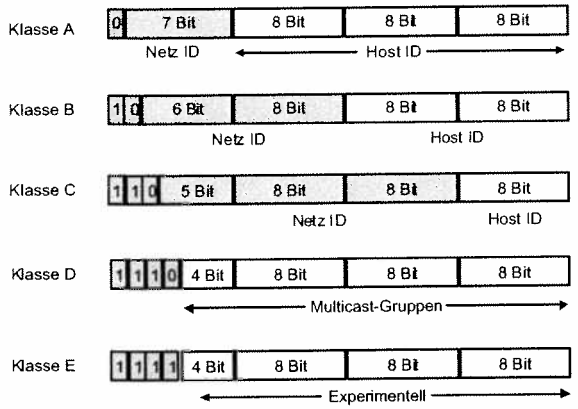


Bild: IPv4-Adressklassen

Internet-Adressen

Jedes Endsystem (Rechner, Router) im Netz wird beim Einsatz der Protokollfamilie TCP/IP durch eine logische IP-Adresse identifiziert. Für alle Endsysteme und Netzkomponenten, die unter Verwendung von TCP/IP kommunizieren, ist eine eindeutige IP-Adresse erforderlich. Jede IP-Adresse (sog. Unicast-Adresse) hat im allgemeinen folgende Struktur: Netz-ID , Host-ID (ID = Identifkation). Die Netz-ID (auch als Netz-ID bezeichnet) identifiziert sämtliche Systeme, die sich im gleichen Netz befinden. Alle Systeme im gleichen Netz müssen dieselbe Netz-ID tragen. Die Host-ID identifiziert ein beliebiges Endsystem (Arbeitsstation, Server, Router, ...) im Netz. Die Identifikation Host-ID muss für jedes einzelne Endsystern in einem Netz (d.h. für eine Netz-ID) eindeutig sein. Eine IP-Adresse bestimmt weltweit eindeutig einen Rechner. Es werden fünf Klassen von IP-Adressen definiert, um den Aufbau der Netze unterschiedlicher Größe zu ermöglichen. Die Adresse einer Klasse legt fest, welche Bits für die Netz-ID und welche für die Host-ID verwendet werden. Sie bestimmt ebenfalls die mögliche Anzahl der Netze und Endsysteme (Hosts

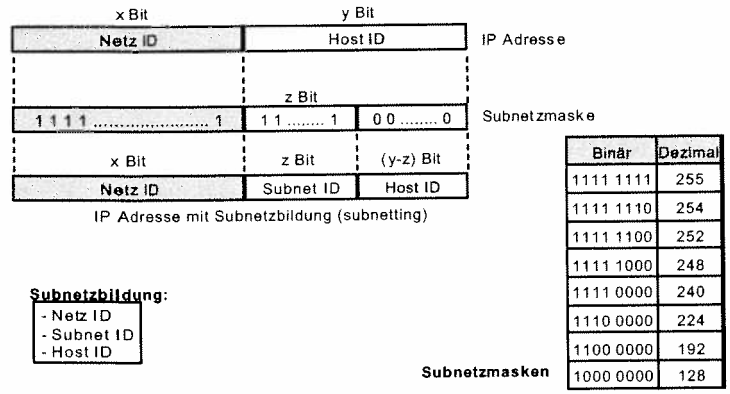
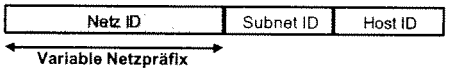


Bild: Subnetzbildung (Subnetting)

Bildung von Subnetzen

Ein Subnetz stellt eine geschlossene Gruppe der Endsysteme (Hosts) dar, und diese Gruppe wird mit einer Subnetz-ID identifiziert. Wird ein physikalisches Netz auf mehrere Teilnetze aufgeteilt, so bezeichnet man diese Teilnetze als Subnetze. Das ganze physikalische Netz kann auch als ein Sonder-Subnetz gesehen werden. Die Subnetze entstehen, wenn autonome Netze in mehrere physikalische oder logische Netze aufgeteilt werden. Zu einem Subnetz können auch mehrere physikalische Netze zusammengefasst werden. Dieser Gruppe von physikalischen Netzen muss eine gemeinsame Subnetz-ID zugewiesen werden.



Ersetzen der festen Netzklassen durch Netz-Präfixe variabler Länge (13 bis 27 Bit)

Bild: Variable Netzmaske

- Security
- Loose source routing
- Strict source routing
- Record route
- Stream identifier
- Timestamp

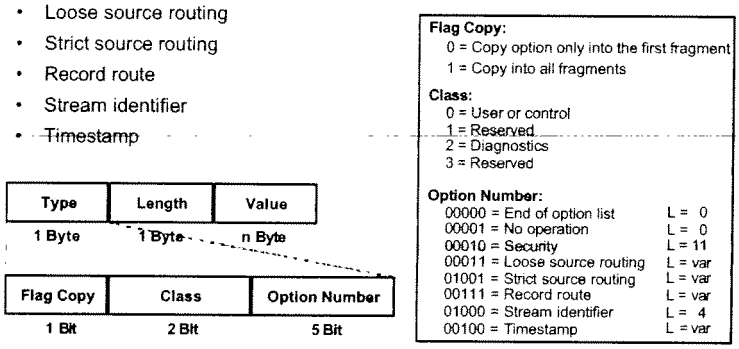


Bild: IP Optionen

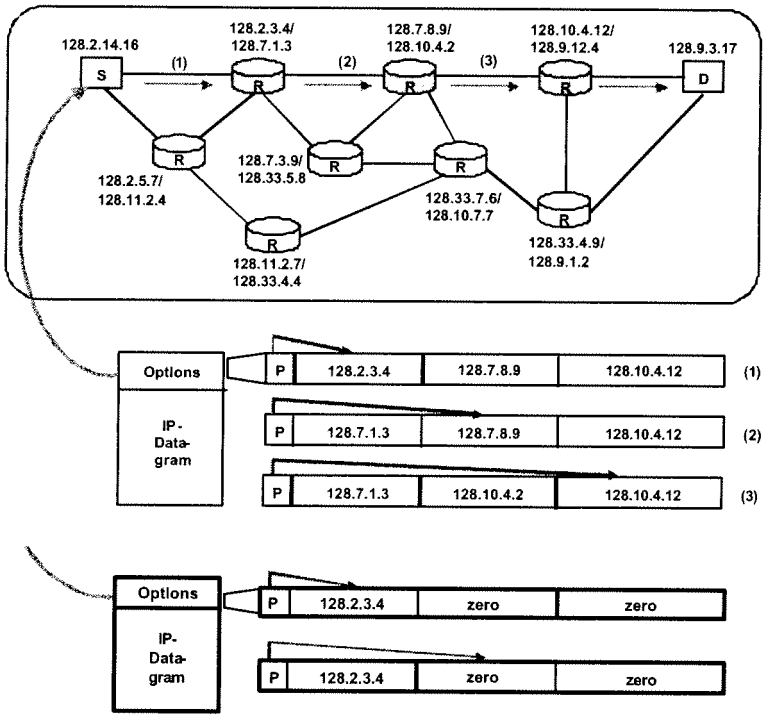


Bild: IP Source Routing

Bild: IP Record Routing

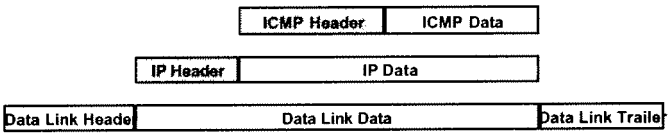
ICMP (Internet Control Message Protocol)

In jedem Netz treten Fehler auf, die an Verursacher oder davon Betroffene gemeldet werden müssen. Diese Aufgabe wird in IP-Netzen vom Protokoll ICMP (Internet Control Message Protocol) übernommen. Hierfür stellt das ICMP eine Vielzahl von sog. ICMP-Nachrichten zur Verfügung. Das ICMP wurde bereits im Jahr 1981 im RFC 792 spezifiziert. Der Funktionsumfang von ICMP wurde später im RFC 1256 erweitert. An dieser Stelle ist hervorzuheben, dass es sich hier um das ICMP für IPv4. Das Protokoll ICMP für IPv6 existiert ebenfalls.

Zu den wichtigsten Aufgaben des Protokolls ICMP gehören:

- Unterstützung der Diagnose.
- Das Hilfsprogramm ping: Üblicherweise wird dieses Hilfsprogramm in Netzen zum Feststellen der Erreichbarkeit des Kommunikationspartners verwendet. Dazu ICMP sendet eine Echo-Anforderung an eine IP-Adresse und wartet auf die Echo-Antworten. Das Programm ping meldet die Anzahl der empfangenen Antworten und die Zeitspanne zwischen Senden der Anfrage und Eingang der Antwort.
- Hilfsprogramm trace (bzw. traceroute) als weiteres Analysewerkzeug wird zum Verfolgen von Routen eingesetzt. Es sendet eine Echo-Anforderung an eine IP-Adresse und analysiert die eingehenden Fehlermeldungen.
- Unterstützung der Aufzeichnung von Zeitmarken (Timestamps) sowie Ausgabe von Fehlermeldungen bei abgelaufenen Timestamps von IP-Paketen.
- Verwaltung von Routing-Tabellen.
- Berichtigung der Flusskontrolle, um eine Überlastung eines Routers bzw. eines Zielrechners zu vermeiden (Source Quench).
- Mitwirken bei der Auffindung der maximal zulässigen Größe von IP-Paketen, d.h. von MTU (Maximum Transfer Unit).

- RFC 792
- ICMP reagiert mit Fehler- und Kontroll-Meldungen bei fehlerhaftem IP-Betrieb.
- ICMP benutzt IP-Datagramme, um Meldungen zu versenden.
- ICMP meldet keine Fehler bei den ICMP-Meldungen selbst.
- ICMP reagiert nicht bei Fehlern der Prüfsumme eines IP-Datagramms.
- ICMP meldet nur Fehler beim ersten fragmentierten Paket.



ICMP (Internet Control Message Protocol) wird immer zusätzlich zu IP benötigt. Es tauscht Nachrichten für die Steuerung der Datenübertragung sowie Fehlermeldungen zwischen Routem und Hosts aus. ICMP-Nachrichten werden zur Übertragung in einem IP-Datagramm gekapselt. Das Protokoll ICMP wird normalerweise als Teil der Schicht 3 betrachtet, aber ausnahmsweise werden die Daten dieses Protokolls in IP-Paketen transportiert. Dem Protokoll ICMP wurde die Protokollnummer 1 im IP-Header zugeordnet.

Bild: Internet Control Message Protocol (ICMP)

ICMP-Nachrichten

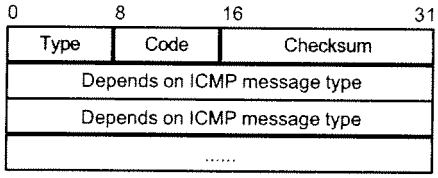
Da ICMP unterschiedliche Informationen zu transportieren hat, enthalten die ICMP-Nachrichten einen Header, der in allen Nachrichten immer gleich ist. Die Bedeutung von ICMP-Nachrichten, die direkt nach dem Header folgen, ist von einzelnen Fehlern bzw. Diagnosesituationen abhängig.

Die einzelnen Angaben im ICMP-Header lauten:

- **Type:** Unterscheidung von einzelnen ICMP-Nachrichten.
- **Code:** Eine weitere Unterteilung der Nachricht innerhalb eines Typs. Beispielsweise in der Nachricht "Destination unreachable" wird dem Absender eines IP-Pakets mitgeteilt, warum es nicht übermittelt werden konnte; z.B.
  - 0 = Netz nicht erreichbar,
  - 1 = Rechner nicht erreichbar,
  - 2 = Protokoll nicht erreichbar,
  - 3 = Port nicht erreichbar,
  - 4 = Fragmentierung erforderlich und DF-Bit gesetzt
- **Checksum (Prüfsumme):** eine Prüfsumme, die nur die ICMP-Daten auf Fehler überprüft.

Falls eine Fehlermeldung zu einem Rechner in einer ICMP-Nachricht ankommt, so stellt sich die Frage, auf welches IP-Paket und welches Protokoll sich die Fehlermeldung bezieht. Abhängig vom Typ (und manchmal auch Code) werden in den ICMP-Nachrichten noch weitere Informationen als ICMP-Daten (Fehler-, Diagnose-Angaben etc.) direkt nach dem Header übermittelt. Die Bedeutung von ICMP-Daten ist von einzelnen Fehler- bzw. Diagnose-Situationen abhängig. Die ICMP-Fehlermeldungen beinhalten neben der Fehlermeldung auch immer den IP-Header und die ersten 64 Bits des diese fehlerhafte Situation verursachenden IP-Pakets.

Empfängt ein Rechner beispielsweise eine ICMP-Nachricht mit Typ = 3 und Code = 1 (Destination Unreachable Message), so kann er nach der Type- und Code-Angabe genau bestimmen, was die Ursache des Fehlers ist. In diesem Fall wird dem Absender eines IP-Pakets mitgeteilt, dass der Zielrechner nicht erreichbar ist. Der Header und weitere 64 Bits dieses IP-Pakets sind in der Destination-Unreachable-Nachricht als ICMP-Daten enthalten.



Type field: type of the ICMP message
Code field: corresponding error specification

Bild: ICMP Message Format

Der ICMP-Header ist je nach Nachrichtentyp unterschiedlich aufgebaut. Er enthält unteren andern die folgenden Felder:

- **Type:** Gibt den Typ einer ICMP-Meldung an.
- **Code:** Kann den Typ näher beschreiben.
- **Prüfsumme:** Prüfsumme über die gesamte ICMP-Nachricht. Verwendet denselben Algorithmus wie IP.
- **Identifier und Sequenznummer:** kennzeichnet zusammengehörige Anfragen und Antworten (Typ 0 oder 8).
- **Optionale Daten:** Hier stehen Daten, die in einem echo request vom Sender zum Empfänger übertragen wurden. Im echo reply werden die Daten unverändert zurückgeschickt.

Type	Function	Type	Function
0	Echo Reply	15	Information Request
1	-		Information Reply
2	-	17	Adress Mask Request
3	Destination Unreachable	18	Adress Mask Reply
4	Source Quench	19	Reserviert for Security
5	Redirect	20-29	Reserviert for Robustness Experiments
6	Alternate Host Address	30	Trace Route
7	-	31	Datagram Conversion Error
8	Echo	32	Mobile Host Redirect
9	Router Advertisement	33	IPv6 - Where are You
10	Router Solicitation	34	IPv6 - Am Here
11	Time Exceeded	35	Mobile Registration Request
12	Parameter Problem	36	Mobile Registration Reply
13	Timestamp	37-255	-
14	Timestamp Reply		

Bild: Type Field Numbers

Nachrichten des Typs destination unreachable senden den IP-Header und die ersten 64 Bits des nicht zustellbaren Datagramms an dessen Absender zurück.

ICMP-Fehlermeldungen

Der häufigste Einsatz von ICMP liegt in der Meldung verschiedener Arten von fehlerhaften Situationen. Ein Rechner oder ein Router gibt eine ICMP-Fehlermeldung zurück, wenn er feststellt, dass ein Fehler oder eine außergewöhnliche Situation während der Weiterleitung bzw. der Übergabe an ein Transportprotokoll (TCP oder UDP) eines IP-Pakets aufgetreten ist.

- **Destination unreachable:** Destination error or fragmentation Required.
- **Source Quench:** Senderate drosseln.
- **Redirect:** Route ändern
- **Time Exceeded:** Lebenzeit des IP-Paketes überschritten oder bei Framentierung dauerte die Assemblierzeit am Empfänger zu lang.
- **Parameter Problem:** Parameterfehler (Type-of-Service oder Optionen)
- **Router advertisement:** Router-Bekanntmachung.
- **Router Solicitation:** Suche nach einem Router.
- **Trace Route:** Ermittlung von Übermittlungszeiten des gesamten Weges.
- **Echo request/reply:** verwendet für Ping-Vorgang bei der schrittweisen Ermittlung von Übermittlungszeiten.
- **Time stamp request/reply:** Uhrzeit-Ermittlung für Diagnose und Überwachung.
- **Information request/reply:** z.B. verwendet zur Ermittlung der Netz-ID einer IP-Addresse.
- **Address mask request/reply:** verwendet zur Ermittlung welche Subnetz-Maske momentan in einem Netz aktuell ist.

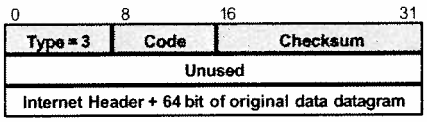
Bild: Verwendung von ICMP Messages

Diese außergewöhnlichen Situationen, die eine ICMP-Fehlermeldung verursachen, sind:

- Destination Unreachable Message (Ziel nicht erreichbar)
- Source Quench Message (Übertragungsrate reduzieren)
- Redirect Message (Umleitung im Netz)
- Time Exceeded Message (Zeit überschritten)
- Parameter Problem Message (Ungültige Parameter)

Weitere ICMP Nachrichten gehören zum

- Routingmanagement,
- Request/Reply.



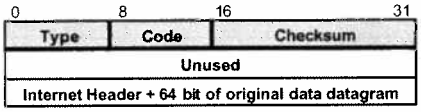
Code	Meaning
0	Network unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed but do not fragment bit has been set
5	Source route failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Communication with destination network administratively prohibited
10	Communication with destination host administratively prohibited
11	Network unreachable for type of service
12	Host unreachable for type of service

Bild: ICMP Messages: Destination unreachable

Destination Unreachable Message
(Ziel nicht erreichbar)

Ein IP-Paket kann nicht an den Zielrechner übergeben werden. In diesem Fall wird die Nachricht Destination Unreachable an den Quellrechner gesendet, um darauf hinzuweisen, dass der Empfänger nicht erreichbar ist. Die Ursachen hierfür sind unterschiedlich. Eventuell existiert der Zielrechner nicht mehr, oder es ist kein passendes Protokoll im Zielrechner geladen.

Source Quench: Type 4
Redirect: Type 5



Source Quench Message

Code	Meaning
0	Datagram could not be processed

Redirect Message

Code	Meaning
0	Redirect datagrams for the network
1	Redirect datagrams for the host
2	Redirect datagrams for the service type and network
3	Redirect datagrams for the service type and host

Bild: ICMP Messages; Source Quench and Redirect

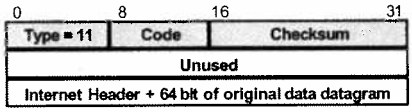
Source Quench Message
(Übertragungsrate reduzieren)

Ist ein Rechner nicht in der Lage, die zu schnell ankommenden IP-Pakete rechtzeitig zu verarbeiten, wird die Nachricht Source Quench an die Quelle gesendet, damit diese die Sendung von IP-Paketen für einen gewissen Zeitraum unterbricht.

Redirect Message
(Umleitung im Netz)

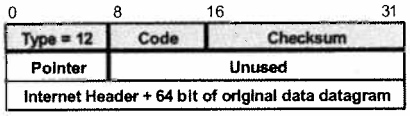
Bemerkt ein Router, dass es für ein IP-Paket eine bessere Route gibt als über diesen Router, so kann er dem Quellrechner eine Empfehlung mit der Nachricht Redirect geben, weitere IP-Pakete zum gleichen Zielrechner über einen anderen Router zu verschicken. Die IP-Adresse dieses Routers wird im Feld ICMP-Data übermittelt.

Time Exceeded Message



Code	Meaning
0	Time-to Live = 0
1	Time for fragmentation exceeded

Parameter Problem Message



Pointer	Meaning
1	Error in Type-of-Service Field
20	Error in Type-code of first Option

Bild: ICMP-Messages: Time Exceeded und Parameter Problem

Time Exceeded Message
(Zeit überschritten)

Befindet sich ein IP-Paket so lange im Netz, dass der Time-To-Live-Parameter im IP-Header abgelaufen ist, so wird die Nachricht Time Exceeded vom Router, in dem das betreffende IP-Paket vernichtet wurde, an den Quellrechner zurückgeschickt.

Parameter Problem Message
(Ungültige Parameter)

Ein oder mehrere Parameter im Header des IP-Pakets enthalten ungültige Angaben bzw. unbekannte Parameter. In diesem Fall wird die Nachricht Parameter Problem verschickt.

ICMP für Routing-Management

Jedem Rechner in einem Subnetz muss die IP-Adresse eines Routers als Grenzübergang zu anderen Subnetzen bekannt sein. Diese Adresse wird üblicherweise bei der IP-Konfiguration eines Rechners als Default Gateway angegeben. Das ICMP stellt zwei Nachrichten zur Verfügung, die es ermöglichen, einen Router zu entdecken.

Diese Nachrichten zur Entdeckung von Routern in einem Subnetz sind:

- **Router Solicitation** (Suche nach einem Router)
- **Router Advertisement** (Router-Bekanntmachung)

Ein Rechner kann während seiner Konfigurationsphase eine Nachricht Router Solicitation an alle Systeme (Rechner, Router) in demselben Subnetz verschicken. Diese Nachricht dienst dazu, einen Router zu suchen und enthält im IP-Header eine IP-Multicast-Adresse 244.0.0.1 bzw. eine Limited Broadcast-Adresse 255.255.255.255. Der Router antwortet mit der Nachricht Router Advertis ement, in der er seine IP-Adresse von diesem physika lischen Port bekannt gibt, auf dem die Nachricht Router Solicitation emp fangen wurde. Einem physikalischen Port im Router können mehrere IP-Adressen zugeordnet werden, so dass in der Nachricht Router Advertisement alle IP-Adressen des entsprechenden Router-Ports enthalten sein können.

ICMP-Anfragen

Zusätzlich zu den ICMP-Meldungen, die in den fehlerhaften Situationen generiert werden, gibt es eine Reihe weiterer ICMP-Nachrichten, die für die Anfrage von Informationen und zur Antwort auf eine ICMP-Anfrage verwendet werden können. Hierzu gehören:

- **Echo Request / Reply Message (Echo-Funktion):** Die häufigsten Anfragemeldungen sind die ICMP-Nachrichten für die Implementierung des Programms ping zum Versenden von Diagnose-Nachrichten. Die Nachrichten Echo Request/Reply werden für die Implementierung einer sog. "Are you there"-Funktion verwendet. Hierbei wird von dem ping-Programm ein Echo-Request zu einem bestimmten Ziel (Rechner bzw. Router) gesendet. Das Ziel muss auf den Echo Re quest mit einem Echo Reply antworten. Die Nachricht Echo Request ist die einzige ICMP-Nachricht, auf die jeder IP-fähige Rechner antworten muss.
- **Timestamp Request/Reply Message (Zeitmarkenanfrage):** Ein Rechner oder ein Router gibt eine Zeitmarkenanfrage ab, um von einem anderen Rechner oder Router eine Zeitmarke zu erhalten, die das aktuelle Datum und die Uhrzeit angibt. Ein Rechner oder Router, der eine Zeitmarkenanfrage in der Nachricht Timestamp Request empfängt, antwortet mit der Nachricht Timestamp Reply. Die Nachrichten Timestamp Request und Reply verwendet werden, um die Laufzeit eines IP-Pakets über das Netz zu messen.
- **Information Request/Reply Message (Informationsanfrage):** Diese Nachrichtentypen sollen es einem Rechner ermöglichen, seine IP-Adresse (z.B. von einem Adress-Server) abzufragen. Da die dynamische Vergabe von IP-Adressen heutzutage mit dem Protokoll DHCP (Dynamic Host Configuration Protocol) gemacht wird, hat diese ICMP-Funktion an Bedeutung verloren.
- **Address Mask Request/Response (Abfrage der Subnetz-Maske):** Diese Nachrichtentypen ermöglichen es einem Rechner, die zu verwendende Subnetz-Maske abzufragen. In einem Subnetz, in dem diese Funktion unterstützt wird, sind ein oder mehrere Rechner als Subnetz-Masken-Server gekennzeichnet. Ein Rechner, der seine Subnetz-Maske zu ermitteln versucht, sendet eine Abfrage in der Nachricht Address Mask Request, auf die ein Subnetz-Masken-Server mit einer Nachricht Address Mask Response antwortet, in der die zu verwendende Subnetz-Maske enthalten ist.

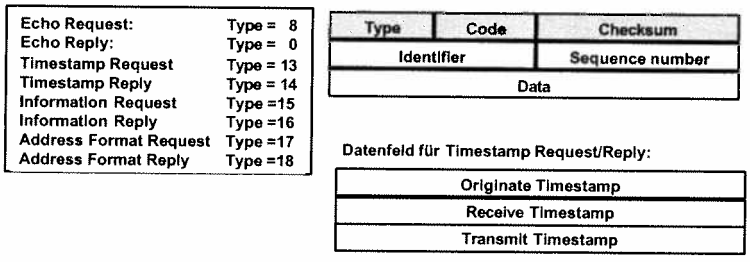


Bild: ICMP-Messages: Request und Reply

Pfad-MTU (PMTU) Ermittlung

Eine wichtige Funktion von ICMP besteht in der Unterstützung der Feststellung der Maximum Transfer Unit (MTU) für ein entfemtes, über Router zu erreichendes IP-Netz. Dieses Verfahren wird als Path MTU (PMTU) Discovery bezeichnet, und ist laut RFC 1191 in Routern zu unterstützen.

Die PMTU wird im Zusammenspiel zwischen IP-Quellsystemen und den in der PMTU Übermittlungsstrecke liegenden Routern entsprechend folgendem Ablauf festgestellt:

1. Die IP-Instanz des Senders generiert zunächst IP-Pakete mit gesetztem Don't Fragment-Bit (DF = 1) und der maximalen MTU des lokalen Netzes. Diese MTU entspricht in der Regel auch der des in diesem Netz liegenden IP-Interfaces des Default Gateway und somit des ersten Hops.
2. Überschreitet ein erzeugtes IP-Paket die MTU eines Transfernetzes, so dass der zugehörige Router es eigentlich fragmentieren müsste, wird es von diesem verworfen und der Sender erhält die ICMP-Nachricht Destination Unreachable mit dem Statuscode "fragmentation needed and DF set". Ferner fügt der Router die maximal mögliche IP-Paketgröße (in Bytes) in die ICMP-Nachricht ein.
3. Der Sender ist somit aufgefordert, seine ursprüngliche MTU auf die nun bekannte Obergrenze zu reduzieren und die Datagramme erneut zu übertragen.
4. Dieses Verfahren kann periodisch wiederholt werden, um z. B. wechselnden Routen zu entsprechen.

Protokolle ARP (Address Resolution Protocol) und RARP (Reverse ARP)

Diese Protokolle sind Hilfsprotokolle für die Adressierung von IP-Paketen. Das Protokoll ARP hat die Aufgabe, für eine Ziel-Adresse die korrespondierende MAC-Adresse zu ermitteln. Das Protokoll RARP ermöglicht, für eine MAC-Adresse die entsprechende IP-Adresse zu bestimmen. RARP wird vorwiegend von Rechnern ohne Festplatte (z. B. Netz-Computer NCs) genutzt, die als die Stationen am LAN dienen und ihre IP-Adresse nicht selbst speichern können. In Routern wird oft eine zusätzliche Lösung für das Protokoll ARP eingesetzt, die als Proxy ARP bezeichnet wird.

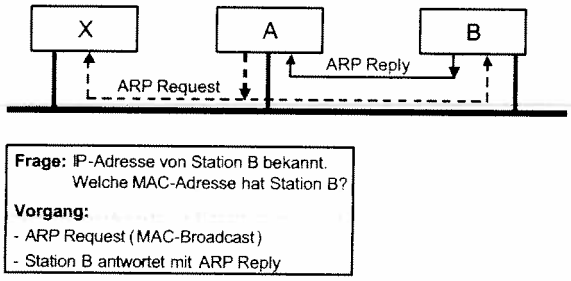


Bild: ARP: Address Resolution Protocol

ARP (Address Resolution Protocol), RFC 826
ARP setzt IP-Adressen auf MAC-Adressen um. ARP beschränkt sich dabei auf ein physikalisches Teilnetz. Die Übersetzung wird dynamisch vorgenommen, indem ein Rechner A mittels Broadcast eine ARP-Anfrage mit der zu übersetzenden IP-Adresse aussendet. Wenn Rechner B darin seine IP-Adresse erkennt, antwortet er mit seiner Hardwareadresse

Die bei ARP-Anfragen erhaltenen Adressübersetzungen werden in einem Cache gespeichert. Beim Senden von Paketen wird die Adressübersetzung zuerst im Cache gesucht. Nur falls sie dort nicht vorhanden ist, wird eine ARP-Anfrage gesendet. Einträge im Cache bleiben nur für eine bestimmte Zeit gültig.

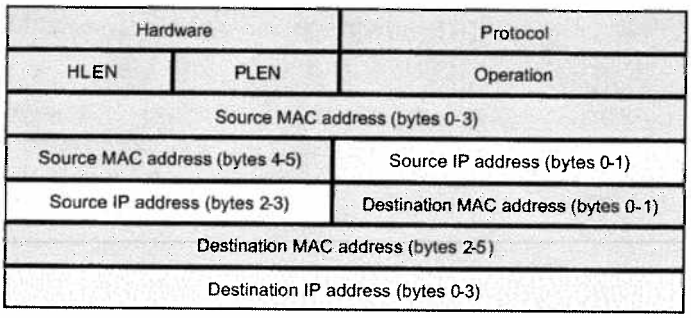


Bild: ARP Request/Reply

ARP-Pakete

- **Hardware:** LAN-Typ (z.B. Ethernet, IEEE 802.x - LANs) in welchem das Paket generiert wurde.
- **Protokoll:** Netzprotokoll von welchem die Operation angefordert wurde. Für das IP-Protokoll gilt den Wert 800.
- **Hardware Address Length (HLEN):** Länge der Hardwareadresse in Bytes (normalerweise MAC-Adresse mit 6 Bytes).
- **Protocol Address Length (PLEN):** Länge der Protokolladresse in Bytes. Bei IPv4-Adressen (4 Bytes).

- **Operation:** = 1: ARP Request, = 2: ARP Reply (RFC 826), = 3: RARP Request, = 4: RARP Reply (RFC 903).
- **Sender MAC Address:** enthält die MAC-Adresse des Absenders.
- **Sender Protocol Address**: enthält die IP-Adresse des Absenders.
- **Destination MAC Address:** enthält die gesuchte MAC-Adresse (in ARP-Reply).
- **Destination Protocol Address:** enthält die IP-Adresse, für die die MAC-Adresse ermittelt wird.

Protokoll ARP
Zwei Adressierungsstufen sind zu unterscheiden. Einerseits müssen die Hardwarekomponenten (Endsysteme, Router) in jedem Netz eindeutig identifiziert werden. Hierfür verwendet man physikalische Netzadressen. In LANs mit einem gemeinsamen Medium werden die Netzadressen als MAC-Adressen bezeichnet. Da diese Adressen unstrukturiert sind und somit keine Hinweise auf die Lokation enthalten, werden sie auch als Nummern von LAN-Adapterkarten gesehen. Andererseits müssen die Daten in Form von IP-Paketen zwischen zwei Kommunikationspuffern in Endsystemen ausgetauscht werden. Diese Kommunikationspuffer sind im logischen LAN-Modell an der Grenze zwischen den Schichten 3 und 4 zuzuordnen. Liegt ein IP-Paket in einem Endsystem am LAN zum Senden vor, so wird dieses Paket in einen MAC-Rahmen eingebettet. Im Header des MAC-Rahmens ist eine entsprechende MAC-Adresse des Zielsystems enthalten. Somit muss eine Tabelle mit den Zuordnungen IP-Adresse zu MAC-Adresse in LAN-Endsystemen vorhanden sein.

Früher wurde dieses Problem in jedem Rechner durch statische Tabellen gelöst, in die man manuell alle Zuordnungen zwischen MAC- und IP-Adressen eintragen musste. In dieser Form war der Verwaltungsaufwand sehr hoch und das ganze System unflexibel. In den heutigen IP-Netzen werden diese Zuordnungen mit dem Protokoll ARP realisiert. Das Protokoll ARP ist ein Hilfsprotokoll zur Ermittlung einer physikalischen Interface-Adresse (MAC-Adresse) für ein höheres Protokoll (z.B. IP). Es ist für die Zuordnung von MAC-Adressen zu Protokoll-Adressen verantwortlich. Das Protokoll ARP legt eine dynamisch organisierte Adressermittlungstabelle mit IP-Adressen und den zugehörigen MAC-Adressen an. Diese Tabelle wird oft als ARP-Cache bezeichnet. Wenn das Protokoll IP die Anforderung erhält, ein Paket an eine IP-Adresse im gleichen Subnetz zu senden, sucht es zuerst im ARP-Cache nach der korrespondierenden MAC-Adresse. Falls kein Eintrag vorhanden ist, wird versucht, mit Hilfe von ARP die gesuchte MAC-Adresse zu ermitteln.

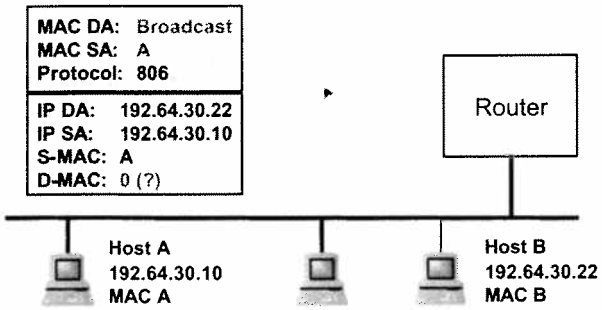


Bild: ARP-Request (MAC-Broadcast) von Station A

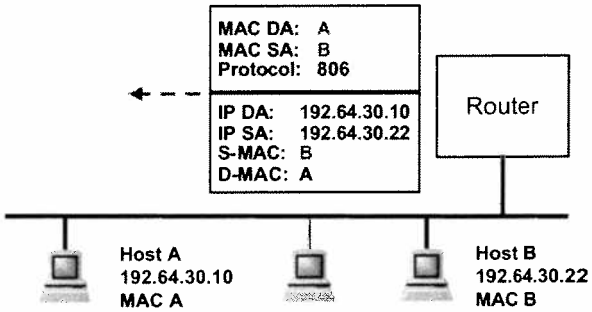


Bild: ARP-Reply (MAC-Unicast) von Station B

Hierfür wird ein ARP-Request als ein MAC-Broadcast verschickt. In diesem Request werden alle Endsysteme im selben Subnetz aufgefordert, die gesuchte Adresszuordnung von IP-Adresse zu MAC-Adresse zukommen zu lassen. Ein Endsystem schickt immer ein ARP-Reply als MAC-Unicast mit der gesuchten Zuordnung zurück. Anschließend wird die Adress-Korrespondenz im Cache abgelegt.

Die Ermittlung einer MAC-Adresse im Endsystem A nach dem Protokoll ARP erfolgt wie folgt. Die Broadcast-Nachricht ARP-Request enthält die IP-Adresse der angeforderten MAC-Adresse und wird in allen Endsystemen im LAN gelesen. Sobald ein Endsystem die eigene IP-Adresse im ARP-Request erkennt (hier Endsystem B), antwortet es mit einem ARP-Reply. Die beim Endsystem A eingehende Antwort wird im ARP-Cache vermerkt und steht damit für spätere Übertragungen zu Verfügung. Falls innerhalb einiger Sekunden keine Antwort eingeht, wird die Anforderung wiederholt.

Damit nicht bei jeder Übertragung erneut Anforderungen ARP-Request gesendet werden müssen, kopiert auch das Endsystem B, d.h. das auf ARP-Request antwortet, die Zuordnung von IP-Adresse und MAC-Adresse des ARP-Request-Absenders (Endsystems A) in seinen eigenen ARP-Cache. Bei einer eventuellen Übertragung in Gegenrichtung (von A zu B) ist es daher nicht mehr nötig, eine ARP-Anforderung in umgekehrter Richtung zu senden, da die MAC-Adresse der IP-Adresse, der gerade geantwortet wurde, bereits bekannt ist.

Den Aufbau von Nachrichten ARP-Request und -Reply (ARP-PDU) zeigt das Bild. Es ist hier hervorzuheben, dass diese Nachrichten direkt in MAC-Frames transportiert werden. Sie werden somit auf dem MAC-Level übermittelt. Die Folge dessen ist, dass der ARP-Request von Routern nicht weitergeleitet werden kann, da Router auf dem IP-Level operieren und somit auf MAC-Broadcast-Nachrichten nicht reagieren. Diese Tatsache hat in der Praxis einen Nachteil. Als Folge dessen ist eine Proxy ARP Lösung notwendig.

In manchen TCP/IP-Implementierungen werden für Einträge im ARP-Cache ein Zeitlimit (time-out) gesetzt. Falls der Eintrag innerhalb dieses Zeitraums, oft 15 Minuten, nicht verwendet wird, wird er gelöscht. Einige Systeme arbeiten wiederum mit einem zeitgesteuerten Aktualisierungsprinzip. Alle 15 Minuten wird dann eine Anforderung ARP-Request gesendet, um sicherzustellen, dass die Cache-Einträge dem aktuellen Systemzustand entsprechen. Da MAC-Adressen normalerweise nur verändert werden, wenn eine Adapterkarte bzw. der ganze Rechner ausgetauscht wird, scheint dieses Prinzip von keiner großen Bedeutung zu sein.

In den Token-Ring-LANs, falls mehrere LANs miteinander vernetzt werden, muss das sogenannte Source Routing in Endsystemen unterstützt werden. Um das Source Routing unterstützen zu können, enthält der ARP-Cache in Endsystemen am Token-Ring eine zusätzliche Spalte mit der Angabe des nächsten Router-Abschnittes Next-RD (RD: Route Designator).

Probleme kann es mit ARP geben, wenn in einem Netz zwei Stationen die gleiche IP-Adresse besitzen. In einem solchen Fall kann keine exakte Zuordnung zwischen IP-Adresse und MAC-Adresse getroffen werden, d.h. die Daten werden nicht korrekt weitergeleitet, oder es wird aufgrund einer nicht identifizierten Verbindung eine Fehlermeldung produziert. In einem gut organisierten Netz ist mit diesem Problem nur ganz selten zu rechnen.

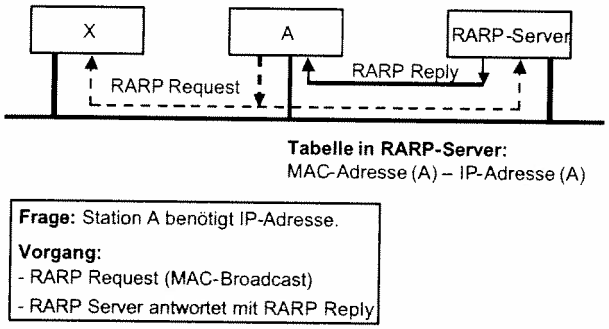


Bild: Reverse Address Resolution Protocol (RARP)

RARP (Reverse Address Resolution Protocol)

Die RARP-Anfrage eines Hosts wird mittels Broadcast an alle anderen Teilnehmer des Teilnetzes gesendet. Diejenigen Stationen, die als RARP-Server aufgesetzt sind, kennen die IPAdresse von Host A und teilen diese in einer RARP-Antwort mit. Falls mehrere RARP-Server aktiv sind, empfängt A mehrere Antworten, von denen jedoch nur die erste ausgewertet wird.

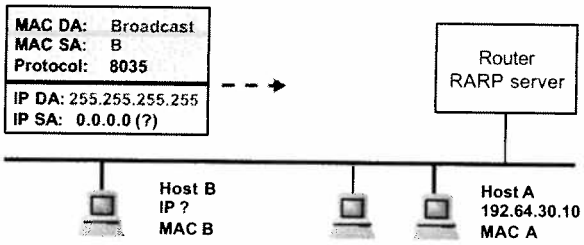


Bild: RARP-Request (MAC Broadcast)

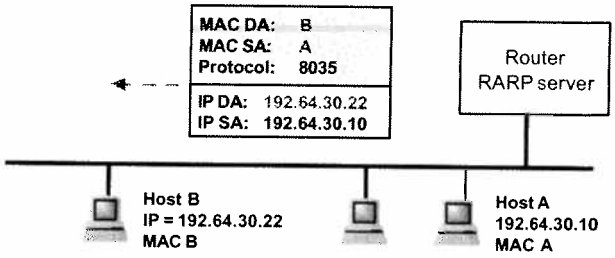


Bild: RARP-Reply vom RARP-Server

Protokoll RARP

Das Protokoll RARP (Reverse Address Resolution Protocol) ist für Stationen gedacht, die ihre IP-Adresse nicht selbst speichern können (z. B. Remo te-Boot-Stationen ohne Festplatte). RARP ist das Gegenstück zu ARP. Deshalb bietet RARP Funktionen, die es ermöglichen, aus einer bekannten MAC-Adresse die zugehörige IP-Adresse zu finden. Bei RARP ist es notwendig, einen speziellen Server festzulegen, in dem eine RARP-Tabelle enthalten ist. Der Server sucht in dieser Tabelle nach der IP-Adresse, die mit der angeforderten MAC-Adresse übereinstimmt und gibt die gesuchte IP-Adresse als RARP-Antwort (Reply) bekannt.

Das RARP-Prinzip setzt voraus, dass mindestens ein Rechner als RARP-Server fungiert und dass dieser Server über eine Tabelle verfügt, in der allen MAC-Adressen eine eindeutige IP-Adresse zugeordnet ist.

Der Aufbau von RARP-Nachrichten ist wie bei ARP. Beim Protokoll RARP werden im Feld Operation die Werte 3 für RARP-Request und 4 für RARP-Reply verwendet. Wenn ein RARP-Request gesendet wird, kennt das aussendende Endsystem nur die eigene MAC-Adresse und kann daher auch nur diese Adresse im MAC-Frame angeben. In der Antwort RARP-Reply vom Server wird die gesuchte IP-Adresse eingetragen. In dieser Antwort kann auch die IP- und MAC-Adresse des RARP-Server angegeben werden. Dies ist allerdings nicht erforderlich.

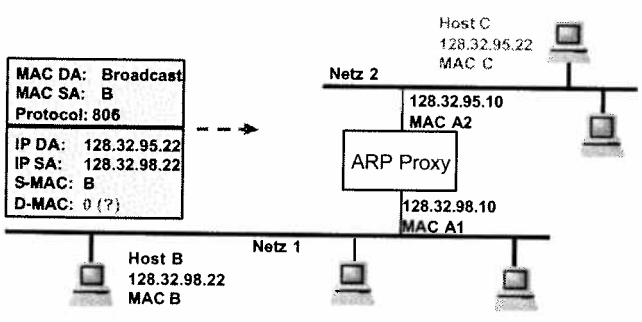


Bild: ARP-Request (MAC Broadcast in Netz 1)

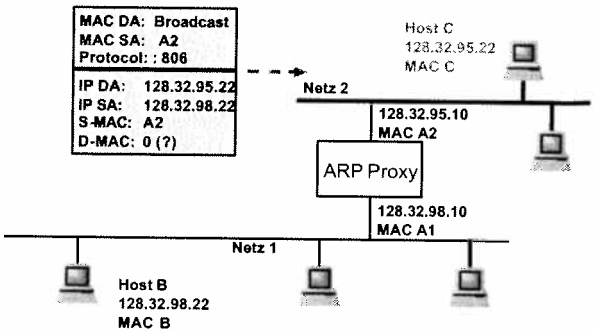


Bild: ARP-Request (MAC Broadcast in Netz 2)

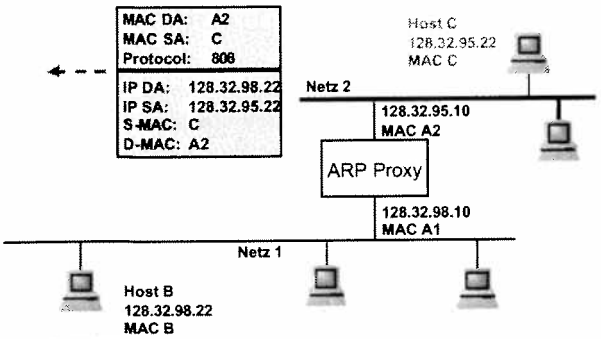


Bild: ARP-Reply von Station C (Netz 2)

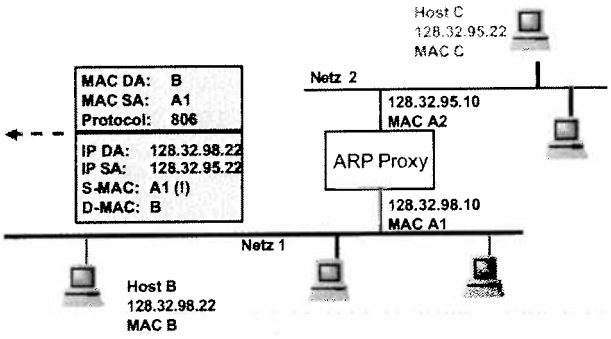


Bild: ARP-Reply von Station C (Netz 1)

Proxy-ARP,

Proxy-ARP (RFC 1027) ist eine Lösung, die es ermöglicht, mehreren physikalischen Netzen denselben Netz-ID bzw. Subnetz-ID zuzuordnen.

Das Ziel ist ein Shared Medium LAN (beispielsweise Ethernet LAN) mit ISDN so zu integrieren, dass diese beiden physikalisch unterschiedlichen Netze logisch als ein Subnetz gesehen werden können. Hier sind externe Rechner über das leitungsvermittelnde ISDN an ein Ethernet LAN angebunden. Aus organisatorischen Gründen müssen diese externen Rechner transparent, also mit IP-Adressen des lokalen Subnetzes (d.h. Ethernet LANs) eingebunden werden. Für die Übermittlung der IP-Pakete zwischen den externen Rechnern und dem Router wird das Protokoll PPP (Point-to-Point Protocol) verwendet.

Im LAN werden die Endsysteme mit den MAC-Adressen als LAN-Hardware-Adressen identifiziert. Außerdem ist das LAN ein Broadcast-Netz, während ISDN ein leitungsvermittelndes Netz darstellt, in dem das Broadcast nicht unterstützt werden kann. Das Protokoll ARP setzt ein Broadcast-Netz voraus. Somit lässt sich dieses Protokoll im ISDN nicht realisieren. Um die beiden Netze LAN und ISDN so zu integrieren, dass sie ein Subnetz bilden, ist die Proxy ARP-Funktion im Router nötig. Diese Funktion soll es ermöglichen, für die LAN-Endsysteme die ISDN-Endsysteme unter einer MAC-Adresse x, d.h. des Router-Ports seitens des LANs, zu verbergen. Die Proxy-ARP-Funktion besteht in diesem Fall darin, dass eine besondere ARP-Tabelle im Router an dessen LAN-Port mit der MAC-Adresse x enthalten ist. In dieser Tabelle werden die IP-Adressen von ISDN-Endsystemen eingetragen, und deren IP-Adressen wird die MAC-Adresse x des Routers von der LAN-Seite zugeordnet. Mit einer solchen ARP-Tabelle wird den LAN-Endsystemen mitgeteilt, dass die ISDN-Endsysteme unter der MAC-Adresse x des Routers zu erreichen sind.

Liegt bei einem LAN-Endsystem ein IP-Paket, das an ein ISDN-Endsystem z. B. mit der IP-Adresse y gesendet werden soll, so prüft dieses LAN-Endsystem zunächst, ob das Ziel sich im gleichen Subnetz befindet. Da dies gerade der Fall ist, wird das IP-Paket in einem MAC-Frame direkt an das Ziel gesendet. Ist die Ziel-MAC-Adresse dem Quell-Endsystem unbekannt, so sendet es nach dem Protokoll ARP eine Broadcast-Nachricht ARP-Request an alle Systeme in dessen Subnetz. Diese Broadcast-Nachricht wird auch vom Router empfangen, der mit einem ARP-Reply antwortet, indem der IP-Adresse y des ISDN-Endsystems die MAC-Adresse x zugeordnet wird. Nach dem Empfang von ARP-Reply vermerkt das Quell-Endsystem in seinem ARP-Cache, dass der IP-Adresse y die MAC-Adresse x entspricht. Somit wird der MAC-Frame im nächsten Schritt direkt an den Router abgeschickt. Der Router leitet gemäß der Routing-Tabelle das empfangene IP-Paket an den ISDN-Port weiter.
Mit Hilfe der Proxy-ARP-Funktion ist es möglich, mehrere physikalische Netze mit Hilfe eines Router so zu koppeln, dass sie ein heterogenes Netz bzw. Subnetz bilden und damit nur eine (Sub)Netz-ID besitzen. Es ist hierbei darauf hinzuweisen, dass eine Proxy-ARP-Lösung eine Not-Lösung ist, wenn man kein Subnetting realisieren kann. Wäre Subnetting möglich, so sollte man dem Ethernet LAN eine Subnetz-ID und dem ISDN eine weitere Subnetz-ID zuweisen. Bei einer derartigen Lösung ist die Proxy-ARP-Funktion im Router nicht nötig.

Unterschiedliche LANs können mit der Proxy-ARP-Funktion ein Subnetz bilden. In diesem Fall stellen zum Beispiel Ethernet und Token-Ring zwei getrennte Broadcast-Netze dar. An dieser Stelle ist hervorzuheben, dass ARP-Nachrichten die Nachrichten der MAC-Schicht sind, so dass sie über den Router nicht weitergeleitet werden können. Dies bedeutet, dass eine Broadcast-Nachricht aus dem Ethernet-Teil den Token-Ring nicht erreichen kann. Umgekehrt können die Broadcast-Nachrichten aus dem Token-Ring die Ethernet-Seite nicht erreichen.

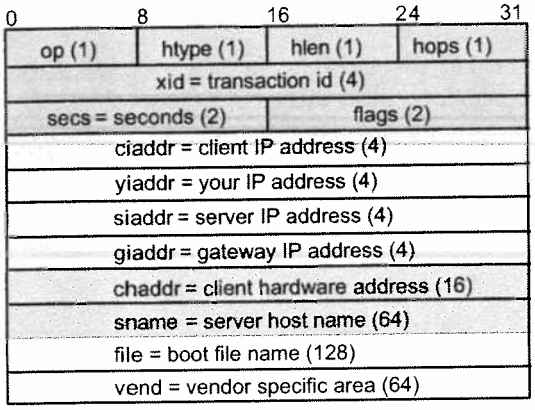
Mit der Proxy-ARP-Funktion im Router kann ein solcher Effekt erreicht werden, dass die Endsysteme am Ethernet den Eindruck gewinnen, die Token-Ring-Endsysteme wären am Ethernet angeschlossen. Umgekehrt wird den Token-Ring-Endsystemen vorgemacht, dass sich ihre Kommunikationspartner am Token-Ring statt am Ethernet befänden. Eine solche Täuschung ist mit Hilfe entsprechender ARP-Tabellen möglich. Eine Tabelle seitens des Ethernet, signalisiert den Ethernet-Endsystemen, dass die Token-Ring-Endsysteme unter der MAC-Adresse g zu erreichen sind. Dabei handelt es sich um die MAC-Adresse des Ethernet-Ports im Router. Die zweite ARP-Tabelle seitens des Token-Ring-LANs signalisiert den Endsystemen am Token-Ring, dass die Ethemet-Endsysteme unter der MAC-Adresse h erreichbar sind.

In einigen Fällen kann es sinnvoll sein, Endsystemen auf unterschiedlichen Medien (z. B. Ethernet und Token-Ring, bzw. Ethernet und FDDI) das gleiche IP-(Sub-)Netz zu definieren. In diesem Fall stellt die Proxy -ARP-Funktionalität ein geeignetes Instrument zur Kopplung dieser Frontend Netze an das Hochgeschwindigkeitsnetz Backend-Netz zur Verfügung, wo z. B. ein Host an seinem FDDI-Interface zwei getrennte IP-Adressen in den IPNetzen A und B zugewiesen bekommt. Über diese IP-Adressen ist er dann sowohl für Stationen am Ethernet über das IP-Netz A wie auch für Rechner am Token-Ring am IP-Netz B transparent erreichbar. Die Router mit Proxy -ARP-Funktionaliät gewährleisten hierbei nicht nur die Umsetzung der MAC-Adressen, sondern auch die notwendige Fragmentierung der IP-Pakete entsprechend der maximalen MTU für das jeweilige LAN.

Proxy-ARP ist insbesondere dann hilfreich, wenn Endsysteme (Hosts) den Internet-Standard für die Subnetz-Adressierung nicht unterstützen, d.h. sie unterstützen kein benutzerspezifisches Subnetting. In diesem Fall handelt es sich um die alte Generation von Endsystemen (Hosts).

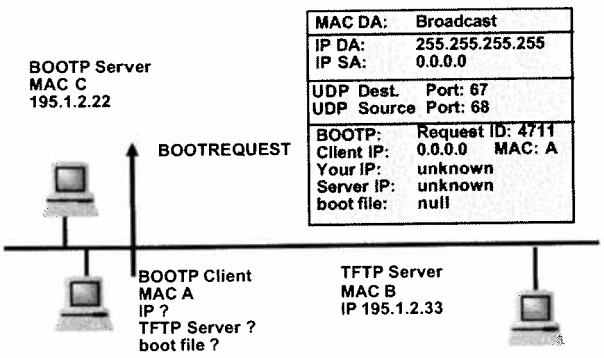
BOOTP (Bootstrap Protocol)

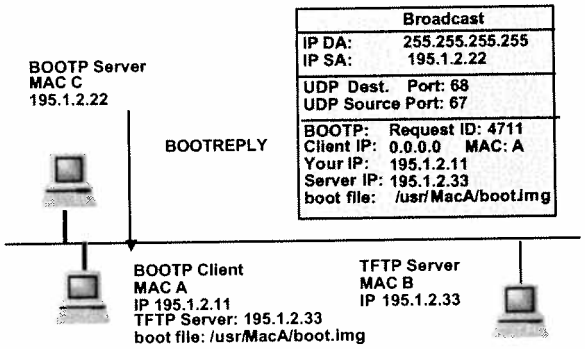
BOOTP (RFC 951, RFC 1542, RFC 1532) ist eine Alternative zu RARP, die es Endsystemen ohne Festplatte erlaubt, ihre eigene IP-Adresse (und weitere Startinformationen wie Router-Adresse, Name-Server-Adresse und Subnetz-Maske) durch Anfrage bei einem Server herauszufinden, eine Datei in den Arbeitsspeicher zu laden und auszuführen. BOOTP benutzt TFTP (Trivial File Transfer Protocol) für die Dateiübertragung und UDP. BOOTP ist ein Protokoll der Anwendungsschicht.



- **Operation code:** 1: BOOTREQUEST, 2: BOOTREPLY
- **hardware address type,** same assigned numbers as ARP (= 1 für 10Mbit/s Ethernet).
- **hardware address length:** Länge der Hardware-Adresse, d.h. physikalischen Netzadresse (6 Bytes für eine MAC- Adresse).
- **number of hops;** client sets to zero, incremented by gateways in case of cross-gateway booting.
- **transaction id;** used to match this boot request with the responses it generates.
- **seconds;** filled in by client, seconds elapsed since client started trying to boot.
- **flags:** MSB is broadcast bit; shall be set if client is unable to receive unicast messages until he knows its IP address.

- **client IP address;** filled in by client in bootrequest if known.
- **your (client) IP address;** filled by server if client doesn't know its own address (ciaddr was 0).
- **server IP address;** IP address of server holding the boot file (TFTP server), returned in bootreply by BOOTP server.
- **gateway IP address;** used in optional cross-gateway booting.
- **client hardware address;** filled-in by client.
- **optional server host name,** null terminated string.
- **boot file name;** null terminated string; 'generic' name or null in bootrequest, fully qualified directory-path name in boot-reply
- **optional vendor-specific area,** (RFC 2132), zum Beispiel information about subnet mask, list of routers in preference order, time server, name server, DNS-Server, host name, boot file size ...





DHCP (Dynamic Host Configuration Protocol)

DHCP (Dynamic Host Configuration Protocol) beschrieben in RFCs 1533, 1534, 1541, 1542) ist eine erweiterte Version des BOOTP-Protokolls, die zusätzlich die Fähigkeit einer automatischen und dynamischen Belegung mit wiederverwendbaren (also nicht fest einer Station zugeordneten) IP-Adressen und von Konfigurationsoptionen bietet. Eine dynamische Belegung mit IP-Adressen ist beispielsweise für drahtlose LANs erforderlich. Die IP-Adresse und die zugehörige Subnet-Maske werden für eine bestimmte Zeit (Lease-Dauer) einem Station zur Verfügung gestellt. DHCP ist interoperabel mit Endsystemen, die BOOTP benutzen, und soll BOOTP langfristig ablösen. DHCP ist - wie BOOTP - ein Protokoll der Anwendungsschicht.

Dynamische Vergabe und Ermittlung von IP-Adressen

Durch die Vergabe von IP-Adressen können Rechner in IP-Netzen und speziell im Internet angesprochen werden. Im intuitiven Umgang sind IP-Adressen jedoch nicht intuitiv genug. Es ist sinnvoll, statt einer IP-Adresse einen Rechner über seinen Namen zu adressieren. Dies kann im Prinzip durch eine statische Tabelle - die Host-Datei - erfolgen; sobald aber eine Vielzahl Rechner in entfernten IP-Netzen, d.h. speziell im Internet, erreicht werden sollen, wird die Pflege der Host-Dateien schnell unhandlich. Um das Problem der dynamischen Namensauflösung im Internet zu lösen, wurde das Domain Name System (DNS) geschaffen. Das Domain Name System stellt eine verteilte Datenbank dar, die im Grunde genommen mit ihrem Informationsgehalt das Internet abbildet. Entsprechend der Bedeutung des DNS für das Internet hat sich die Vergabe dynamischer IP-Adressen im Intranet entwickelt. Über das Dynamic Host Configuration Protocol (DHCP) kann eine dynamische und konsistente Vergabe von IP-Adressen und anderen wichtigen IP-Informationen für Rechner im Intranet erreicht werden.

Protokoll DHCP

Um Endsysteme ohne Festplatte als in TCP/IP-Netzen zu starten und automatisch zu konfigurieren, wurde das Protokoll BOOTP (BOOT Protocol) entwickelt (RFC 1532). Ein Rechner ohne Festplatte ist normalerweise nicht in der Lage, seine IP-Adresse, die benötigten Programme seines Betriebssystems oder den TCP/IP-Programmcode in ausgeschaltetem Zustand zu speichern. Das Protokoll BOOTP soll solche Rechner in die Lage versetzen, alle für den Betrieb am TCP/IP-Netz benötigten Informationen von einem BOOTP-Server abzurufen. Dabei handelt es sich um einen Rechner im Netz, der auf eingehende BOOTP-Anforderungen ständig wartet und die Antworten auf die Anforderungen erzeugt. Da heutzutage Rechner ohne Festplatte am Netz nur selten sind, hat das Protokoll BOOTP an Bedeutung verloren.

Das Protokoll DHCP (Dynamic Host Configuration Protocol) kann als eine BOOTP neue und erweiterte Generation des Protokolls BOOTP gesehen werden. Mit Hilfe des Protokolls DHCP ist es möglich, die IP-Adressen und anderen zusätzliche Konfigurationsparameter jenen Rechnern automatisch zuzuweisen, die für die Nutzung von DHCP konfiguriert sind. In diesen Rechnern muss das Protokoll DHCP implementiert werden. Mit Hilfe des Protokolls DHCP ist es somit möglich, sämtliche TCP/IP-Konfigurationsparameter zentral zu verwalten und zu warten. Insofern besteht auch die Möglichkeit die Endsysteme in TCP/IP-Netzen nach dem Plug-and-Play Prinzip zu installieren. Jeder einzelne Rechner in einem Netz muss sowohl über einen eindeutigen Namen als auch eine eindeutige IP-Adresse verfügen, um mit anderen Rechnern kommunizieren zu können. Die IP-Adressen können dem Rechner entweder manuell oder automatisch zugewiesen werden. Bei der manuellen Zuweisung handelt es sich um statische IP-Adressen, die ein Administrator manuell konfigurieren und bei Bedarf neu zuordnen muss. Bei der dynamischen Zuweisung wird einem Rechner automatisch eine IP-Adresse zugewiesen, wenn er eingeschaltet wird. In diesem Fall spricht man von dynamischen IP-Adressen. Durch den Einsatz des Protokolls DHCP lassen sich vor allem jene Probleme beseitigen, die mit dem manuellen Konfigurieren von IP-Adressen verbunden sind. Die erste Version des Protokolls DHCP wurde Ende 1993 im RFC 1541 veröffentlicht und als Standard im März 1997 durch den RFC 2131 mit einer neuen DHCP-Version abgelöst.

Das Protokoll DHCP funktioniert nach dem Client/Server-Prinzip. Ein DHCP-Server ist ein Rechner, in dem sämtliche Konfigurationsparameter für die Rechner (oft nur innerhalb eines Subnetzes) abgespeichert worden sind. Die Rechner, die auf den DHCP-Server zugreifen, um bestimmte Konfigurationsangaben abzufragen, werden als DHCP-Clients bezeichnet. Wenn ein DHCP-Client gestartet wird, fordert er von einem DHCP-Server die Information über dessen Konfigurationsparameter (wie IP-Adresse und Subnet Mask) an. Optional kann der Client auch zusätzliche Angaben wie z.B. die Adressen von Routern (Default-Gateway), Domain Name Server (DNS) beim Server abrufen. Diese zusätzlichen Konfigurationsparameter werden als Optionen beim Protokoll DHCP definiert. Die Beschreibung von allen derartigen Optionen enthält das Dokument RFC 1533.

Es ist hervorzuheben, dass einige Rechner weiterhin manuell konfiguriert werden können. Oft handelt es sich hierbei um die Rechner mit der Kommunikationssoftware der alten Generation, so dass das Protokoll DHCP nicht unterstützt werden kann (Nicht-DHCP-Client).

Beim DHCP-Protokoll können sogenannte DHCP-Relay-Agenten implementiert werden. Ein solcher Agent hat die Aufgabe, DHCP-Nachrichten in andere Subnetze weiterzuleiten, die nicht über einen eigenen DHCP-Server verfügen. Ein Relay-Agent wird entweder in einen IP-Router oder in einen für diesen Zweck konfigurierten Rechner implementiert. Der Einsatz von Relay-Agenten hat den Vorteil, dass nicht für jedes Subnetz ein eigener DHCP-Server zur Verfügung gestellt werden muss.

Andererseits besteht die Gefahr, dass beim Ausfall eines DHCP-Server einige Clients nicht in der Lage sind, am Netzbetrieb teilzunehmen. Es ist deswegen erforderlich, sowohl redundante DHCP-Server als auch redundante DHCP-Relay-Agenten immer einzuplanen. Aus diesen Gründen lässt das Protokoll DHCP mehrere DHCP-Server sowie mehrere DHCP-Relay-Agenten zu.

Bei der dynamischen Zuweisung wird einem Rechner eine IP-Adresse für einen bestimmten Zeitraum zugeteilt. Dieser Zeitraum wird mit dem englischen Wort Lease bezeichnet. Der Rechner kann aber auch selbst die Adresse vorher wieder freigeben, wenn er sie selbst nicht mehr benötigt. Der Vorteil besteht darin, dass eine von einem DHCP-Client nicht mehr benötigte IP-Adresse an einen beliebigen anderen DHCP-Client vergeben werden kann.

Wenn beim DHCP-Server eine Anforderung eintrifft, wählt er die IP-Adresse aus einem Pool von IP-Adressen aus und bietet sie dem DHCP-Client an. Falls der Client die angebotene IP-Adresse akzeptiert, wird sie ihm für einen festgelegten Zeitraum (Lease) zur Verfügung gestellt. Wenn keine IP-Adressen mehr im Pool beim DHCP-Server vorhanden sind, kann einem Client auch keine Adresse zur Verfügung gestellt werden, so dass er nicht initialisiert werden kann.

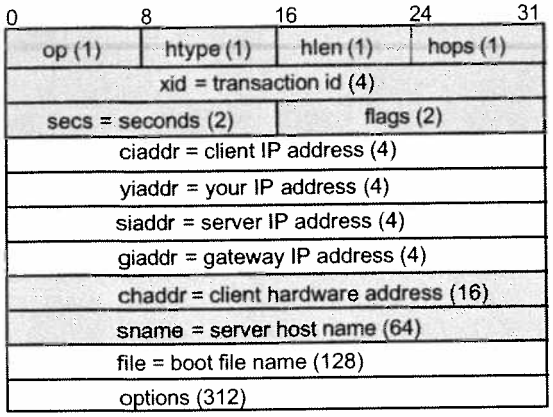


Bild: DHCP Nachrichtenformat

Aufbau von DHCP-Nachrichten

Zwischen einem DHCP-Client und einem DHCP-Server werden festgelegte DHCP-Nachrichten übermittelt, für die das verbindungslose Protokoll UDP eingesetzt wird. Der DHCP-Client stellt einen Anwendungsprozess im einem Rechner dar und ist über den Well-Known Port 68 zu erreichen. Der DHCP-Server ist ein Anwendungsprozess in einem dedizierten Rechner und ist erreichbar über den Well-Known Port 67. Dese Port-Nummern werden im UDP-Header angegeben.

Bemerkung: nur letztes Feld ist in BOOTP und DHCP verschieden.

Die folgenden Felder werden in DHCP-Nachrichten verwendet:

- **op (1 Byte):** Operation: Angabe, ob es sich um eine Anforderung (Request) oder eine Antwort handelt.
- **htype (1 Byte):** Hier wird der Netztyp gemäß RFC 1340 (Assigned Number) mitgeteilt (z.B. 6 = IEEE 802 x-LANs).
- **hlen (1 Byte):** Länge der Hardware-Adresse, d.h. physikalischen Netzadresse (6 für eine MAC- Adresse).
- **hops (1 Byte, optional):** Hier wird die Anzahl von Routern mit der DHCP-Relay-Funktion auf dem Datenpfad angegeben.
- **xid (4 Bytes), Transaktions-ID:** Dies ist die Identifikation für die Transaktion zwischen dem Client und Server, um den DHCP-Clients im Server die Antworten zu den richtigen Anforderungen (Requests) zuordnen zu können.
- **secs (2 Bytes), Sekunden:** Wird vom Client ausgefüllt und bedeutet die Zeit in Sekunden, die seit Beginn des Vorgangs abgelaufen ist.
- **flags (2 Bytes):** Das höchstwertige Bit dieses Feldes zeigt an, ob ein Client in der Lage ist, die IP-Pakete zu empfangen. Ist dies der Fall, verfügt der Client noch über eine gültige IP-Adresse. Die restlichen Bits dieses Feldes werden zur Zeit nur auf 0 gesetzt und sind für zukünftige Zwecke reserviert.
- **ciaddr (4 Bytes), Client-IP-Adresse:** Wird vom Client ausgefüllt, falls er eine IP-Adresse besitzt.
- **yiaddr (4 Bytes), Your-IP-Adresse:** Hier wird die IP-Adresse eingetragen, die der Server dem Client zugewiesen hat.
- **siaddr (4 Bytes), Server-IP-Adresse:** Hier wird die IP-Adresse des Server angegeben (z.B. in der Nachricht DHCP-OFFER), der bei der nächsten Anforderung benutzt werden soll.
- **giaddr (4 Bytes, optional), IP-Adresse des Gateways bzw. Routers** mit der DHCP-Relay-Funktion.
- **chaddr (16 Bytes), Client-NIAC-Adresse**.
- **sname (64 Bytes, optional), Server-Name:** Ein Client, der den Namen eines Server kennt, von dem er Konfigurationsparameter haben will, trägt hier diesen Namen ein und stellt somit sicher, dass nur der angegebene Server auf dessen Anforderung antwortet. Enthält dieses Feld "Alle Bits 0", so kann jeder DHCP-Server im Netz antworten.
- **file (128 Bytes, optional), File-Name:** Der File-Name ist ein alphanumerischer String (Zeichenfolge). Diese Angabe ermöglicht einem DHCP-Client, eine bestimmte Datei zu bestimmen, die er vom Server abrufen will. Der Server ist somit in der Lage, die richtige Datei auszuwählen und sie z.B. mittels des Protokolls FTP dem Client zukommen zu lassen.
- **options (312 Bytes, optional):** Zusätzliche bzw. herstellerspezifische Konfigurationsparameter. Dieses Feld enthält sogenannte DHCP-Optionen, die im Dokument RFC 1533 festgelegt sind.

Ablauf von DHCP

Der Einsatz des Protokolls DHCP zur automatischen Konfiguration der IP-Adressen bedeutet, dass der Benutzer eines Rechners keine IP-Adressierungsinformationen mehr von einem Administrator benötigt, um TCP/IP-Parameter zu konfigurieren. Der DHCP-Server stellt allen DHCP-Clients die erforderlichen Konfigurationsinformationen zur Verfügung.

Das Protokoll DHCP lässt mehrere DHCP-Server zu. Ein wichtiger Grund dafür ist die Server-Verfügbarkeit. Fällt ein Server aus, werden seine Funktionen automatisch durch andere Server übernommen.

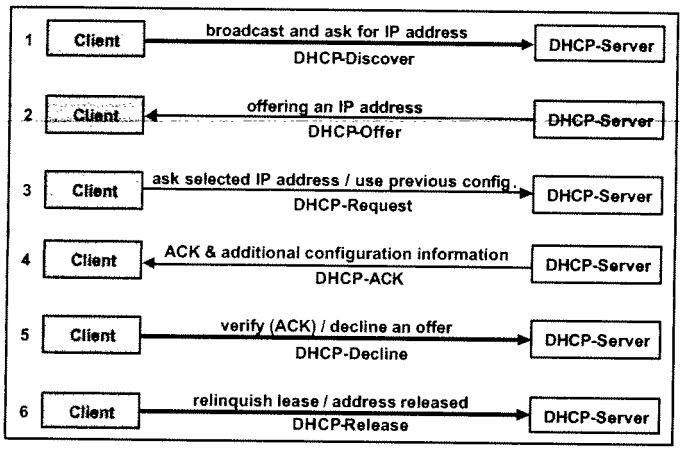


Bild: Ablauf des DHCP-Protokolls

Lease-Aufbau
1) Anforderungsphase: DHCP-Discover
2) Angebotsphase: DHCP-Offer
3) Auswahlphase: DHCP-Request
4) Bestätigungsphase: DHCP-AcK
5) Bestätigungsphase: : DHCP-Decline

Lease-Abbau
6) Abbauphase: DHCP-Release

Es sind vier Phasen nötig, um einem Rechner eine IP-Adresse zuweisen zu können:

1) Anforderungsphase: Der Client sendet die Nachricht DHCP-Discover in einem IP-Broadcast-Paket (Ziel-IP-Adresse = 255.255.255.255) als eine Anforderung, um von einem Server die benötigten IP-Adressierungsinformationen (IP-Adresse, Subnet Mask etc.) zu bekommen. Ein Wert, der unbedingt in dieser Nachricht angegeben werden muss, ist die MAC-Adresse des Clients (im Feld: chaddr).

Die Client-Anforderung DHCP-Discover als Broadcast wird normalerweise auf das eigene Subnetz eingeschränkt. Diese Client-Anforderung kann aber über eventuell vorhandene DHCP-Relay-Agenten in die weiteren Subnetze weitergeleitet werden. Der Einsatz von DHCP-Relay-Agenten hat dann eine große Bedeutung, wenn nicht alle Subnetze über ihre eigenen DHCP-Server verfügen.

2) Angebotsphase: Jeder DHCP-Server kann mit einer Nachricht DHCP-Offer dem Client sein Angebot von IP-Adressierungsinförmationen zukommen lassen. Der Server versucht zuerst, dem Client direkt das Angebot zu senden. Aber dies ist nicht immer möglich. Hierbei sind zwei Fälle zu unterscheiden:

1. Der Client wird gerade initialisiert, so dass er noch über keine eigene IP-Adressse verfügt. In diesem Fall sendet der Server sein Angebot als Broadcast-Nachricht (IP-Adresse 255.255.255.255). Diese DHCP-Nachricht enthält bereits die MAC-Adresse des betreffenden Clients, so dass nur der richtige Client diese Nachricht lesen darf.
2. Der Client verfügt bereits über eine IP-Adresse, doch die Lease-Dauer geht zu Ende, so dass er diese Adresse auf die nächste Lease-Periode verlängern möchte. In diesem Fall wird das Angebot vom Server direkt an den Client gesendet.

3) Auswahlphase: In dieser Phase wählt der Client die IP-Adressierungsinformationen des ersten von ihm empfangenen Angebots aus und sendet eine Broadcast-Nachricht DHCP-Request, um das ausgewählte Angebot anzufordern. In der Nachricht DHCP-Request ist der Name des ausgewählten DHCP-Server enthalten. Es kann hier auch die angebotene IP-Adresse mit Hilfe der Option Requested IP Address (Angeforderte IP-Adresse) bestätigt werden.

Die Nachricht DHCP-Request wird als Broadcast verschickt, um allen übrigen DHCP-Server, die möglicherweise seine Angebote für den Client reserviert hatten, mitteilen zu können, dass sich der Client für einen anderen Server entschieden hat. Diese übrigen Server können die reservierten Parameter wieder freigeben, um sie anschließend anderen Clients anzubieten.

4) Bestätigungsphase: Dieser DHCP-Server, der vom Client ausgewählt wurde, antwortet mit der Nachricht DHCP-Ack, die alle Konfigurationsparameter für den Client enthält. Nach dem Empfang von DHCP-Ack und nach dem Eintragen von Parametern wird beim Client der Konfigurationsvorgang beendet. In dieser Phase können eventuell noch die weiteren verspäteten Angebote eintreffen. Sie werden nun vom Client einfach ignoriert.

Das Protokoll DHCP stellt noch weiteren Nachrichten zur Verfügung:

- **DHCP-Nak:** Diese Nachricht wird in der Bestätigungsphase verwendet und dann von einem ausgewählten DHCP-Server an einen Client gesendet, um darauf zu verweisen, dass die in der Nachricht DHCP-Request geforderten Konfigurationsparameter abgelehnt wurden. Dies kann dann erfolgen, wenn:
  - ein Client versucht, die Lease für seine bisherige IP-Adresse zu verlängern und diese Adresse nicht mehr verfügbar ist.
  - die IP-Adresse ungültig ist, weil der Client in ein anderes Subnetz umgezogen ist.
- **DHCP-Release**: Mit dieser Nachricht teilt ein DHCP-Client einem Server mit, dass einige Parameter (z.B. IP-Adresse) nicht mehr benötigt werden. Damit werden diese Parameter freigegeben und stehen anderen Clients zur Verfügung; dies müssen die Remote-PCs am ISDN tun.
- **DHCP-Decline:** Mit dieser Nachricht teilt ein DHCP-Client dem Server mit, dass einige "alte" Parameter (wie z.B. dessen MAC-Adresse) ungültig sind.
- **DHCP-Inform:** Diese Nachricht ist nur in neuen Protokoll DHCP enthalten (RFC 2131). Diese Nachricht kann ein Client nutzen, dem eine statische IP-Adresse manuell zugeteilt wurde, doch er möchte dynamisch zusätzliche Konfigurationsparameter vom DHCP-Server zugeteilt bekommen.

Alle DHCP-Clients versuchen, ihre Lease zu erneuern, sobald die Leasedauer zu 50 Prozent abgelaufen ist. Um seine Lease zu erneuern, sendet der Client eine Nachricht DHCP-Request direkt an den DHCP-Server, von dem er zuvor die Konfigurationsparameter erhalten hat. Der DHCP-Server bestätigt dies dem Client mit einer Nachricht DHCP-Ack, in der eine neue Lease-Dauer und alle aktualisierten Konfigurationsparameter enthalten sind. Wenn der Client diese Bestätigung erhält, aktualisiert er entsprechend seine Konfigurationsparameter.

Versucht ein Client, seine Lease zu erneuern, ist jedoch der gewünschte DHCP-Server nicht erreichbar, kann der Client die Parameter (IP-Adresse) dennoch weiter verwenden, weil noch 50% der Lease-Dauer verfügbar ist. Wenn die Lease nach dem Ablauf von 50% der Dauer nicht vom ursprünglichen DHCP-Server erneuert werden konnte, versucht der Client nach Ablauf von 87,5% der Lease-Dauer, einen anderen DHCP-Server in Anspruch zu nehmen. Hierfür sendet der Client eine Broadcast-Nachricht DHCP-Request.

Jeder beliebige DHCP-Server kann darauf antworten:
- mit einer Nachricht DHCP-Ack, wenn er diese Lease erneuert hat, oder
- mit einer Nachricht DHCP-Nak, wenn er den DHCP-Client zur Neuinitialisierung und Übernahme einer neuen Lease für eine andere IP-Adresse zwingen will.

Wenn ein DHCP-Client neu gestartet wird, versucht er zuerst vom ursprünglichen DHCP-Server dieselbe IP-Adresse zu erhalten. Er erreicht dies, indem er einen DHCP-Request als Broadcast verschickt und die zuletzt erhaltende IP-Adresse angibt. Wenn dies keinen Erfolg hat und die Lease-Dauer noch nicht zu Ende ist, kann der DHCP-Client dieselbe IP-Adresse über die verbleibende Lease-Dauer noch verwenden.

Wenn die Lease-Dauer abläuft oder eine Nachricht DHCP-Nak empfangen wird, muss der DHCP-Client unmittelbar die Verwendung der IP-Adresse einstellen und einen neuen Prozess der Vergabe von neuen IP-Adressen starten. Ist die Lease bei einem Client abgelaufen, der keine neue Lease erhalten hat, wird die TCP/IP-Kommunikation so lange eingestellt, bis eine neue IP-Adresse zugewiesen werden kann.

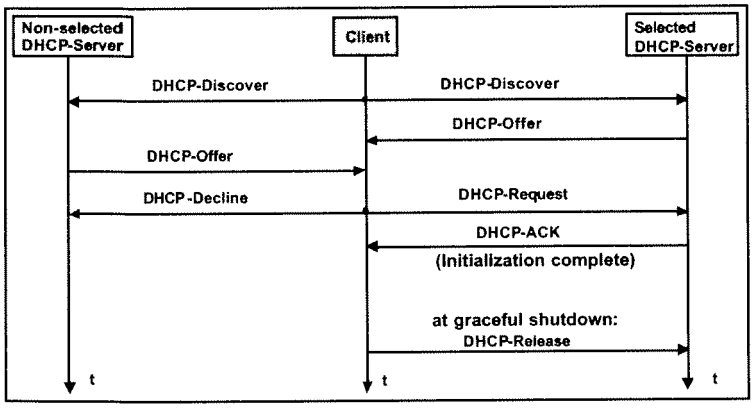


Bild: Netz mit mehreren DHCP-Server

Netz mit mehreren DHCP-Server

Wenn in einem Netz mehrere DHCP-Server benötigt werden, muss ein eindeutiger Bereich von IP-Adressen für jedes Subnetz geplant werden. Ein Pool von IP-Adressen ist eine Folge von IP-Adressen, die für die Vergabe an Clients zur Verfügung stehen. Um sicherzustellen, dass Clients möglichst immer eine IP-Adresse erhalten, ist es wichtig, für jedes Subnetz mehrere Bereiche auf den verschiedenen DHCP-Server zu reservieren.

Im allgemeinen sollte man die verfügbaren IP-Adressen folgendermaßen auf die DHCP-Server verteilen:

- Jeder DHCP-Server sollte über einen Bereich mit ca. 75% der für das eigene Subnetz bestimmten IP-Adressen verfügen.
- Jeder DHCP-Server sollte für jedes Remote-Subnetz über einen Bereich mit ca. 25% der für dieses Remote-Subnetz bestimmten IP-Adressen verfügen.

Wenn der DHCP-Server eines Clients nicht verfügbar ist, kann dieser Client immer noch eine IP-Adresse von einem anderen DHCP-Server zugeteilt bekommen, der sich in einem anderen Subnetz befindet, unter der Voraussetzung, dass ein DHCP-Relay-Agent im Router implementiert ist.

Bei DHCP-Server sind folgende Punkte zu beachten:

- Bevor ein DHCP-Server die IP-Adressen an DHCP-Clients vergeben kann, muss er über einen Bereich von gültigen IP-Adressen verfügen.
- Deshalb ist es notwendig, jedem DHCP-Server eine eindeutige statische IP-Adresse (manuell) zuzuweisen. Der DHCP-Server selbst kann kein DHCP-Client sein.
- Nicht-DHCP-Clients besitzen statische IP-Adressen, die manuell eingegeben werden müssen.
- Die statischen IP-Adressen dürfen im Pool von für DHCP-Clients verfügbaren IP-Adressen nicht enthalten sein.
- Falls die IP-Adressen mit einen DHCP-Server den Clients in mehreren Subnetzen vergeben werden, müssen alle Router, die diese einzelnen Subnetze verbinden, auch als DHCP-Relay-Agenten dienen.
- Werden mehrere Subnetze mit Routern vernetzt, aber diese Router nicht die DHCP-Relay-Funktion unterstützen, ist in jedem Subnetz mit DHCP-Clients zumindest ein DHCP-Server erforderlich.

Das Protokoll DHCP ist Gegenstand mehrerer Internet-Dokumente RFCs 1533, 1534, 1541, 1542 und 2131.

IP-Version 6 (IPv6)

Im Jahre 1992 war abzusehen, dass das bestehende Internet Protokoll (Version 4) an seine Grenzen stoßen wird, vor allem bezüglich des Adressraumes. Deshalb wurde eine Arbeitsgruppe der IETF unter dem Begriff IP next generation (IPng) gegründet, die Vorschläge für ein neues Protokoll erarbeitet hat - heute unter dem Namen IP-Version 6 (IPv6) bekannt.

Ziele für diese neue Version waren:
- IP - Internet Protokoll
- größerer Adressraum,
- automatische Konfiguration (z. B. der Adressen),
- leichteres Routing,
- bessere Netzstrukturierung,
- verbesserte Sicherheitsfunktionen,
- Unterstützung für Echtzeit- und Multimedia-Dienste.

Gelöst wurde dieses durch einen vereinfachten Protokollkopf. Gegenüber den 13 Feldern bei IPv4 (ohne Optionen) sind es nur noch 8 Felder bei IPv6 (ohne Erweiterungsköpfe). Für zusätzliche Funktionen können bei Bedarf Erweiterungsköpfe angefügt werden. Also wenn z. B. eine Sicherheitsfunktion nicht benötigt wird, dann verbraucht sie auch keinen Platz.

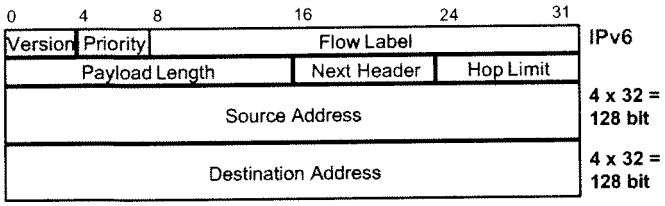


Bild: IPv6 Header

Die Felder im 40 Byte langen IPv6-Header haben die folgende Bedeutung:

- **Version:** enthält den Wert 6.
- **Priority** (Traffic Class): Dringlichkeit des Pakets.
- **Flow Label:** Zur Angabe des Typs der enthaltenen Daten.

- **Payload Length:** Länge der Nutzdaten im Feld, das auf den Header (bzw. die Extension Headers) folgt. Die Länge wird in Byte (Byte) angegeben, der maximale Wert beträgt 64 KByte. Durch ein Erweiterungsfeld (mit Extension Header des Typs Fragmentation) sind größere Werte möglich.
- **Next Header:** Zahl, definiert den Typ eines nächsten Headers, der unmittelbar nach dem Feld Zieladresse folgen kann.
- **Hop Limit:** Zahl (anfänglicher Maximalwert ist 254), die in jedem Zwischensystem dekrementiert (um 1 verringert) wird. Das Datagramm wird vernichtet, falls der Wert 0 wird, bevor das Ziel erreicht ist.
- **Quell- und Zieladresse:** Die Länge der Adressen beträgt 16 Byte (128 Bit), damit beträgt die Größe des Adressraums 3,4 x 10^{+38}. Die Adressen sind hierarchisch aufgebaut um den Routing-Aufwand zu verringern. Eine sog. Cluster-Adresse bezeichnet eine geografische Region des Netzes.

Priority (Traffic Class)
Das Prioritätsfeld unterscheidet zwei Verkehrsarten: Lastgesteuerter Datenverkehr, der einer Flusskontrolle unterworfen werden darf, wie sie z. B. TCP bereitstellt, und Echtzeit-Verkehr, der eine konstante Datenrate und eine konstante Verzögerungszeit erfordert. Innerhalb jeder Verkehrsart gibt es 8 Prioritätsklassen.

Flow Label
Mit dem Flow Label wird ein Ansatz In Richtung einer Verbindung innerhalb des verbindungslosen IP gemacht: Angenommen, mehrere Pakete gehören irgendwie zusammen, z. B. sie transportieren alle Anteile einer Datei, dann können diese Pakete mit einem einheitlichen Wert für das Flow Label gekennzeichnet werden; dieser Wert wird zufällig bestimmt. Router merken sich Flows und behandeln alle Pakete, die zu einem Flow gehören, in gleicher Weise - wie, das muss vorher über andere Mechanismen (spezielles Protokoll oder manuell) ausgehandelt werden. Da es kein definiertes Ende eines Flows gibt, können die Router nur über die Alterung den Zustand eines Flows wieder löschen.

Hop Limit
Das Feld Hop Limit entspricht grob der Funktionalität des Time-to-Live Feldes in IPv4. Es hatte sich herausgestellt, dass die Angabe einer wirklichen Zeit keinen Vorteil bringt, mehr noch, die gängigen Implementierungen sowieso schon das Hop Limit als reinen Hop Count, also die Anzahl Router auf dem Weg, verwendet haben.

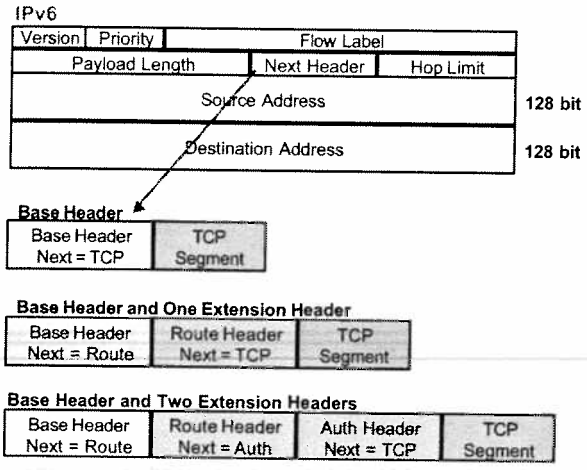


Bild: Extension Header in IPv6

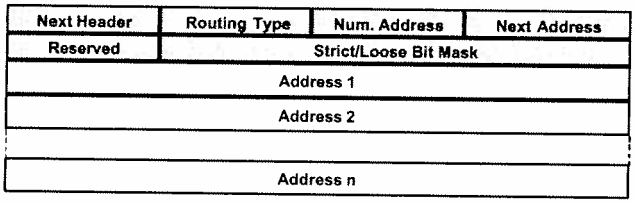
Next Header

Das Feld Next Header zeigt auf den ersten Erweiterungs-Header, soweit ein solcher vorhanden ist, oder auf das Transportprotokoll, vergleichbar dem Protokoll-Feld in IPv4. Wenn Erweiterungs-Header notwendig sind, dann werden diese durch das Feld Next Header angekündigt.

Zusätzliche Information kann in **bis zu sechs weiteren Header** (extension header) angegeben werden. Dabei ist eine Reihenfolge vorgegeben.

1) Hop-by-Hop Options Header,
2) Routing Header,
3) Fragment Header,
4) Authentication Header,
5) Encapsulating Secure Payload (ESP) Header,
6) Destination Options Header.

1) Hop-by-Hop Options Header: Dieser Header ist der einzige, der von jedem Router auf dem Weg zum Ziel ausgewertet wird, alle anderen Erweiterungs-Header haben nur Ende-zu-Ende-Relevanz.



- Strict ⇒ Discard if Address [Next-Address] <> neighbor
- Type = 0 ⇒ Current source routing
- Type > 0 ⇒ Policy based routing (later)
- New Functionality: Provider selection, Host mobility, Auto-readdressing (route to new address)

Bild: Routing Header

2) Routing Header: Damit wird der Weg durch das Netz explizit angegeben. Dazu enthält der Routing Header eine Tabelle von Router-Adressen. Man nennt dies Source Routing, da die Quelle den kompletten Weg vorgibt. Bis zu 24 IPv6-Adressen können angegeben werden.

3) Fragment Header: Im Gegensatz zu IPv4 wird in IPv6 keine Fragmentierung auf dem Weg zum Ziel durchgeführt, sondern diese Aufgabe wird an die sendende Station delegiert. Sollte auf dem Weg ein Router ein zu großes Paket erhalten, so sendet er eine ICMP-Nachricht an den Sender mit der Meldung, dass zu fragmentieren ist. Der Sender wird dann sein Datenpaket in Fragmente zerlegen, jedes in ein eigenes IPv6-Paket verpacken und mit einem Fragment-Header versehen. Dieser enthält die notwendigen Elemente, wie sie in IPv4 schon im Protokollkopf vorgesehen sind: Identification, Offset und More-Fragments-Flag. Die minimale Paketgröße wurde von 576 Byte bei IPv4 auf 1280 Byte bei IPv6 angehoben.

4) Authentication Header: Dieser Header bei IPv6 und Encapsulation Security Header dienen der Authentisierung und Verschlüsselung der Daten. Er enthält eine Prüfsumme, welche die Authentifikation des Senders erlaubt.

5) Encapsulating Secure Payload (ESP) Header: Dieser Header enthält die Schlüsselnummer und die verschlüsselten Nutzdaten

6) Destination Options Header: Dieser Header transportiert zusätzliche Empfänger-Informationen und interessiert nur den Empfänger. Das Format des Inhalts muss vorher vereinbart werden.

Koexistenz von IPv4 und IPv6 - Migration

Ein Wechsel von IPv4 zu IPv6 an einem Stichtag ist wegen der Größe des Internet nicht möglich und auch nicht wünschenswert. Deshalb müssen die beiden Versionen über längere Zeit koexistieren. Die Herausforderung liegt damit im Zusammenwirken von Netzen auf Basis IPv4 mit Netzen auf Basis IPv6: Regeln für den Transport des Protokolls durch Domänen des anderen Protokolls sowie Übergänge sind notwendig und wurden inzwischen auch beschrieben.

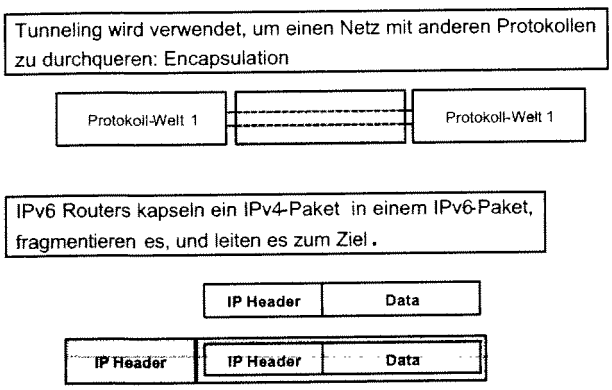


Bild: Tunneling

Koexistenz von IPv4 und IPv6 - Migration

Es gibt zwei Möglichke iten:

- **Tunneln:** IPv6-Pakete werden durch IPv4 unverändert weitergegeben. Die Pakete werden also nur in Systemen (Routern) ausgewertet, die auf IPv6 ausgelegt sind.
- **Parallele Implementierung** von IPv4 und IPv6 *(dual stack operation):* Systeme, auf denen beide Protokollstapel zur Verfügung stehen, können mit beiden Versionen kommunizieren.

Eine **Migration** von IPv4 zu IPv6 ist damit für einzelne Systeme, unabhängig von ihrer Umgebung, möglich. Bezüglich Sicherheit ist **IPSEC** fester Bestandteil von IPv6.

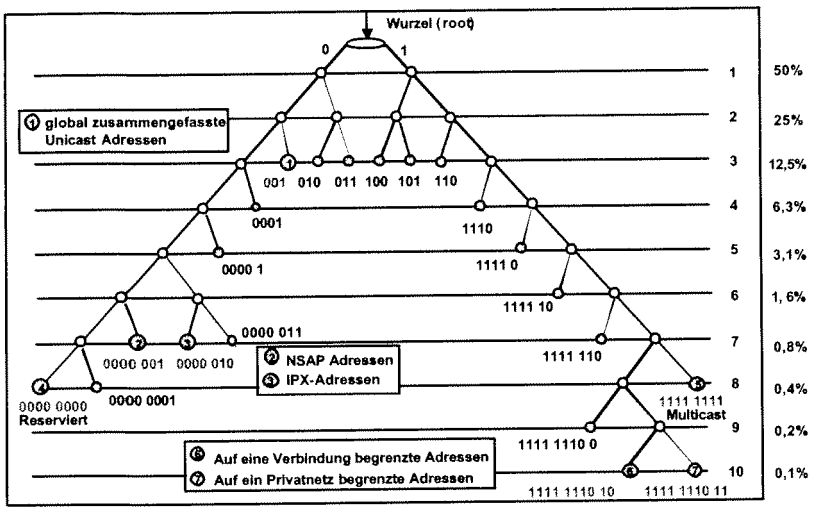


Bild: IPv6-Adressen: Formatpräfix

Im Gegensatz zur einfachen Unterteilung von Adressen nach Klassen A bis E in Netzen beim IPv4 ist die Unterscheidung von Adresstypen beim Protokoll IPv6 sehr flexibel und somit aufwendiger. Um welchen Adresstyp es sich handelt, bestimmt ein Formatpräfix, der eine variable Länge besitzt und durch die ersten (von links gelesen) Bits bestimmt wird. Durch diesen Präfix kann der ganze Adressraum in Adressklassen aufgeteilt werden.

3.2b Internet-Referenzmodell: Internetschicht

Version: Dez. 2003

Bemerkung vorab: Die genauen Formate der Routing-Protokolle dienen zur Informationsergänzung. Der Zweck der verschiedenen Formatfelder der einzelnen Routing-Protokollen soll jedoch erklärt werden können. Die Feldbezeichnungen werden jeweils angegeben.

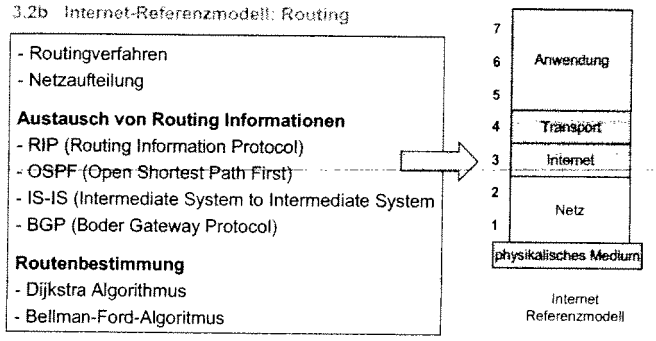


Bild: Übersicht

IP-Routing

Endsysteme und Router-Endsysteme sind für den Transport von IP-Paketen (Datagrammen) auf Router angewiesen. Ein Router gehört mindestens zu zwei Teilnetzen, zwischen denen er IP-Pakete weiterreichen kann. Dieser Vorgang wird als forwarding bezeichnet. Router werten die IP-Zieladresse eines Paketes aus, stellen auf Grund ihrer Routing-Tabelle fest, zu welchem nächsten Router das Paket weiterzuleiten ist, und geben es an der entsprechenden Netzschnittstelle (Port) aus.

Grundlagen von Routing

Mit Routern kann man Vernetzungen realisieren, in denen die optimalen Wege (Routen) für die Datenübertragung nach verschiedenen Kriterien bestimmt werden. Als **Kriterien** können Auslastung, Durchsatz, Gebühren und Übertragungszeit in Betracht kommen. Bei einer Änderung der Lage im Netz (z.B. Leitungsunterbrechung, Router-Ausfall) sollten die Router auf einen alternativen Weg umschalten können.

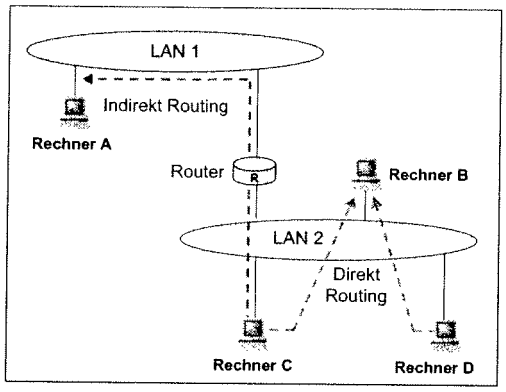


Bild: Direktes and indirektes Routing

Der Ablauf beim IP-Routing unterscheidet mehrere Fälle:

- **Direct routing:** Ziel und Quelle sich in demselben Subnetz befinden.
- **Source routing:** die Quelle gibt den Weg vor.
- **Indirect routing:** Normalfall. Die Adresse des Ziels wird in der Routing-Tabelle gesucht. Diese enthält die Adresse des nächsten, in Richtung auf das Ziel zu durchlaufenden Routers, an den die Datagramme dann gesendet werden.
- **Default routing:** trift ein, wenn für das Zielnetz kein Eintrag in der Routing-Tabelle existiert. Die Datagramme werden dann an einen Router gesendet, der als Default Gateway spezifiziert wurde. Default routing ist insbesondere für Teilnetze sinnvoll, die nur über einen Router mit der Außenwelt verbunden sind (Stub Network).

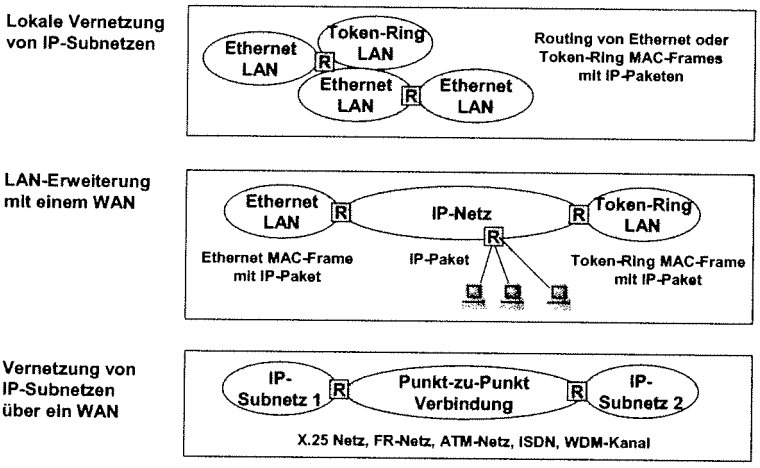


Bild: Einsatzgebiete von Routern in IP-Netzen

Aufgaben von Routern

Ein Router (genauer ein IP-Router) ist im allgemeinen ein Kopplungselement zwischen zwei bzw. mehreren Netzen, der die IP-Pakete auf der Basis von IP-Zieladressen von einem Netz ins andere weiterleitet. Im allgemeinen werden mehrere IP-Subnetze mit einem IP-Router verbunden. Wenn auch Funktionen oder Informationen höherer Schichten einbezogen werden, wird ein Router auch als Gateway bezeichnet.

Die wichtigsten Einsatzgebiete von Routern in IP-Netzen sind:

- lokale Vernetzung der IP-Subnetze auf Basis von LANs,
- Erweiterung eines LANs mit einem WAN,
- standortübergreifende Vernetzung von IP-Subnetzen über ein WAN.

1) Lokale Vernetzung von IP-Subnetzen

Die IP-Pakete in LANs werden in MAC-Rahmen übertragen. Ein Router bei der IP-Kommunikation zwischen zwei LANs leitet IP-Pakete von einem IP-Subnetz ins andere IP-Subnetz. Hierbei wird das IP-Paket aus dem empfangenen MAC-Rahmen herausgenommen (LAN A) und in den zu sendenden MAC-Rahmen (LAN B) eingebettet. Dies bedeutet, dass die Kopplung von LANs mit Router-Hilfe innerhalb der Netzschicht (Schicht 3) stattfindet. Deshalb ist es möglich, dass die beiden LANs unterschiedliche Zugriffsverfahren (MAC) verwenden. Es kann sich hierbei um LANs unterschiedlicher Typen (z.B. IEEE mit Router 802.3/Ethernet und 802.5/Token-Ring) handeln, in denen unterschiedliche MAC-Verfahren verwendet werden.

2) LAN-Erweiterung mit einem WAN

Bei der LAN-Erweiterung mit einem WAN mit einem Router unterscheidet man zwei Fälle:

- **Die Rechner am WAN bilden ein IP-Subnetz.** Es handelt es sich um eine klassische Vernetzung von IP-Subnetzen. Hier besteht die Aufgabe des Routers in der Weiterleitung der IP-Pakete aus einem Subnetz ins andere.
- **Die Rechner am LAN und die Rechner am WAN bilden ein IP-Subnetz.** Hier muss der Router seitens des LAN die Funktion Proxy-ARP (Address Resolution Protocol) unterstützen. Die Aufgabe des Routers besteht in diesem Fall in der Weiterleitung der IP-Pakete aus dem LAN ins WAN und auch umgekehrt. Das WAN kann ein X.25-Netz, Frame-Relay-Netz, ATM-Netz oder ISDN sein. Die wichtigste Voraussetzung ist dabei, dass der Rechner am LAN mit dem Rechner am WAN kommunizieren kann. Hierfür müssen sie das Protokoll IP verwenden. Bei der IP-Kommunikation zwischen einem Rechner am LAN und einem anderen Rechner am WAN leitet der Router nur die IP-Pakete vom LAN ins WAN und umgekehrt. Beispielsweise bei der Übermittlung eines IP-Pakets in WAN-Richtung wird das IP-Paket aus dem empfangenen MAC-Rahmen herausgenommen und in den zu sendenden WAN-Rahmen eingebettet. Wird als WAN das ISDN eingesetzt, verwendet man oft das Protokoll PPP (Point-to-Point Protocol). In diesem Fall werden die IP-Pakete in den PPP-Rahmen übertragen. Das Protokoll PPP wird auch zukünftig bei der direkten Übertragung der IP-Pakete über WANs auf WDM-Basis (Wavelength Division Multiplexing) eingesetzt. Somit können die Gigabit-LANs mit WDM-basierten WANs nach dem hier dargestellten Prinzip räumlich uneingeschränkt erweitert werden.

3) Vernetzung von IP-Subnetzen über ein WAN

Um die IP-Subnetze standortübergreifend über ein WAN zu vernetzen, sind zwei Router nötig. Bei der Übermittlung eines IP-Pakets vom LAN zum WAN wird das IP-Paket im Router aus dem empfangenen MAC-Frame herausgenommen und in den zu sendenden WAN-Frame eingebettet. Der umgekehrte Vorgang findet im Router bei der Übermittlung in Gegenrichtung statt. Bei der Übermittlung vom WAN zum LAN wird das IP-Paket aus dem empfangenen WAN-Frame herausgenommen und in den zu sendenden MAC-Frame des LANs eingebettet.

Routing und Forwarding

Die Routing-Tabellen werden von den Routern selbst aufgebaut, indem sie mit Hilfe von Routing-Protokollen mit anderen Routern Informationen über existierende Pfade austauschen. Aus diesen Informationen können dann mit Hilfe von Routing-Algorithmen geeignete Pfade durch das Netz berechnet werden.

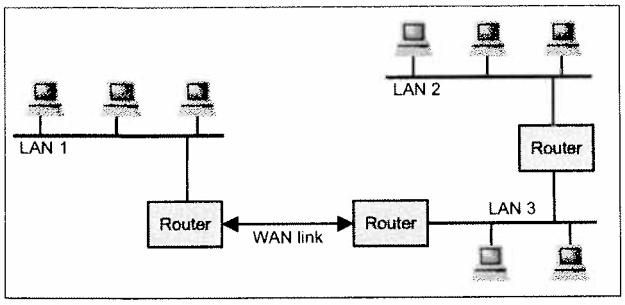


Bild: Router-Vernetzung

Router und Gateways

Router sind sehr komplexe Koppelelemente, die man nicht alleine am Durchsatz messen kann. Trotzdem ist gerade bei Einsatz von Echtzeitapplikationen die Performance eines Routers ein entscheidender Leistungsengpass. Das ist auch nicht verwunderlich, wenn man bedenkt, dass Router multiprotokollfähig sind, verschiedene Routing-Algorithmen verwenden und ebenfalls für das Weiterleiten (Forwarding) der Daten an das richtige Subnetz zuständig sind.

Unter Routing versteht man die Wegefindung über verschiedene Subnetze. Die zentrale Funktion eines Routers besteht darin, die an ihn adressierten Pakete gemäß ihrer Adresse auf der Netzschicht (z.B. IP), in Abhängigkeit verschiedener Entscheidungskriterien, auf einem bestimmten Weg weiterzuleiten. Die Netzschicht hat dabei als Hauptaufgabe, eine Wegewahl für den Datenstrom vorzunehmen. Im Vergleich zur Kopplung von LANs durch Bridges, stellt eine Anbindung über Router eine effizientere Möglichkeit zur Verfügung, um redundante Netzstrukturen bezüglich dynamischer Wegewahl und alternativer Routen auszunutzen. Dazu tauschen sie in Form von Managementprotokollen Informationen unter sich aus, um Wege durch die sie verbindenden Netze zu schalten. Ein weiteres Merkmal ist die Nichttransparenz für alle an einem LAN-Port angeschlossenen Stationen. Das heißt, ein Router verarbeitet und transportiert nur Pakete, die direkt an ihn auf der MAC-Schicht adressiert sind. Pakete, die im selben Subnetz adressiert sind, werden nicht beachtet. Dadurch ergibt sich auch eine Begrenzung von Broadcasts, da Pakete nur dann weitergeleitet werden, wenn deren Zielnetz (Netzschicht) bekannt ist. Das heißt, es

werden sogenannte Broadcast-Domänen aufgebaut. Zusätzlich findet durch den Einsatz von Routern eine Unterteilung in eine hierarchische Netzstruktur statt. Durch den Einsatz der Adresse der Netzschicht lassen sich Netze in Subnetze mit unterschiedlichen Netzadressen unterteilen. Dies kann man im Gegensatz zur flachen Adressierung der MAC-Schicht vornehmen.

Router sind ebenfalls protokollabhängig. Das heißt, der Router fällt die Vermittlungsentscheidung aufgrund der Interpretation des Protokolls auf der Netzschicht. Wenn ihm diese Möglichkeit zur Interpretation fehlt, kann das entsprechende Paket nicht weiter vermittelt werden. Nicht-routebare Protokolle, wie z.B. NetBIOS, können entweder gebridged oder in ein routebares Paket (z.B. TCP/IP) eingepackt und weitergeleitet werden. Gegeben ist aber eine Transparenz gegenüber MAC-Protokollen. Das heißt, solange ein Router über eine entsprechende Schnittstelle verfügt, ist sein Einsatz unabhängig vom verwendeten Schicht-2-Protokoll. Router haben eine eigene Adresse im LAN und müssen bekannt sein. Die Gesamtzuverlässigkeit des Netzes wird durch sie erhöht. Im LAN/WAN-Einsatz tun sich Router besonders bei der Funktionalität des Routings sowie der Begrenzung von Broadcasts auf das Subnetz hervor. Fehlerhafte Pakete der Sicherungsschicht und Netzschicht werden erkannt und nicht weiterbefördert. Somit wird verhindert, dass sie sich auf andere Subnetze ausbreiten. Router können mit äußerst unterschiedlichen Medien und Netzen zusammenarbeiten. Sie unterstützen, im Gegensatz zu Brücken, das Segmentieren, Nummerieren und Wiederzusammensetzen von Paketen. Dieses ist ein wesentlicher Punkt, da Subnetze unterschiedliche Zugriffsverfahren (Ethernet, Token Ring, FDDI, ATM) haben können und im Weitverkehrsbereich Paketgrößen im allgemeinen differieren.

Einteilen kann man Router nach der implementierten Protokollvielfalt und der Fähigkeit, auch als Brücke arbeiten zu können. Ein Single Protocol Router verbindet die angeschlossenen Netze nur über ein Protokoll miteinander. Für alle anderen Schicht-3-Protokolle ist dieser Router unzulässig. Router, die mehrere Protokolle, wie z.B. DECnet, TCP/IP, XNS, IPX/SPX, AppleTalk usw. simultan verarbeiten können, werden Multiprotocol Router genannt. Hier sind sozusagen mehrere Protokollstapel gleichzeitig implementiert. Es handelt sich dabei um unterschiedliche logische Netze, die im Sinne einer LAN-WAN-LAN-Kopplung über einen Single Protocol Router miteinander verbunden werden. Beim Empfang eines Pakets stellt der Router über die (interpretierte) Netzadresse fest, welches Protokoll das Paket hat, und verarbeitet es mit der entsprechenden Protokollroutine. Koppelelemente, die sowohl Bridging als auch Routing erlauben, werden Bridging Router bzw. Hybrid Router genannt. Entscheidend ist diese Funktionalität, wenn es sich um Pakete handelt, die nicht geroutet werden können, weil der Router beispielsweise das entsprechende Protokoll nicht unterstützt. Dies erkennt der Bridging Router nach Interpretation der Kontrollinformation und entscheidet daraufhin, ob es geroutet oder gebridged wird. Für diese fremden Protokolle ist der Router damit wieder transparent.

Die Grundfunktion eines Routers ist somit auf die Wegewahl innerhalb eines vermaschten Netzes beschränkt. Zwischen den Routern der angeschlossenen Netze werden dabei periodisch Routing Informationen verschickt, durch die der Router die vorhandenen Pfade kennenlernt. Diese Werte werden in einer Routing-Tabelle festgehalten. Bei großen Netzen, die aus vielen kleineren Subnetzen bestehen können, kann die Größe der Tabelle sehr stark zunehmen. Um dieses zu verhindern, besteht die Möglichkeit, einzelne Netze im TCP/IP-Bereich in sogenannte Routing-Domänen einzuteilen. Innerhalb der Domänen verwenden die Router eigene Protokolle, die Interior Gateway Protocol (IGP) genannt werden. Routing-Domänen zeigen untereinander ihre Erreichbarkeit durch das Exterior Gateway Protocol (EGP). Die Unterteilung in verschiedene Routing-Domänen reduziert erheblich die Aktualisierungen, die zwischen den verschiedenen Domänen vorgenommen werden müssen. Trotzdem sollte beachtet werden, dass ein Router immer eine zusätzliche Fehlerquelle darstellen kann. Darum ist immer eine genaue Abschätzung des Aufwands zu berücksichtigen, bevor man unnötig Ressourcen verschwendet.

Als Voraussetzung für das Fällen einer Transportentscheidung müssen nun in einem Router zusammenfassend folgende Funktionen realisiert sein:

- Aufbau und Wartung einer Routing-Tabelle: Enthalten sind Informationen, über welche Wege und deren Kosten sowie zusätzliche Transportbedingungen (z.B. Filter) Pakete weitergeleitet werden können.
- Sammeln von Informationen zur Wartung der Routing-Tabelle: Auswertung von Nachrichten anderer Router sowie das Messen von Verzögerungen und Interpretation von Schicht-3-Protokollinformationen.
- Weitergabe der Informationen an andere Router.
- Berechnung und Aufrechterhaltung erforderlicher Verbindungswege, aufgrund der in der Routing-Tabelle enthaltenen Informationen (Routing-Algorithmus).

Einen Routing-Vorgang kann man sich beispielsweise anhand eines Ethernet-Netzes vorstellen, das IP-Pakete vermittelt. Das Ethernet-Paket erreicht zunächst den Router-Port und wird anschließend, da es die MAC-Zieladresse des Routers enthält, zunächst in einem FIFO zwischengespeichert. Danach werden die Daten erst im Hauptspeicher des Routers abgelegt, wobei die CPU einen Interrupt enthält, der sie über die Ankunft des Pakets informiert. Nachdem eine Dienstroutine, die für alle Ethernet-Schnittstellen des Routers zuständig ist, alle Ethernet-Protokollinformationen überprüft hat (Frame Check Sequence, Rahmenlänge usw.) und ein Protokoll Identifier für ein bekanntes Schicht-3-Protokoll entdeckt wurde (IP), wird im nächsten Schritt die Korrektheit des IP-Protokollkopfes festgestellt. Das beinhaltet das Überprüfen der Protokollnummer, Feststellen der Paketlänge und Berechnen der Kopfprüfsumme. Erst danach beginnt die eigentliche Funktionalität eines Routers. Nach der Überprüfung aller gesetzten IP-Adressfilter wird die Routing-Tabelle eingesehen. Ist hier für die IP-Adresse keine Route vorgesehen, schickt der Router eine Fehlermeldung an den Absender des Pakets und beendet damit die Verarbeitung. Ist

hingegen eine Route vorgesehen, so prüft der Router die maximale Paketgröße, die auf dem Pfad erlaubt ist und führt gegebenenfalls eine Fragmentierung des Pakets durch. Anschließend werden die weiteren IP-Schritte durchgeführt, wie die Berechnung der Prüfsumme und die Aktualisierung des Feldes Time-to-Live (TTL). Das Paket gelangt danach in die Sendewarteschlange (Sending Queue) der LAN Network Interface Card (NIC) des Zielweges. Dieses Paket ist für die Einkapselung der Daten in die entsprechende MAC-Schicht verantwortlich und setzt als MAC-Quellenadresse die des Routers voraus. Als MAC-Zieladresse wird entweder die eines Ziel-Routers oder die einer Zielstation eingesetzt.

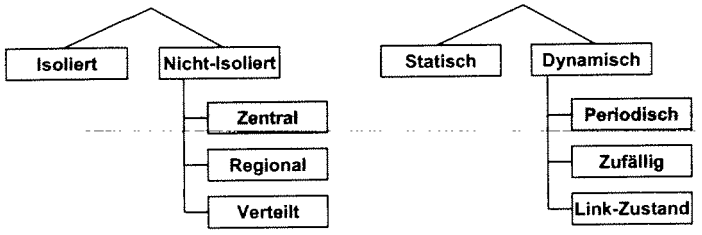
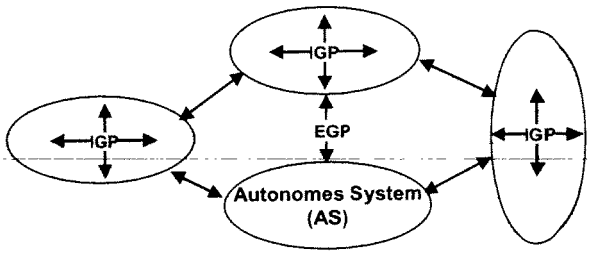


Bild: Einteilung von Routingmechanismen



IGP: Interior Gateway Protocol
EGP: Exterior Gateway Protocol

Bild: Aufteilung in Autonome Systeme (AS)

Das Routing-Verfahren lässt sich weiterhin in zwei Kategorien unterteilen:

- **Statisches Routing:** Routing-Tabellen sind in den jeweiligen Routern fest vorgegeben.
- **Dynamisches Routing:** Routing-Protokolle legen den optimalen Weg dynamisch fest.

Das statische Routing besitzt Vorteile durch die relativ geringe Komplexität. Allerdings lohnt sich der Einsatz auch nur dann, wenn die Netzkonfiguration sich langfristig nicht ändert und sich das zu erwartende Lastaufkommen in überschaubaren Grenzen hält. Wenn diese Voraussetzungen nicht gegeben sind, sollte dynamisches Routing implementiert werden. Das dynamische Routing heißt, dass bei laufendem Netzbetrieb Änderungen vorgenommen werden können und je nach Netzerweiterung alternative Wegewahl an die jeweilige Lastsituation angepasst werden kann. Das Routing-Protokoll entscheidet alleine über den optimalen Weg. Unterschiedliche Parameter bzw. Metriken (Schwellwerte) können dabei verwendet werden. Anzahl der Hops von der Quelle bis zum Ziel sowie die Kosten, Kapazität und Fehlerrate der Leitung können dabei eine Rolle spielen.

Um diese Flexibilität zu erreichen, müssen Router untereinander Informationen austauschen. Dabei werden die Routing-Tabellen initialisiert und aktualisiert. Die Routing-Protokolle erzeugen dabei eine zusätzliche Netzlast, die für die Übertragung von Nutzinformationen verloren geht. Wichtig für ein Routing-Verfahren ist ein gutes Konvergenzverfahren. Gemeint ist damit die Schnelligkeit, mit der nach einer Änderung wieder ein stabiler Zustand erreicht wird. Zu beachten ist dabei auch, wie stark die Konvergenzzeit in einem Netz bei steigender Anzahl der Router wächst.

Unterschieden werden im Grunde zwei verschiedene dynamische Routing-Verfahren:

- **Vector Distance Routing** (Distanz-Vektor-Routing),
- **Link State Routing** (Link-Status-Routing).

Vector Distance Routing

Das Distanz-Vektor-Routing Protokoll ermittelt den optimalen Weg nach der Anzahl der bis zum Ziel zurückgelegten Hops. Dabei schickt jeder Router seine gesamte Routing-Tabelle an alle anderen Router, die über direkte Verbindungen erreichbar sind. Beim Eintreffen einer Tabelle des benachbarten Routers wird die eigene mit dieser verglichen und entsprechend verändert. Anschließend sendet der Router seine neue Routing-Tabelle wieder an seinen Nachbarn weiter. Die maximale Entfernung darf 14 Hops betragen. Abgesehen von dem Nachteil schlechter Konvergenz der Routing-Information, ist dieser Algorithmus jedoch einfach zu implementieren und verbraucht wenig Speicherplatz. Die Hello-Nachricht bzw. Hello-Protokoll, welches abhängig von der Verzögerung ist, und das Routing Information Protocol (RIP, RFC-1058) sind zwei nach dem Vector Distance Algorithmus arbeitende Protokolle in IP-Netzen.

Link State Routing

Das Link-Status-Routing legt für die Berechnung der Tabellen eine vollständige Topologiebasis zugrunde. Bei Änderungen innerhalb der Tabellen werden nur die Änderungen weitergeleitet und dies auch nur an die Nachbarn der eigenen Hierarchieebene. Als Metriken können auch andere Kriterien (wie Auslastung, Durchsatz, Gebühren und Übertragungszeit) zur Berechnung des optimalen Weges herangezogen werden. Der Algorithmus besitzt daher eine schnellere Konvergenz und verursacht weniger Overhead. Ein sehr bekannter Vertreter ist das Protokoll Open Shortest Path First (OSPF). OSPF wird RIP mittelfristig ablösen, da der zugrundeliegende Algorithmus effizienter arbeitet. OSPF ist in der Version 1 nach der IETF-Spezifikation RFC-1131 definiert, aktuell ist momentan die Version 2 nach RFC-1583, während RFC-2328 bereits eine Erweiterung der Version 2 definiert. OSPF baut direkt auf IP auf und zeichnet sich neben der hierarchischen Strukturierung durch die Überbrückung größerer Entfernungen als 14 Hops sowie flexiblere Metriken aus.

Routing-Verfahren
Wichtige Routing-Verfahren sind RIP, OSPF und BGP. Die unterschiedlichen Eigenschaften führen zu unterschiedlichen Anwendungsbereichen. Ältere Verfahren werden nach und nach verbessert oder ganz abgelöst.

Routing-Algorithmus (Distanz-Vektor oder Link-State), IP-Adressen (klassenbasiert oder klassenlos), die verwendbaren Metriken (Hop Count, Verzögerung, Bandbreite, Zuverlässigkeit), die Skalierbarkeit (Einsatz in kleinen und in großen Netzen), die Konvergenzdauer (Zeitspanne von einer Topologieänderung, z. B. durch Ausfall eines Link, bis zum Erreichen korrekt aufdatierter Routing-Tabellen in allen Routern), die Nutzung alternativer Pfade (load balancing), Anforderungen an Verarbeitungs- und Übertragungsleistung für Routing-Zwecke, Sicherheit gegen böswillige Manipulation und der Aufwand für das genaue Verständnis des Verfahrens sowie für den Entwurf, die Konfiguration und die Fehlersuche.

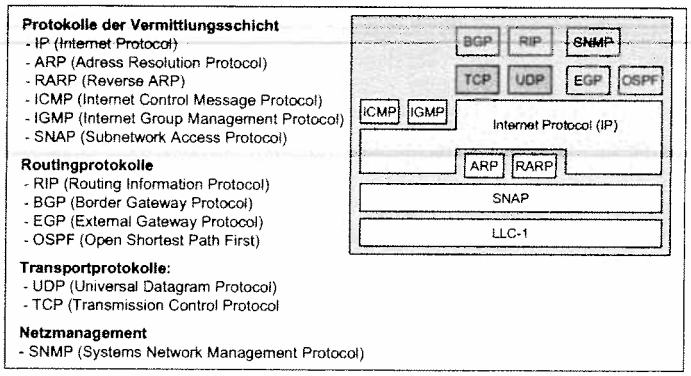


Bild: Protokolle und Routing im Internet

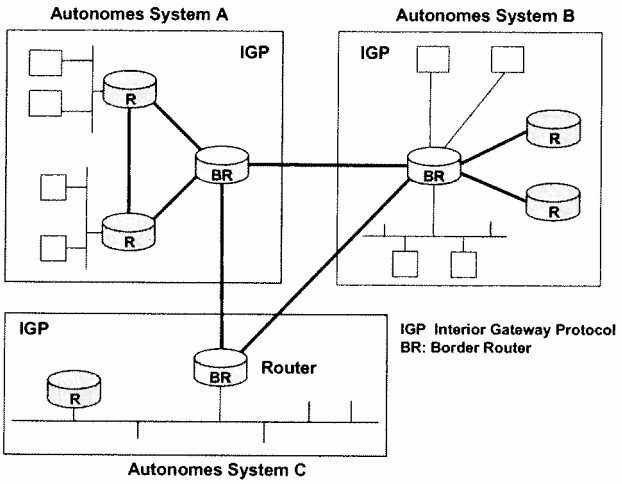


Bild: Autonome Systeme (AS)

In der IP-Welt unterscheidet man interne und externe Routing-Protokolle.

Externe Routing-Protokolle verbinden Autonome Systeme (AS), auch Routing Domains genannt. Die AS werden meist über WAN-Verbindungen gekoppelt. Die beteiligten Router der verbundenen AS kommunizieren dann mit einem externen Routing-Protokoll (Interdomain Protocol), wie EGP (Exterior Gateway Protocol) oder das modernere BGP (Border Gateway Protocol).

Interne Routing Protokolle, auch IGP-Protokolle (Interior Gateway Protocol) genannt, werden innerhalb eines AS werden verwendet. Hier unterscheidet man das statisch arbeitende Distanz-Vektor-Protokoll und das dynamisch, auf Netzveränderungen wesentlich effizienter reagierende Link-State-Protokoll.

Überblick über die Routing-Protokolle

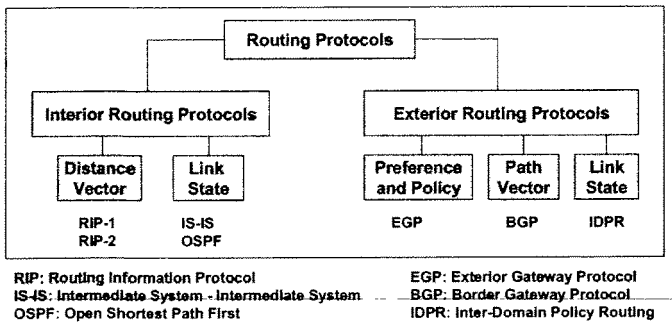


Bild: Hierarchie von Routing-Protokollen

RIP (Routing Information Protocol)
RIP (RFC 1058, RIP Version 2, RFC 1723) ist das bekannteste Distanz-Vektor-Protokoll. Es vewendet das User-Datagram Protokoll (UDP). Als Kriterium für die Wegewahl (Metrik) wird der Weg mit dem kleinsten Hopcount verwendet. Über einen im Protokoll implementierten Polling-Mechanismus wird alle 30 Sekunden ein RIP-Broadcast-Nachricht verteilt, das die Netzadresse und den Hopcount aller bekannten Netze enthält. Erhält ein Router nach 90 Sekunden kein RIP-Update, wird der entsprechende Tabelleneintrag als ungültig markiert und nach 270 s gelöscht.

Dieser Mechanismus kann in großen vermaschten Netzen zu erheblichen Kapazitätseinbußen führen. RIP-Nachrichten lassen eine maximale Nachrichtengröße von 512 Byte zu und damit sind bis zu 24 Routing-Einträge pro RIP-Nachricht möglich. Werden die Routing-Tabellen größer, müssen mehrere RIP-Nachrichten gesendet werden. Um akzeptable Konvergenzzeiten zu erreichen, wurde unter RIP die maximale Anzahl beteiligter Router in einem mit RIP organisierten Verbund auf 15 begrenzt (Hopcount = 16 bedeutet, dass das Zielnetz nicht erreichbar ist). Trotz der oben genannten Einschränkungen ist RIP auf Grund der einfachen Implementierung sehr stark verbreitet und vielfach die einzige Möglichkeit, Router verschiedener Hersteller zu koppeln.

Als Metrik wird die Größe Hopcount (die Anzahl der Hops auf einem Pfad entspricht der Anzahl der Router, die ein IP-Paket zwischen Quelle und Ziel durchläuft) verwendet. Deshalb wird immer der kürzeste Pfad gewählt, selbst wenn längere Pfade mit besseren Eigenschaften verfügbar sind. RIP kann zur Vermeidung von geschlossenen Pfaden (Pakete kommen zum Ausgangsrouter zurück und kreisen, bis ihr TTL-Wert auf null gefallen ist) während der Konvergenzdauer die folgenden Maßnahmen nutzen:

- **Split horizon:** Pfade, deren Existenz ein Router über eine bestimmte Schnittstelle gelernt hat, werden nicht über diese Schnittstelle wieder ausgesendet.
- **Hold-down timer:** Änderungen in einer Routing-Tabelle werden für eine kurze Zeitspanne eingefroren.
- **Poison reverse**: Wenn ein Pfad aus einer Routing-Tabelle verschwindet, wird dieser nicht einfach aus der Tabelle entfernt, sondern das Ziel wird als nicht erreichbar markiert.

RIPv2
Dieses Protokoll ergänzt die Routing-Tabellen durch die folgenden Felder:
- **Route tag:** unterscheidet Pfade innerhalb des jeweiligen RIP-Bereichs von Pfaden, die von außerhalb dieses Bereichs importiert wurden. Dient der Zusammenfassung von Bereichen mit unterschiedlichen Routing-Verfahren.
- **Subnet mask:** enthält die zu einer IP-Adresse gehörige Subnetzmaske. Dies ermöglicht die Verwendung klassenloser Adressen.
- **Next hop:** enthält die explizite IP-Adresse des nächsten Routers auf dem Weg zum Ziel.

IGRP (Inferior Gateway Routing Protocol)
IGRP ist ein weiteres Distanz-Vektor-Protokoll, das im Gegensatz zu RIP auch für größere Netze mit unterschiedlichen Bandbreiten und Verzögerungs-Charakteristiken geeignet ist. Als Entscheidungsparameter für die Wegewahl werden die Bitrate, Verzögerungszeit, Last, Zuverlässigkeit sowie die maximale Tranferrate ausgewertet. Diese komplexe Metrik ist z.B. notwendig, um die Router-Funktion Load-Sharing (Lastverteilung über mehrere Wege) zu unterstützen. Bei IGRP findet standardmäßig alle 90 Sekunden ein Update-Prozess statt. Dieser zeitliche Konvergenzradius ist konfigurierbar.

EIGRP (Enhanced IGRP)
Enhanced IGRP ist die Weiterentwicklung von IGRP, verwendet aber den Link-Zustands-Algorithmus. Routing-Updates erfolgen nur bei Änderung der aktuellen Netzstruktur. Die Routing-Informationen werden gezielt an die Netzteile versendet, die von der Änderung betroffen sind. EIGRP kann sowohl als Interior Gateway Protocol als auch als Exterior Gateway Protocol betrieben werden.

OSPF (Open Shortest Path First)
Ein für die IP-Welt entwickeltes Link-Zustands-Protokoll ist OSPF (Open Shortest Path First), das zu den Interior-Gateway-Protokollen gehört. Es ist ungleich mächtiger und effizienter als RIP, dafür aber auch rechenintensiver.

Zum Funktionsumfang gehören:

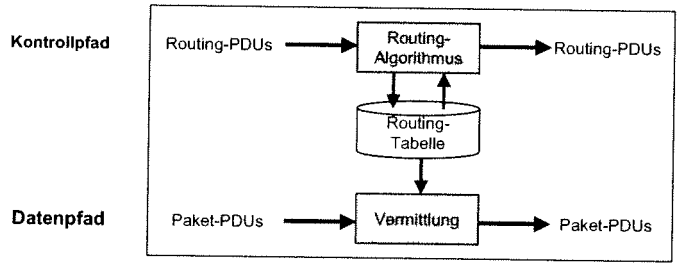
- Unterstützung hierachischer Routing-Strukturen,
- definierbare Routing-Kosten,
- ein optimierter Routing-Update-Mechanismus,
- Service-Priorisierung,
- Load Balancing,
- Unterstützung unterschiedlich langer Subnet-Masken,
- OSPF-Authentifikation.

OSPF erlaubt die Zusammenfassung mehrerer Netze innerhalb eines Autonomen Systems zu Netzbereichen (Areas). Die Netztopologie wird nicht außerhalb der Areas bekanntgegeben, sondern nur die in der Area vorhandenen Netzadressen und die in Area führenden Router. Durch Einführung der Netzbereiche haben nicht mehr alle Router eines Autonomen Systems die gleiche Topologie-Datenbank, die Router jeder Area haben eine eigene. Router, die mehreren Areas angehören, verwalten auch dementsprechend viele Datenbanken. Die Verbindung der Areas erfolgt über eine sog. OSPF-Backbone-Area, die durchgängig strukturiert ist. Das heißt, die Ziel-Area muss erreichbar sein, ohne die Backbone-Area zu verlassen. Sollte aus zwingenden Gründen die Area aus mehreren Abschnitten bestehen, so können diese über virtuelle Links verbunden werden. Die Backbone-Area besitzt immer die ID 0.0.0.0; als Area-ID wird meistens die IP-Netzadresse verwendet. Die Kopplung Autonomer Systeme im IP-Umfeld erfolgt über die oben genannten Exterior-Gateway-Protokolle (EGP, BGP).

EGP (Exterior Gateway Protocol)
EGP gehört zu den Interdomain-Routing-Protokollen und ist für den Verbund mehrerer Autonomer Systeme bestimmt. Dazu wird in jedem Autonomen System mindestens ein Router als Exterior Gateway eingerichtet, der das eine Autonome System mit dem anderen verbindet. Dadurch, dass nicht alle Router mit dem anderen Autonomen System Routing-Informationen austauschen müssen, wird der Overhead deutlich reduziert. Dieses Konzept geht davon aus, dass zwischen den Autonomen Systemen im Vergleich zur domain-internen Kommunikation der Kommunikationsbedarf gering ist. Der gesamte Datenaustausch zwischen zwei verschiedenen Autonomen Systemen läuft über einen Router. EGP baut zu den Nachbar-Exterior-Gateways eine Beziehung auf, die periodisch überwacht wird. Über diese Verbindung werden die Routing-Informationen aller bekannten Netze der jeweiligen Autonomen Systeme im Datagramm-Verfahren aktualisiert bzw. ausgetauscht. Einziges Entscheidungskriterium für die Weiterleitung der Daten ist, ob das Ziel erreichbar ist oder nicht.

BGP (Border Gateway Protocol)
BGP ist als Ersatz für EGP entwickelt worden. Es ist ein Distanz-Vektor-Protokoll, das im Unterschied zu EGP auf eine gesicherte Transportbeziehung (TCP) aufsetzt und mehrere Routing-Wege unterstützt. Jeder Router sendet periodisch seine komplette Routing-Tabelle an alle BGP-Nachbarn. Auf Basis dieser Routing-Informationen berechnet jeder BGP-Router seine bevorzugte Route zum Zielnetz.

Die Routing-Tabelle besteht prinzipiell aus folgenden Einträgen:
- die Adresse des jeweiligen Zielnetzes,
- der anzusteuernde BGP-Nachbar (Gateway),
- die Sequenz der Autonomen Systeme bis zum Zielnetz.



Kontrollpfad
- Steuert das Routen der Daten, ist aber nicht direkt im Routing-Prozess involviert
- Routingprotokolle sind oberhalb der Schicht 3 angesiedelt
- Die Aktualisierung der Routingtabelle geschieht durch den jeweils eingesetzten Algorithmus
- Routingtabelle enthält Routinginformation, die das Weiterleiten der Pakete ermöglicht
- Wegewahl beim Routing wird anhand der Routinginformation in der Routingtabelle durchgeführt

Datenpfad
Vermittlung der Pakete auf Schicht 3 (Vermittlungsschicht)

Bild: Kontroll- und Datenpfad in Routern

Routing-Konzept
Je nach Anwendungsprofil und Netzgröße sind alle oben genannten Routing-Protokolle, auch in Form von Mischvarianten, mehr oder weniger vorteilhaft einsetzbar.

Ein Routing-Konzept ist von vielen Faktoren abhängig, dazu gehören:
- Netzdesign und damit verbunden der Anspruch an Qualität und Verfügbarkeit aller Verkehrsbeziehungen in einem Gesamtnetz,
- Rechner-Systeme und Protokoll-Welten, die zu verbinden sind,
- Vorhandene Netzstrukturen (Altlasten).

Wegen des geringen Aufwands bei der Administration und der Möglichkeit, flexibel auf Routerausfälle, Verbindungsstörungen oder Adressänderungen reagieren zu können, ist das dynamische Routing dem statischen Routing vorzuziehen. Die Einschränkungen der Distanz-Vektor-Protokolle bedingen bei größeren Netzen die Konfiguration eines Link-State-

Routing-Protokolls, mit Ausnahme des proprietären IGRP (Inferior Gateway Routing Protocol), das auch bei größeren Netzen praxisbewährt im Einsatz ist. Bei Konfiguration eines Link-State-Netzes bestimmt im wesentlichen die Einteilung der Autonomen Systeme in verschiedene Areas die Reaktionszeit auf Topologieänderungen und den durch die Routing-Protokolle entstehenden Overhead.

Die Verfügbarkeit vermaschter oder ringförmiger Netzstrukturen ist gegenüber sternförmigen Netzstrukturen naturgemäß höher, aber auch kostenintensiver. Doch auch bei sternförmigen Netzstrukturen kann eine Verbesserung der Verfügbarkeit über Backup-Wege erreicht werden. Selbst Lastteilung unter Ausnutzung des Backup-Weges ist bei Konfiguration des Routing-Protokolls möglich, falls dieses Feature implementiert wurde. In vielen Fällen gibt es für einzelne Rechner an der kleineren LANs nur ein zuständiger Router, über den die Pakete an alle anderen Netze weitergeleitet werden. Hier würde weder statisches noch dynamisches Routing, sondern Default Routing sinnvoll sein. Sind zwei oder drei Router zuständig, könnte man z.B. die Default Route mit einem oder zwei statischen Routen kombinieren. Die Router in einem Kern-Netz oder Backbone-Netz sollte man grundsätzlich für dynamisches Routing konfigurieren.

Router-Management
Die meisten Router stellen für die Administration, z.B. zur Einstellung von Protokollen oder Filtern, einen interaktiven Dialog über Konsole-Ports (serielle Schnittstelle) oder auch virtuelle Terminals (Telnet) zur Verfügung. Diese Administrationsmöglichkeiten reichen bei größeren Router-Netzen nicht aus. Für den Betrieb sind zusätzliche Wartungs- und Verwaltungsfunktionen erforderlich, wie z.B. die Fähigkeit, System- und Netzdaten zu sammeln und effektiv zu verwalten.

Für große Router-Netze muss heute eine Managementplattform folgende Anforderungen erfüllen:
- Skalierbarkeit des Managementsystems, um die Netzbelastung zu verringern,
- Unterstützung mehrerer Operating-Konsolen (Multiuser-Fähigkeit),
- praktikable Filtermöglichkeiten der Managementdaten,
- automatische Fehlerkorrelation, um die Fehlerursache von den Nachfolgeerscheinungen zu trennen,
- Unterstützung von Problembehebungen durch Info-Datenbanken und automatisch gespeicherte Lösungen vorheriger Probleme.

Router-Typen
Zur Klassifizierung bzw. Unterscheidung verschiedener Router-Typen lassen sich folgende Faktoren unterscheiden:
- Funktionalität,
- Management,
- Hardware,
- Performance.

Die **Funktionalität** eines Routers hängt u.a. von den dynamisch gelernten Adressen und den statischen Einträgen ab, die in verschiedenen Tabellen geführt werden sollten. Weiterhin sollten manuell editierte Tabellen geladen werden können. Mannigfaltige Filterfunktionen sind ebenfalls zu berücksichtigen bzw. einzuplanen. Ferner ist die Anzahl der unterstützten LAN-Protokolle wichtig, falls es sich um einen Multiprotokoll-Router handelt.

Beim **Management** ist die Online-Konfiguration der Routing-Tabelle entscheidend. Dabei muss man in der Lage sein, gezielt Ports zu aktivieren bzw. deaktivieren. Kostenparameter für dynamisch gelernte Routen sollten geändert werden können. Protokollaktivitäten sollten in Logfiles dokumentiert werden.

Die **Hardware** ist für die Vielfalt der Schnittstellen wichtig. Ob man Ethernet, Token Ring, FDDI, ATM oder andere Technologien einsetzt, hängt von den Anbindungsmöglichkeiten des Routers ab. Ein modular erweiterbarer Router ist in größeren Netzen sicherlich ein Kostenvorteil. Welche Rechenkapazität für die Paketvermittlung eingesetzt wird oder ob eine Bus- oder Koppelnetz-Architektur verwendet wird sind weitere wesentliche Faktoren.

Der **Performance** als maximaler Durchsatz hängt von der Leistung der einzelnen Ports bzw. der gleichzeitigen Aktivität mehrerer Ports ab. Bei dem Einsatz von Filtern ist eine Verzögerung einzuplanen, die nicht zu groß werden darf. Auch kann der Durchsatz von den verwendeten Protokollen bzw. deren Eigenschaften abhängen. Das Routing zwischen unterschiedlichen LANs und die Umsetzung auf andere Paketgrößen ist ebenfalls für die Performance ausschlaggebend. Die Bewertung der Qualität eines Routers ist deshalb nicht so einfach möglich wie bei anderen Kopplungselementen im LAN und WAN.

Ein Router kann nur Subnetze mit identischen Netzprotokollen miteinander verbinden. Er kann also nicht Informationen von einem DECnet-Paket in ein IP-Paket übertragen. Dazu werden Gateways benötigt.

Gateway-Funktionen
Das Routen von Protokollen unterschiedlicher Netzarchitekturen ist nur durch Gateways, welche alle sieben Schichten des OSI-Referenzmodells abdecken, möglich. Gateways sind für die Kommunikation zwischen Geräten mit völlig unterschiedlichen Protokollstacks gedacht. Sie können alle Protokollschichten vollständig aufeinander abbilden, sind aber auch in der Lage nur bestimmte Schichten abzudecken (z.B. Layer-4 Firewalls).

Die vollständige Abbildung beinhaltet:

- Adressenumsetzung,
- Formatumsetzung,
- Codekonvertierung,
- Zwischenpufferung der Pakete,
- Paketbestätigung,
- Multicast,
- Unterstützung von Routing-Protokollen,
- Flusskontrolle,
- Filterung,
- Firewall (IP Security),
- Tunneling,
- Geschwindigkeitsanpassung.

Das heißt, Gateways haben die Aufgabe, Nachrichten von einem Rechnernetz in ein anderes zu übermitteln, wobei vor allem die Übersetzung der Kommunikationsprotokolle notwendig ist. Es werden allerdings nur die Funktionen umgesetzt, die von beiden Systemen unterstützt werden. Ein Gateway ist auf der kleinsten gemeinsamen Schicht der miteinander zu verbindenden Netze angesiedelt. Die Funktionalität von Gateways ist oftmals höher, als die Nutzung gemeinsamer Standardprotokolle.

Internet Routing: Paketvermittlungsfunktion (Forwarding)
Ein Router ist ein Kopplungselement, das unterschiedliche Netztypen wie LANs und WANs innerhalb der Netzschicht miteinander logisch verbinden kann. Weil in dieser Schicht Kommunikationsprotokolle angesiedelt sind, muss der Router in der Lage sein, die Zieladressen nach dem Protokoll zu interpretieren. Das macht den Router protokollabhängig. Weil die Adressen der Netzschicht eindeutig die Lage eines Endsystems im Netz bestimmen, können die Router die Übertragungswege, sogenannte Routen (Datenpfade), für die Übertragung von Paketen auswählen. Die Bestimmung von Übertragungswegen nach bestimmten Kriterien nennt man Routing. Die Realisierung der Routing-Funktion ist die wichtigste Aufgabe von Routern. Es gibt mehrere Routing-Verfahren, nach denen die Router die optimalen Routen bestimmen.

Die IPv4-Adresse gehört zu einer Klasse A, B oder C. Die auf diesen Klassen basierte IP-Adressierung stellt eine klassenweise IP-Adressierung dar. Da IP-Adressen bei derartiger Adressierung knapp geworden sind, wurde eine neue Art der IP-Adressierung eingeführt, sog. klassenlose IP-Adressierung. Diese neue IP-Adressierungsart wird bei VLSM-Networking (Variable Length Subnet Mask) und CIRD (Classless Interdomain Routing) angewandt.

Um die besten Routen zu bestimmen, müssen die Router bestimmte Informationen (sog. Routing-Informationen) miteinander austauschen. Die Regel, nach der dieser Austausch erfolgt, bezeichnet man als Routing-Protokoll.

Es sind zwei Arten der Routing-Protokolle zu unterschieden:
- **Zustandsunabhängige Routing-Protokolle:** Hier wird nur die Entfernung zum Ziel berücksichtigt. Der Zustand des Netzes (z.B. Belastung von Routern, Bandbreite von Leitungen, Zuverlässigkeit, Sicherheit) findet keine Berücksichtigung bei der Bestimmung von Routen. Das Routing-Protokoll RIP (Routing Information Protocol) ist ein zustandsunabhängiges Routing-Protokoll.
- **Zustandsabhängige Routing-Protokolle:** Hier wird der Zustand des Netzes bei der Ermittlung der Route berücksichtigt. Das Routing-Protokoll OSPF (Open Shortest Path First) ist ein zustandsabhängiges Protokoll.

Die großen Netze werden in der Regel auf Autonome Systeme aufgeteilt. Hierbei wird ein Routing-Protokoll zwischen den autonomen Systemen eingesetzt. Das BGP (Border Bateway Protocol) ist ein derartiges Routing-Protokoll. Die aktuelle BGP Version 4 (BGP-4) unterstützt das CIRD-Konzept.

Adressierung beim Router-Einsatz
Beim Internetworking ist zwischen zwei Adresstypen zu unterscheiden. Einerseits müssen die Rechner als Hardware-Komponenten adressiert werden. Dafür sind sogenannte physikalische Adressen notwendig. Die physikalischen Adressen im Schichtenmodell sind der physikalischen Schicht zuzuordnen.

- Die **MAC-Adressen in LANs** sind gerade physikalische Adressen. Ein LAN ist ein Broadcast-Netz, so dass jeder MAC-Frame in jedem Rechner empfangen werden kann, wenn die MAC-Adresse des Rechners mit der Ziel-MAC-Adresse im Frame übereinstimmt. So trifft die MAC-Adresse keine Aussage darüber, wo sich der Zielrechner befindet.
- Ein WAN ist kein Broadcast-Netz, so dass eine physikalische WAN-Adresse eindeutig die Stelle bestimmt, wo der Zielrechner angeschlossen ist. Aus diesem Grund wird die physikalische WAN-Adresse oft in Standards als **SNPA (Subnet-Point of Attachment)** bezeichnet, d.h. sie definiert eindeutig den Anschlusspunkt des Rechners am WAN.

Sowohl in LANs als auch in WANs müssen die Software-Einheiten (z.B. Anwendungen) ebenfalls adressiert werden. Dafür sind die logischen Adressen vorgesehen, die IP-Adressen darstellen. Das sind die Adressen innerhalb der Netzschicht, die als Zugangspunkte zu den Netzdiensten angesehen werden können. Die IP-Adressen sind der Grenze zwischen Schicht 3 und Schicht 4 zuzuordnen.

Wichtig ist: jeder Port eines Routers muss eine IP-Adresse haben.

Schichtenmodell für die Vernetzung mit Router

Das Schichtenmodell für die Vernetzung von IP-Subnetzen mit Router-Hilfe zeigt, dass der Quellrechner den Rahmen gezielt an den Router sendet, indem er die physikalische Router-Adresse angibt. Sind im Subnetz mehrere Router vorhanden, so muss der Quellrechner entscheiden, an welchen Router das abzusendende Paket geschickt werden soll. Dies bedeutet, dass der Quellrechner ebenso wie jeder Router über gewisse Routing-Informationen (RI) verfügen muss. Deshalb muss jeder Rechner entsprechend konfiguriert werden, um mit dem Router zusammenarbeiten zu können.

Dazu ist zu beachten, dass

- IP-Adressen angesehen werden können;
- die Protokolle der LLC-Teilschicht in den beiden LANs unterschiedlich sein können,
- Unterschiedliche LAN-Typen mit Router-Hilfe vernetzt werden können.

Übermittlung eines IP-Pakets

Bei der Adressierung der Vernetzung von LANs über WANs beim Router-Einsatz ist auf folgendes zu achten:

- Jeder Quellrechner muss entscheiden, ob ein zu sendendes IP-Paket in sein Subnetz oder über einen Router in ein anderes Subnetz übertragen werden soll. Um dies zu entscheiden, werden die Subnetz-IDs (ID: Identification) in der Ziel- und Quell-IP-Adresse miteinander verglichen. Stimmen sie überein, so wird das entsprechende IP-Paket ins eigene Subnetz geschickt und die physikalische Adresse des Zielrechners im IP-Paket eingetragen. Sind diese Subnetz-II)ls unterschiedlich, wird im abzusendenden Paket die physikalische Router-Adresse angegeben.
- Jeder Router besitzt pro Port eine physikalische Adresse. Verbindet der Port den Router mit einem WAN, stellt die physikalische Adresse eine WAN-Adresse (z.B. ISDN-Rufnummer) dar. Verbindet der Port den Router mit einem LAN, stellt die mit diesem Port verbundene physikalische Adresse eine MAC-Adresse dar.
- Jeder physikalischen Adresse des Routers wird eine IP-Adresse zugeordnet, so dass ein Router über mehrere IP-Adressen angesprochen werden kann.
- Die Ermittlung von physikalischen Router-Adressen zu den IP-Adressen erfolgt mit Hilfe des Protokolls ARP.

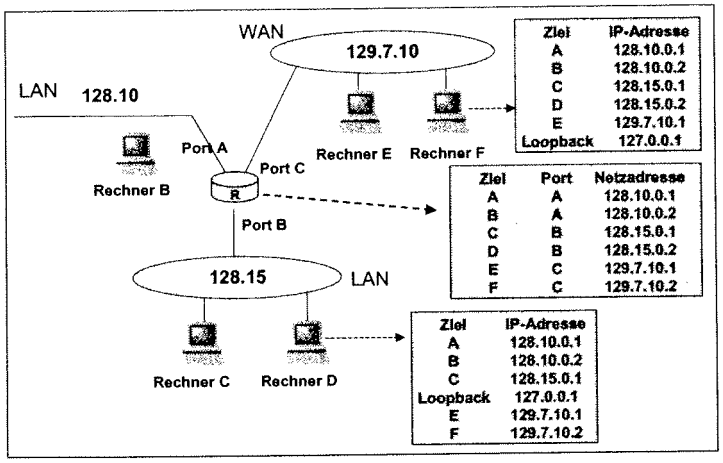


Bild: Routing- und Adresstabellen

Routing-Tabelle

Ein Router hat die Aufgabe, ein empfangenes IP-Paket auf optimale Weise in einem Verbund von mehreren Netzen weiterzuleiten. Um dies zu erreichen, enthält er eine Tabelle mit den Angaben hinsichtlich der Weiterleitung von empfangenen IP-Paketen. Diese Tabelle wird Routing-Tabelle (RT) genannt.

Ein Router muss nur die Adressen von Subnetzen interpretieren. Jedem Subnetz wird eine Zeile in der RT zugeordnet, in der angegeben wird, über welchen Port, d.h. in welches Subnetz, ein Paket weitergeleitet werden soll. Ein Paket enthält normalerweise die IP-Zieladresse, die den Ort des Rechners im Netz eindeutig bestimmt. Aus dieser IP-Adresse ist das Subnetz eindeutig erkennbar.

Eine Routingtabelle nach einem Routing-Protokoll enthält normalerweise noch zusätzliche Angaben, wie z.B. die Routen-Qualität (Kosten, Übertragungsdauer) oder die Zeitspanne seit der letzten Aktualisierung der Route.

In kleinen Netzen kann eine Routing-Tabelle manuell angegeben werden. Ist das Netz groß, wo mehrere LANs und WANs miteinander verbunden sind, werden die Routing-Tabellen in der Regel durch den Router selbst erstellt und später nach Bedarf auch selbst modifiziert. Um diese Aufgabe zu erfüllen, müssen die Router den Zustand im Netz kennen. Um dies zu erreichen, arbeiten alle Router zusammen, so dass sie sich gegenseitig helfen können. Hierbei ist jeder Router für die eigene Umgebung verantwortlich, d.h. er muss die Netzziele, die über ihn erreichbar sind, allen Nachbar-Routern mitteilen, die Nachbar-Router fassen eigene Angaben mit den Mitteilungen von anderen Nachbar-Routern zusammen und geben diese Zusammenfassung weiter. Dieser Austausch von Daten, genauer gesagt der Routing-Information (RI), führt dazu, dass alle Router nach einer gewissen Zeit den Zustand im Netz kennen. Der Austausch der RI erfolgt nach einem entsprechenden Routing-Protokoll.

Netzziel	Netz-Maske	Nächster Router	Ausgangs-Port	Metrik	Zeit
...	...	...	...	...	...

Bild: Struktur von Routing-Tabellen

Struktur von Routing-Tabellen
Als Ergebnis des RI-Austauschs liegt in jedem Router eine Routing-Tabelle vor. Jeder Eintrag (jede Zeile) in der Routing-Tabelle beschreibt eine Route.

- **Netzziel:** Das Netzziel repräsentiert das Ziel der Route. Das Netzziel kann ein Subnetz bzw. ein Rechner (Host) sein.
  - Ist das Netzziel ein Subnetz, spricht man von Subnetz-Route (auch Netz-Route genannt), d.h. diese Spalte enthält die Subnetz-ID (ID: Identification) des Ziel-Subnetzes.
  - Ist das Netzziel ein Rechner (Host), spricht man von Host-Route, d.h. diese Spalte enthält die IP-Adresse des Zielrechners.
- **Netz-Maske**: Die Subnetz-Maske wird verwendet, um die Subnetz-ID-Bits zu bestimmen. Ist das Netzziel ein Rechner, enthält diese Spalte den Wert 255.255.255.255. Bei der VLSM- bzw. beim CIRD-Einsatz kann die Länge der Subnetz-Maske in dieser Spalte angegeben werden. Ist das Netzziel ein Rechner, ist die Subnetz-Maske 32 Bits lang.
- **Nächster Router:** Diese Spalte wird auch als Gateway bzw. Nächster Hop bezeichnet. Sie enthält die IP-Adresse des nächsten Routers unterwegs zum Netzziel.
- **Ausgangs-Port:** Diese Spalte wird auch als Interface bezeichnet. Sie enthält die Angabe des Router-Ausgangs-Port, über den das zu sendende IP-Paket abgeschickt werden muss. In dieser Spalte wird in der Regel die IP-Adresse des Ausgangs-Ports angegeben.
- **Metrik:** Diese Spalte enthält die Kosten der Route. Beim Routing-Protokoll RIP wird hier die Anzahl von Hops (Anzahl von Routern zum Ziel) angegeben. Beim Routing-Protokoll OSPF kann diese Spalte auch andere Kostenart enthalten.
- **Zeitliche Begrenzung:** Falls ein Eintrag innerhalb einer Zeitbegrenzung nicht mehr aktualisiert würde, wird er gelöscht (Es handelt sich damit um eine Softstate-Information. Bei eine Hardstate-Information muss er durch einen Verbindungsabbau gelöscht werden).

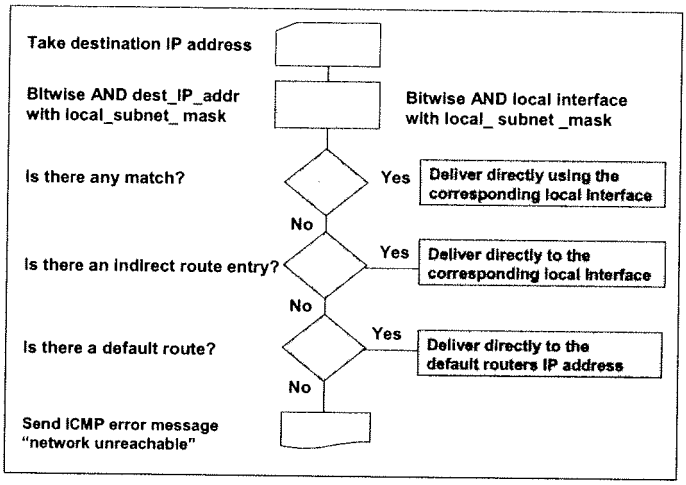


Bild: IP Routing: Forwarding Algorithmus

Bestimmung der besten Route

Um die Route zu bestimmen, wird folgender Prozess durchgeführt:

- Für jeden Eintrag in der Routing-Tabelle wird die Operation Bitweise Übereinstimmung AND zwischen der IP-Ziel-Adresse im Paket und der Netz-Maske ausgeführt und mit dem Inhalt der Spalte Netzziel verglichen. Stimmt das Ergebnis überein, gilt der entsprechende Eintrag als eine eventuelle Route.
- Die Liste von allen eventuellen Routen wird erstellt. Aus dieser Tabelle wird die Route mit der längsten Übereinstimmung ausgewählt, d.h. die Route, die den meisten Bits der IP-Zieladresse im Paket entspricht.

- Falls sich mehrere Routen mit der Übereinstimmung der gleichen Länge ergeben, wird die Route mit dem niedrigsten Wert in der Spalte Metrik als die beste Route ausgewählt.
- Falls es mehrere Routen, die als beste Routen gelten, gibt, kann der Router die geeignete Route aus den besten Routen nach dem Zufallsprinzip auswählen.

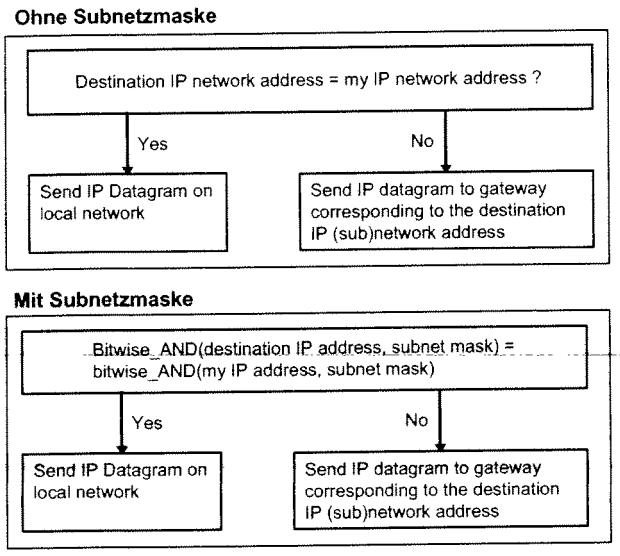


Bild: IP Routing: Forwarding Algorithmus: (Mit/ohne Subnetze)

Im allgemeinen sind folgende Arten von Routen zu unterscheiden:

- **Direkte Routen:** Routen zu den Subnetzen, die direkt erreichbar sind. Bei diesen Routen ist die Spalte Nächster Router leer.
- **Indirekte Routen:** Routen zu den Subnetzen, die über andere Router erreichbar sind. Bei diesen Routen enthält die Spalte Nächster Router die IP-Adresse des nächsten Routers, d.h. eines benachbarten Routers, an den die IP-Pakete auf dieser Route weitergeleitet werden müssen.
- **Host-Routen:** Routen zu den einzelnen Rechnern (d.h. zu den einzelnen Hosts). Bei den Host-Routen enthält die Spalte Netzziel die IP-Host-Adresse, und die Netz-Maske lautet 255.255.255.255.
- **Standard-Route:** Falls keine bessere Route zum Absenden eines Pakets in der Routing-Tabelle enthalten ist, wird die Standard-Route zum Absenden des betreffenden Pakets verwendet. Das Netzziel der Standard-Route ist 0.0.0.0 und die Netz-Maske lautet 0.0.0.0. Jede IP-Zieladresse, die mit 0.0.0.0 durch die Operation Bitweise - AND verknüpft wird, ergibt 0.0.0.0. Deshalb erzeugt der Eintrag in der Routing-Tabelle, der der Standard-Route entspricht, für jede IP-Zieladresse immer eine Übereinstimmung. Gibt es keine bessere Route, wird die Standard-Route zum Absenden des vorliegenden IP-Pakets verwendet, d.h. das IP-Paket wird an einen von vornherein festgelegten Router (sog. Standard-Router) weitergeleitet.

Nicht adaptiv
Routen ändern sich nur sehr selten
- Es besteht die Gefahr, dass bei manueller Änderung die Routenänderungen oft seltener sind als die Verkehrsänderungen.

Adaptiv
Routen ändern sich in Abhängigkeit des Verkehrs bzw. der Netztopologie
- Aktueller Zustand des Netzes wird damit berücksichtigt.
- Änderungen geschehen Periodisch oder zustandsabhängig
- Schleifen und Oszillationen in Routen sind wahrscheinlicher als bei nicht-adaptiven Verfahren.

Gefahr bei verteilten Systemen:
- Knoten haben veraltete oder unvollständige Informationen über Netzzustand.
- Es entsteht eventuell hohe Belastung durch Austausch von Routing-Information.

Bild: Dynamik von Routingverfahren

Routing-Verfahren

Die Routing-Aufgabe besteht in der Bestimmung der günstigsten Route (Pfades) für den Transport von Daten zum Zielrechner (Empfänger) durch das Netz. Die günstigste Route wird nach festgelegten Kriterien ausgewählt. Denkbare Kriterien sind z.B. Routen-Länge, -Kosten, -Qualität und Verzögerungszeit auf der Route.

Unter den Routing-Metriken sind zwei Kategorien zu unterscheiden:

- zustandsunabhängige Routing-Metriken,
- zustandsabhängige Routing-Metriken.

Ein **Routing-Verfahren** wird durch vier Hauptkomponenten bestimmt, dazu gehören die Art der Routing-Information, die Art und Weise der Routen-Bestimmung, Kriterien für die Routen-Bestimmung (Routing-Metrik) und die Strategie für die Gewinnung und den Austausch der Routing-Information.

Routing-Information (RI) sind alle Informationsarten, die dazu dienen, die Routing-Entscheidungen zu unterstützen. Eine komprimierte Form der RI stellt die Routing-Tabelle dar, die eine Basis für die Weiterleitung von Paketen bildet. Um eine endgültige Routing-Tabelle zu erstellen, ist zusätzliche RI notwendig. Hierzu gehört insbesondere die Information über die Topologie der Vernetzung, d.h. welche Subnetze vorhanden und wie sie untereinander gekoppelt sind. Um eine Route zu bestimmen, sind auch die Listen von erreichbaren IP-Adressen und von physikalischen Adressen notwendig. Eine besondere Art der RI sind die Informationen über den Zustand und die Qualität einzelner Subnetze, die für die Berechnung von sogenannten Routing-Metriken dienen.

Der **Routing-Metrik** ist ein Maß für die Qualität der Route. Eine Metrik kann sich z.B. auf Routen-Länge, Kosten, Durchsatz oder Fehlerbitrate beziehen. Zunächst muss unterschieden werden, ob es sich um ein Routing-Paket oder ein Datenpaket handelt. Ist ein empfangenes IP-Paket ein Routing-Paket, d.h. ein Paket mit Routing-Informationen (RI), so wird die in ihm enthaltene RI interpretiert und die vorhandene RI in der RI-Datenbank entsprechend modifiziert und nach dem Routing-Protokoll zu einem gegebenen Zeitpunkt an andere Router verschickt. Enthält das empfangene Paket Nutzdaten, so wird es entsprechend der Routing-Tabelle weitergeleitet.

Routing-Arten

Die Routing-Protokolle, die eine zustandsunabhängige Routing-Metrik (z.B. Routen-Länge) als Maß für die Qualität der Route bei der Routen-Auswahl verwenden, werden als zustandsunabhängige Routing-Protokolle bezeichnet. Am häufigsten wird als Routing-Metrik die Routen-Länge angenommen. In diesem Fall spricht man vom Distance Vector Routing. Die Länge der Route wird in Anzahl von Hops angegeben. Sie ist gleich der Anzahl von Subnetzen, die auf dem Weg zum Ziel liegen, wobei eine Festleitung zwischen zwei Routern auch als ein Subnetz zu zählen ist. Damit ist die Routen-Länge in Hops von einem Router zu einem Subnetz, an das der Router angeschlossen ist, gleich 1.

Das Routing-Protokoll RIP (Routing Information Protocol) verwendet die Hop-Anzahl als Routing-Metrik. Nach modernen Routing-Protokollen wird bei der Routen-Auswahl der Netz-Zustand berücksichtigt, d.h. die Routing-Metriken dieser Protokolle sind zustandsabhängig. Derartige Routing-Strategien bezeichnet man als Link State Routing (LS-Routing). Zu den LS-Protokollen gehört das Routing-Protokoll OSPF (Open Shortest Path First).

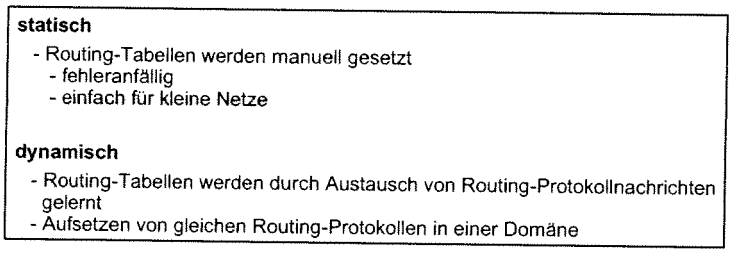


Bild: Statisches und dynamisches Routing

Ein wichtiger Aspekt bei den Routing-Protokollen ist die Bestimmung der Route. Es ist hierbei zunächst zu überlegen, wie man die Route bestimmen kann. Es sind generell zwei Fälle zu unterscheiden:

- **Statisches Routing:** Eine Route kann auf Dauer festgelegt werden. Dazu zählt man eine definierte Standard-Route (die sogenannte Default-Route), die bei der Router-Konfiguration eingegeben wird. Diese Strategie bezeichnet man als statisches Routing. In diesem Fall bedingen Änderungen und Ausfälle im Netz eine Umkonfiguration des Routers. Dies ist also sehr unflexibel und betreuungsintensiv. Somit ist statisches Routing nur sinnvoll bei kleinen Netzen und einer festen Netztopologie.
- **Dynamisches Routing, Adaptives Routing:** Wird die Routing-Information während des Betriebs im Netz ermittelt und zur Aktualisierung von Routing-Entscheidungen verwendet, so spricht man von dynamischem Routing. Die wichtigste Besonderheit des dynamischen Routings ist die Berücksichtigung des aktuellen Netzzustands in den Routing-Tabellen. Hierbei findet eine Adaption von Routen an die Lage im Netz statt. Damit ist eine derartige Routing-Strategie als adaptives Routing zu bezeichnen.

Es stellt sich die Frage: Wer bestimmt die Routen? Sie können sowohl in Endsystemen als auch in den Routern bestimmt werden. Werden die Routen in Endsystemen festgelegt, spricht man von Quellen-Routing (Source Routing). Beim Source-Routing müssen die einzelnen Pakete die vollständige Routen-Angabe aufnehmen. Die Übertragung dieser Angabe in jedes Paket vergrößert die Paketlänge enorm, so dass diese Routing-Strategie sich nicht durchsetzen konnte.

Entscheiden die einzelnen Router über die Weiterleitung von Paketen aufgrund von Routing-Tabellen, haben wir es mit verteiltem Routing zu tun. Die Routen können auch an einer zentralen Stelle im Netz errechnet und zentralisier dann an alle Router als Knoten verschickt werden. Diese Routing-Strategie ist als zentralisiertes Routing zu bezeichnen. Ein derartiges Routing findet oft in Paketvermittlungs-WANs (X.25, FR) statt.

Eine wichtige Komponente jedes Routing-Verfahrens ist die Art und Weise, wie die RI gewonnen und zwischen den Routern ausgetauscht wird. In älteren Routing-Protokollen wird die RI zwischen den Routern in festgelegten Zeitintervallen ausgetauscht. Nach den neuen Routing-Protokollen wird die RI nach Bedarf verschickt. Der Bedarf entsteht dann, wenn z.B. bestimmte Veränderungen der Route bekannt gegeben werden müssen.

Wie die RI ausgetauscht werden kann, ist ein separates Problem. Jeder Router sendet die eigene RI oft an sämtliche Nachbar-Router. Diese RI-Verteilung wird als Flooding bezeichnet. Um das Netz mit der RI nicht zu stark zu belasten, wird ein selektives Flooding realisiert. Hierbei versuchen die Router, die RI nur an einige ausgewählte Nachbar-Router zu senden. Ein Beispiel dafür wäre das Konzept Split Horizon beim Protokoll RIP. Nach diesem Konzept wird verhindert, dass die RI in die Richtung zurückgeschickt wird, aus der sie empfangen wurde.

Grundlegende Vorgehensweise

- Knoten müssen am Anfang nur ihre direkten Nachbarn kennen
- Entdecken neuer Nachbarn mit speziellen Dateneinheiten (z.B. HELLO)
- Bestimmen der Kosten zu den direkten Nachbarn

Link State Broadcast

- Identität und Kosten zu den direkten Nachbarn werden an alle Knoten im Netz durch Fluten weitergeleitet (Broadcast)
- Knoten können Topologie lernen durch die Link State Broadcasts der anderen Knoten

Ergebnis:

- Alle Knoten haben identisches Wissen über das Netz
- Berechnung der kürzesten Pfade durch Link-State-Algorithmus
- Jeder Knoten berechnet die kürzesten Pfade
- Die berechneten Pfade sind aufgrund der identischen Information gleich
- Nach Fluten und Berechnung der kürzesten Pfade in jedem Knoten ist das Netz schleifenfrei und in stabilen Zustand konvergiert

Dijkstra Algorithmus

- Berechnet den Pfad mit den geringsten Kosten von einem Knoten zu allen anderen Knoten im Netz

Bild: Link-State-Routing

Link-State-Routing

Die neuen Routing-Protokolle wie OSPF in IP-Netzen realisieren das LS-Routing (LS: Link State).

Wird ein Router an ein Subnetz angeschlossen, so muss er sich den anderen Routern im Netz vorstellen und sie auch kennenlernen. Um dies zu realisieren, werden Hello-Nachrichten ausgetauscht. Um sich den anderen Routern vorzustellen, sendet ein neuer Router im Netz immer ein Hello-Nachricht als Broadcast-Nachricht, in der er die eigene Kennung und die eigenen Adressen (physikalische Adresse und IP-Adresse) den anderen Routern bekannt gibt. Die Nachbar-Router antworten darauf ebenfalls mit entsprechenden Hello-Nachrichten, so dass der neue Router sie kennen lernen kann. Aufgrund der Hello-Nachrichten modifiziert jeder Router die RI in seiner RI-Datenbank.

Jeder Router muss seine eigene RI weitergeben. Damit konstruiert er ein entsprechendes Paket mit der RI, das die Adressen und Verbindungen zu allen Nachbar-Routern mit der Angabe der Metriken der jeweiligen Verbindungen enthält. Beim Routing-Protokoll OSPF wird eine solche Dateneinheit als LSA-Nachricht (Link State Advertisement) bezeichnet.

Die LSA-Nachrichten werden verschickt, wenn eine Veränderung auftritt, die das Routing beeinflussen kann. Jeder Router verschickt eigene LSA-Nachrichten an seine Nachbar-Router und empfängt auch deren LSA-Nachrichten. Durch den Austausch von LSA-Nachrichten kann sich jeder Router ein Bild von der Netztopologie verschaffen. Damit kann er für sich eine Routing-Tabelle erstellen, die er benötigt, um die empfangenen Datenpakete weiterzuleiten.

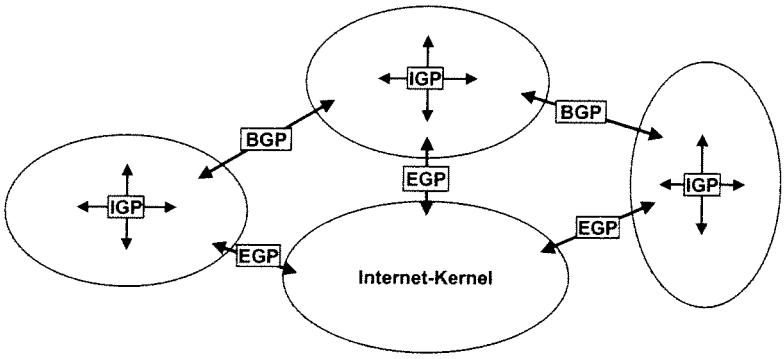
Aufteilung großer Netze in Autonome Systeme (AS) oder Regionen

- **Grund:** Anzahl der Einträge in der Routingtabelle und Menge der ausgetauschten Routinginformation sonst nicht skalierbar mit Netzgröße
- Die Router haben in einem autonomen System normalerweise nur Routing-Informationen über dieses autonome System.
- In jedem autonomen System gibt es zumindest ein ausgezeichnetes Zwischensystem, das als Schnittstelle zu anderen autonomen Systemen dient.

Vorteile

- Skalierbarkeit
- Größe der Routingtabellen ist abhängig von der Größe des autonomen Systems.
- Änderungen von Einträgen in den Routingtabellen werden nur innerhalb eines autonomen Systems weitergegeben.
- Autonomie, Internet = Netz von Netzen
- Routing kann im eigenen Netz kontrolliert werden
- Im administrativen System gib es ein einheitliches Routingprotokoll
- Routingprotokolle der autonomen Systeme müssen nicht identisch sein

Bild: Netzaufteilung in Autonome Systeme (AS)



Interior Gateway Protocol (IGP)
Enhanced IGP
Exterior Gateway Protocol (EGP)
Border Gateway Protocol (BGP)

Routing Information Protocol (RIP)
Open Shortest Path First (OSPF)

Bild: Autonome Systeme und Routing Protokolle

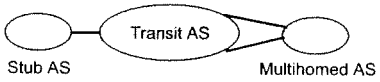
Das globale Internet besteht aus Autonomen Systemen (Autonomous Systems, AS)
- Jedes AS hat eine eindeutige Nummer
(derzeit 16 Bit, Erweiterung auf 32 Bit geplant)

Verbund von Autonomen Systemen

Stub AS
- Kleine Unternehmen
- Anschluss an genau einen Provider

Multihomed AS
- Große Unternehmen
- Anschluss an mehrere Provider
- Kein Transitverkehr

Transit AS
- Provider



Zwei Ebenen des Routing

Intra-AS
- Administrator ist verantwortlich für Wahl des Routingprotokolls

Inter-AS
- Einheitliche Standards

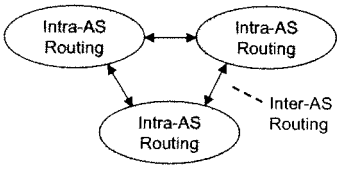


Bild: Routing und Autonome Systeme

Policy

Politische Frage: welcher Transit-Verkehr darf das AS passieren?

Inter: Policies werden vom Provider aufgestellt
Intra: Es gibt nur eine Organisation und deshalb sind wenig Policies erforderlich

Skalierbarkeit

Inter: weitere Ebene der Abstraktion ist dadurch gegeben
Tabellengrößen und Anzahl der Updates können reduziert werden, da Ausfälle innerhalb eines AS meistens verborgen bleiben können
Intra: bessere Stabilität

Leistungsfähigkeit

Inter: Policies sind erforderlich und wichtiger als Leistungs-Metriken
Intra: Konzentration auf Leistungs-Metriken

Bild: Intra- und Inter-AS-Routingprotokolle

AS, IGP und EGP

AS (Autonomes System) nutzen ein einheitliches Routing-Verfahren. Dies führt zur Einteilung IGP (Interior Gateway Protocol, zum Einsatz innerhalb eines AS, auch als Intra-domain Routing bezeichnet) und EGP (Exterior Gateway Protocol, Einsatz zum Routing zwischen verschiedenen AS, auch Inter-domain Routing)

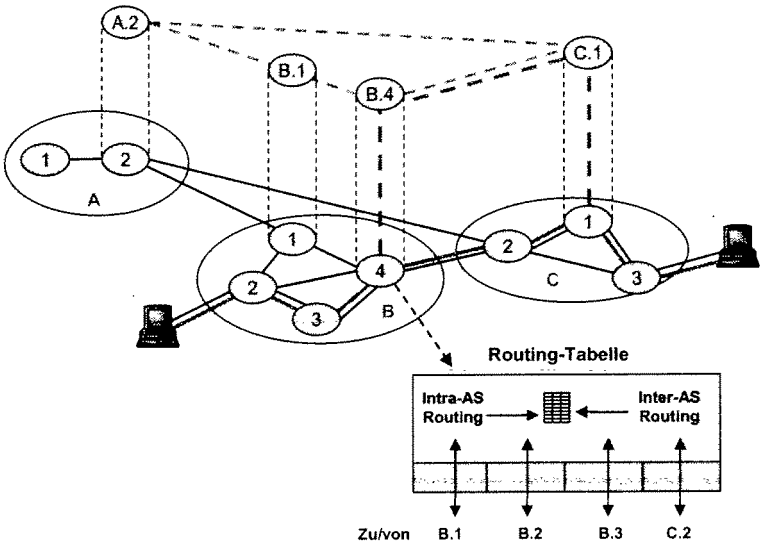


Bild: Inter-AS und Intra-AS-Routing

Intra-AS-Routingprotokolle bezeichnet man auch als Interior Gateway Protocols (IGP).

Die bekanntesten Protokolle hierfür sind

- **RIP** (Routing Information Protocol)
- **OSPF** (Open Shortest Path First)
- **IS-IS** (Intra-Domain Intermediate System to Intermediate System Routing Protocol) ursprünglich ISO/OSI-Routingprotokoll, für IP eingesetzt bei großen Providern
- **EIGRP** (Enhanced Interior Gateway Routing Protocol) CISCO proprietär

Bild: Intra-AS-Routing

Intra -Domain-Protokolle

Eine Besonderheit der IP-Netze ist, dass sie im allgemeinen eine mehrstufige hierarchische Struktur besitzen können. Ein IP-Netz kann aus mehreren Autonomen Systemen (AS) bestehen, die miteinander und mit dem Internet verbunden werden können. Ein AS kann einen Verbund von LANs und WANs darstellen, für den die Kontrolle über die Konfiguration, Adressierung und Namenskonventionen bei einer Verwaltungsautorität (z.B. eine Firma oder eine Universität) liegt. Die innerhalb eines autonomen Systems getroffenen Endscheidungen sollen keine Auswirkungen auf den restlichen Netzteil haben.

Um die Daten in einem Verbund von autonomen Systemen effektiv zu transportieren, müssen entsprechende Routing-Protokolle eingesetzt werden. Hierbei sind folgende Gruppen von Routing-Protokollen zu unterscheiden:

- AS-internes Routing-Protokoll IGP (Interior Gateway Protocol): IGP ist ein Name für jedes Routing-Protokoll, das in einem AS verwendet wird. Als IGP-Protokolle werden oft RIP (Routing Information Protocol) und OSPF (Open Shortest Path First) verwendet. Diese Protokolle werden auch als Intra-Domain-Protokolle bezeichnet.
- Routing-Protokolle zwischen den autonomen Systemen: Für die Realisierung des Routing zwischen den autonomen Systemen dient das Protokoll BGP (Border Gateway Protocol). Dieses Protokoll wird auch als Inter-Domain-Protokoll bezeichnet.

Klassenlose IP-Adressierung (VLSM, CIDR)

Um unterschiedlich große Netze zu unterstützen, wurde der Raum der IPv4-Adressen ursprünglich in drei Klassen aufgeteilt, d.h. in Klasse A, Klasse B und Klasse C. Die auf diesen Klassen basierende Vergabe von IP-Adressen bezeichnet man auch als klassenweise IP-Adressierung. Die Einteilung in die Klassen A, B und C ist einfach zu verstehen und zu implementieren. Nachteilig bei der klassenweisen IP-Adressierung ist jedoch, dass der IP-Adressraum nicht effizient ausgenutzt werden kann. Insbesondere ist keine dieser Adressklassen an die mittelgroßen Organisationen mit beispielsweise ca. 2000 Rechner angepasst. Einerseits lassen sich mit einer IP-Adresse der Klasse C nur 254 Rechner adressieren, was für eine Organisation mit ca. 2000 Rechnern zu klein ist. Verwendet man dagegen in diesem Fall eine IP-Adresse der Klasse B, mit der bis zu 65534 Rechner adressierbar sind, führt dies zu einer schlechten Ausnutzung des IP-Adressraums.

Hat eine Organisation in der Vergangenheit eine IP-Adresse beantragt, so wurde ihr je nach Bedarf eine IP-Adresse der Klasse A, B bzw. C zugewiesen. Dieses Konzept funktionierte auch für lange Zeit. Als das Internet allgemein genutzt wurde, explodierte der Bedarf an IP-Adressen. Es wurde schnell klar, dass der zunehmende Bedarf mit der klassenweisen IPv4-

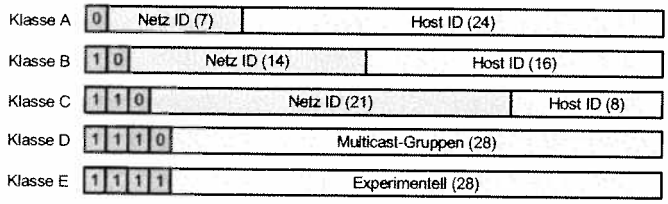
Adressierung nicht zu bewältigen ist. Es kommt noch hinzu, dass man einer Organisation mit ca. 2000 Rechnern in der Vergangenheit eine IP-Adresse der Klasse B anstatt mehrerer Adressen der Klasse C zuweisen musste. Da mit einer IP-Adresse der Klasse B bis zu 65534 Rechner adressierbar sind, hat dies zur Verschwendung von IP-Adressen geführt. Infolgedessen sind die IP-Adressen der Klasse B schnell knapp geworden.

Nachdem abzusehen war, dass die IP-Adressen bei der bisherigen klassenweisen Adressierung binnen kurzer Zeit ausgehen, wurde vor einigen Jahren eine fundamentale Veränderung eingeführt: die klassenlose IP-Adressierung (classless IP). Bei der klassenlosen IP-Adressierung kann die Grenze in einer IP-Adresse zwischen Netz-ID und Host-ID nicht nur an den Byte-Grenzen, sondern an jeder Position innerhalb der IP-Adresse liegen. Die klassenlose IP-Adressierung nutzt eine Netzpräfix-notation.

Die klassenlose IP-Adressierung ermöglicht es, Subnetze zu bilden, bei denen die Subnetz-Masken unterschiedlich lang sein können. Sie lässt somit die Subnetze innerhalb einer Netzinfrastruktur mit variabler Länge von Subnetz-Masken zu. Die Bildung von Subnetzen mit variablen Masken innerhalb von privaten Netzinfrastrukturen bezeichnet man als VLSM Networking (Variable Length Subnet Masks).

Durch den Einsatz der klassenlosen IP-Adressierung im öffentlichen Internet können einerseits die IP-Adressen effektiver ausgenutzt und andererseits die Routen zusammengefasst werden, was die Größen der Routing-Tabellen in Internet-Backbone-Routern drastisch reduziert. Der Einsatz der klassenlosen IP-Adressierung im öffentlichen Internet ist unter dem Schlagwort Classless Interdomain Routing (CIDR) bekannt.

Konzept der klassenlosen IP-Adressierung



Klasse	Anzahl Netze	Anzahl Hosts	Adressbereich
A	126 ($2^7 - 2$)	16.777.214 ($2^{24} - 2$)	1.0.0.0 – 126.0.0.0
B	16.384 (2^{14})	65.534 ($2^{16} - 2$)	128.0.0.0 – 191.255.0.0
C	2.097.152 (2^{21})	254 ($2^8 - 2$)	192.0.0.0 – 223.255.255.0
D			224.0.0.0 – 239.255.255.255
E			240.0.0.0 – 255.255.255.254

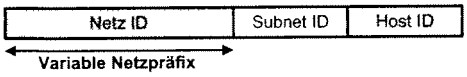
Bild: IPv4-Adressklassen

Bei jeder Klasse A, B und C von IP-Adressen wird die Grenze zwischen der Netz-ID und der Host-ID an einer anderen Stelle innerhalb der 32-Bitfolge gesetzt. Grundlegende Eigenschaft der klassenweisen IP-Adressierung ist, dass jede IP-Adresse genau angibt, welcher Teil der IP-Adresse die Netz-ID sowie die Host-ID darstellt. Die ersten Bits einer IP-Adresse, mit denen man die Adressklasse und die Netz-ID angibt, kann man als Netzpräfix (Network Prefix) bezeichnen. Bei Nutzung des Netzpräfixes zur Angabe von IP-Adressen spricht man von Netzpräfixnotation.

Wie hier ersichtlich ist, kann man die klassenweise IP-Adressierung mit Netzpräfixnotation folgendermaßen interpretieren:

- **Klasse A:** Jede IP-Adresse der Klasse A hat ein Netzpräfix von 8 Bits, bei dem das erste Bit 0 und die restlichen sieben Bits die Netz-ID angeben. Dann folgt eine Host-ID mit der Länge von 24 Bits. Nutzt man die Netzpräfixnotation bei der IP-Adresse der Klasse A, so wird sie als /8-Adresse (Achter-Adresse) bezeichnet
- **Klasse B:** Jede IP-Adresse der Klasse B hat ein Netzpräfix von 16 Bits, bei dem die ersten beiden Bits 10 sind und die restlichen 14 Bit die Netz-ID angeben. Dann folgt eine Host-ID von 16-Bit Länge. Die IP-Adressen der Klasse B werden auch als /16-Adressen bezeichnet.
- **Klasse C:** Jede IP-Adresse der Klasse C hat ein Netzpräfix von 24 Bits, bei dem die ersten drei Bits 110 sind und die restlichen 21 Bit die Netz-ID angeben. Dann folgt eine Host-ID von 8-Bit Länge. Die IP-Adressen der Klasse C werden auch als /24-Adressen bezeichnet.

Die Einteilung in die Klassen A, B und C mit ihren Beschränkungen ist einfach zu verstehen und zu implementieren. Dies ist aber für eine effiziente Belegung des IP-Adressraums nicht sinnvoll. Es fehlt einfach eine Klasse von IP-Adressen, um mittelgroße Organisationen zu unterstützen.



Ersetzen der festen Netzklassen durch Netz-Präfixe variabler Länge (13 bis 27 Bit)

Beispiel: 129.24.12.0/14
Die ersten 14 Bit der IP-Adresse werden für die Netz-Identifikation verwendet

Einsatz in Verbindung mit hierarchischem Routing
- Backbone-Router betrachtet nur z.B. die ersten 13 Bit (kleine Routing-Tabellen, wenig Rechenaufwand)
- Router eines angeschlossenen Providers z.B. die ersten 15 Bit
- Router in einem Firmennetz mit 128 Hosts betrachtet 25 Bit

Durch geschickte Adressvergabe können mehrere ursprüngliche Netze der Klasse C durch ein einziges Präfix zusammengefasst werden

Bild: Variable Netzmaske

Erweitertes Netzpräfix

Um die Subnetze innerhalb eines Netzes zu bilden, muss jedes Subnetz eine Identifikation (Subnetz-ID) erhalten. Sie wird geschaffen, indem man die Host-ID in zwei Bereiche aufteilt. Die Bildung der Subnetze bezeichnet man als Subnetting. Die traditionellen Router in IP-Netzen benutzen die Subnetz-ID der Ziel-IP-Adresse, um den Datenverkehr in eine Umgebung mit IP-Subnetzen weiterzuleiten. Die Netzpräfixnotation kann auch beim Subnetting verwendet werden. Hierbei wird das erweiterte Netzpräfix eingeführt, das sich aus dem Netzpräfix und der Subnetz-ID zusammensetzt. Die Netzpräfixnotation wird beim Classless Interdomain Routing (CIDR) verwendet.

Die Router innerhalb einer Netzumgebung mit mehreren Subnetzen benutzen das erweiterte Netzpräfix, um den Datenverkehr zwischen den Subnetzen weiterzuleiten. Bei den aktuellen Routing-Protokollen (z.B. RIP-2, OSPFv2) wird die Länge des erweiterten Netzpräfixes statt der Netzmaske verwendet. Die Präfixlänge gibt an, wie lang die ununterbrochene Anzahl Einsen der Netzmaske ist.

Die /Präfixlängen-Darstellung der Subnetz-Maske ist kompakter und leichter zu verstehen als das Ausschreiben der Netzmaske in der traditionellen Schreibweise. Durch die klassenlose IP-Adressierung ist die Grenze zwischen Netz-ID und Host-ID variabel definierbar, d.h. sie kann beliebig innerhalb 32-Bit IP-Adressen liegen und eben nicht nur an den Byte-Grenzen. Damit ist eine flexible Vergabe von IP-Adressen möglich.

Bei der klassenlosen IP-Adressierung benötigt man zwei Angaben:
- Netzadresse mit dem zugeteilten Präfix,
- Länge der Subnetz-Maske (d.h. Präfixlänge).

Die klassenlose IP-Adressierung verwendet die folgende Notation: <Netz>< Präfixlänge>.

Eine klassenlose IP-Adresse kann als Internet-Vorwahl eines Netzes angesehen werden, da sie lediglich auf das Netz verweist. Somit stellt sie nicht die IP-Adresse eines Rechners, sondern die Adresse eines Netzes bzw. eines Subnetzes dar: Eine klassenlose IP-Adresse repräsentiert einen Block von IP-Adressen, so dass sie als IP-Adressblock bzw. beim CIDR-Einsatz als CIDR-Block bezeichnet wird.

Präfixlänge in Routing-Tabellen

Die Router, insbesondere die Backbone-Router im Internet, wurden im Laufe der letzten Jahre auf die klassenlose IP-Adressierung umgestellt. Dadurch wurde es möglich, eine Route nicht zu einer einzelnen IP-Adresse (zu einem Subnetz), sondern zum IP-Adressblock (zu mehreren Subnetzen) aufzubauen. Somit werden die einzelnen Routen aggregiert, was allerdings den Einsatz eines klassenlos-fähigen Routing-Protokolls voraussetzt (z.B. BGP-4 im Backbone-Bereich bzw. RIP-2 oder OSPFv2 im lokalen Bereich).

Mit der Umstellung der Router auf die klassenlose IP-Adressierung muss eine Modifikation in den Routing-Tabellen vorgenommen werden. Bei der klassenlosen IP-Adressierung muss das Paar (Route, Präfixlänge in IP-Zieladressse) in der Routing-Tabelle im Router angegeben werden. Man unterscheidet zwischen der Route zum Subnetz und der Route zu einem gesamten Netz.

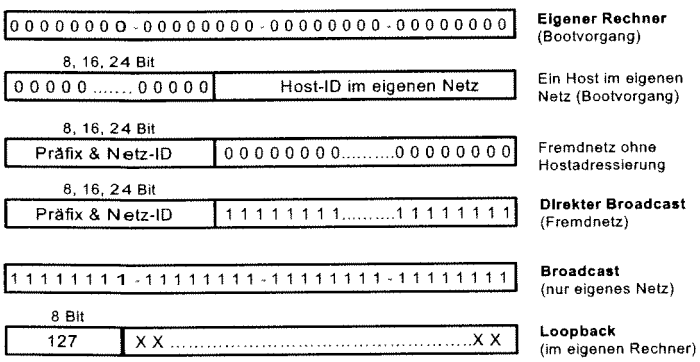


Bild: Spezielle IPv4-Adressen

Ohne Kenntnis der Netz-Maske oder der Präfixlänge kann somit ein Router nicht zwischen der Route zu dem Subnetz mit der Subnetz-ID = 0 und der Route zu dem gesamten Netz unterscheiden. Mit der Entwicklung von Routing-Protokollen, die eine Maske oder eine Präfixlänge mit jeder Route angeben, können auch die Subnetze mit Subnetz-ID = 0 eingerichtet werden.

- **RFC 950** untersagte vorher die Verwendung von Subnetz-IDs, deren Bits nur auf 0 (all-zeros subnet, d.h. Subnetz-ID, die ausschließlich Nullen enthält) und auf 1 (all-ones subnet, d.h. Subnetz-ID, die ausschließlich Einsen enthält) gesetzt werden. Die All-Zeros-Subnetze verursachen bei älteren Routing-Protokollen wie RIP Probleme, die All-Ones-Subnetze stehen in Konflikt mit der IP-Broadcastadresse.
- **RFC 1812** lässt jedoch bei der klassenlosen IP-Adressierung die All-Zeros-Subnetze und All-Ones-Subnetze zu. In Netzumgebungen, wo die klassenlose IP-Adressierung verwendet wird, müssen moderne Routing-Protokolle eingesetzt werden, die mit den All-Zeros und All-Ones-Subnetzen keine Probleme haben. Dies geschieht durch Angabe der Paare (Route, Präfixlänge) in den Routing-Tabellen.

Bemerkung: Die All-Zeros- und All-Ones-Subnetze können bei Rechnern oder Routern, die nur die klassenweise IP-Adressierung unterstützen, einige Probleme verursachen. Sollen All-Zeros- und All-Ones-Subnetze jedoch eingerichtet werden, ist zuerst sicherzustellen, dass diese Subnetze von den beteiligten Rechnern und Routern unterstützt werden.

VLSM-Nutzung

Nicht nur im Internet existiert das Problem einer derartigen Vergabe von IP-Adressen, bei der alle Adressen möglichst belegt werden. Auch innerhalb von Organisationen stellt sich die Frage, wie man den zugewiesenen Adressraum auf die einzelnen Subnetze (Teilorganisationen) effizient verteilen kann. Hierfür kann die klassenlose IP-Adressierung herangezogen werden.

Falls die klassenweise IP-Adressierung verwendet wird, können die Netze nur auf solche Subnetze aufgeteilt werden, die nach der Anzahl von adressierbaren Rechnern (Hosts) gleich groß sind. Im weiteren wird unter Subnetzgröße bzw. Subnetzlänge die maximale Anzahl von Rechnern verstanden, die man im Subnetz adressieren kann.

Beispiel

Nehmen wir an, dass mehrere Subnetze auf der Basis von IP-Adressen der klasse C gebildet werden sollen. Es wurde entschieden, vier Bits für die Subnetz-ID vom Host-ID-Teil wegzunehmen. Somit können 16 Subnetze eingerichtet werden, die (aber!) gleich groß sind. Dass eine Organisation gleich große Subnetze hat, ist ein seltener Fall.

Die Subnetze verschiedener Größe können mit Hilfe der klassenlosen IP-Adressierung eingerichtet werden. Die Bildung von Subnetzen unterschiedlicher Größe bedeutet des Einrichten von Subnetzen, bei denen die Subnetz-Masken variabler Länge, d.h. VLSM (Variable Length Subnet Masks), verwendet werden. Dies führt zu der Situation, dass in einem Netz mehrere Subnetz-Masken unterschiedlicher Länge verwendet werden. Die erweiterten Netzpräfixe haben in diesem Fall unterschiedliche Längen.

Die VLSM-Nutzung hat folgende Vorteile:

- **Rekursive Aufteilung des Adressraums:** Die Bildung von Subnetzen unterschiedlicher Größe kann in mehreren Schritten erfolgen.
- **Bessere Ausnutzung des IP-Adressraumes:** Mehrere Subnetz-Masken erlauben die bessere Ausnutzung des einer Organisation zugewiesenen IP-Adressraumes.
- **Agreggation von Routen:** Mehrere Subnetz-Masken erlauben das Zusammenfassen (Aggregieren) von Routen. Dies führt zu einer Reduktion der zu übertragenden Routing-Informnationen im Backbone-Bereich.

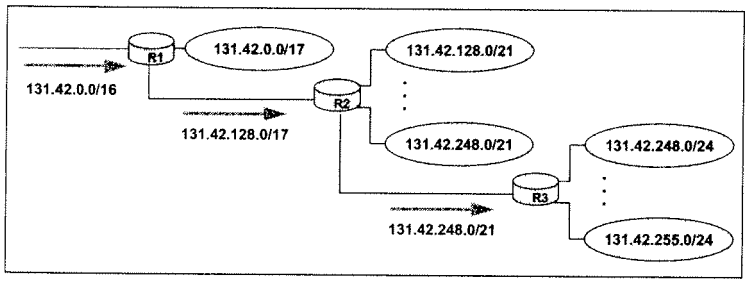


Bild. Aggregation von Routen

VLSM-Einsatz zur Strukturierung der Netze

Anhand von Beispielen soll nun der VLSM-Einsatz näher demonstrieren, wie ein Netz in mehreren Schritten strukturiert werden kann. Es ist hier zu bemerken, dass im gleichen Schritt das Netzpräfix nicht immer die gleiche Länge haben muss. Eine Aufteilung des IP-Adressraumes kann in so vielen Schritten wie notwendig durchgeführt werden.

Aggregation von Routen bei der VLSM-Nutzung

Der VLSM-Einsatz erlaubt eine rekursive Aufteilung des Adressraums einer Organisation, so dass einige Routen zusammengefasst werden können, um die Menge der zu übertragenden Routing-Information beim Austausch von Routing-Tabellen zwischen den Routern reduzieren zu können. Zuerst wird das ganze Netz in Teil-Netze geteilt. Dann werden einige Teile weiter geteilt usw. Eine derartige Strukturierung der Netze ermöglicht die Aggregation von Routen. Dies wird erreicht, indem die detaillierte Routing-Information über ein Teilnetz vor den außen liegenden Routern anderer Teilnetze verborgen wird.

Voraussetzungen für den effizienten VLSM-Einsatz

Beim VLSM-Einsatz in privaten Netzen und bei der Aggregation von Routen ist folgendes zu beachten:

- Beim Routing-Protokoll müssen die Netzpräfixe in den Routen Ankündigungen übermittelt werden.
  RIP-2 und OSPFv2 erlauben den VLSM-Einsatz, indem sie das Netzpräfix oder die entsprechende Netz- bzw. SubNetz-Maske mit der Routen-Ankündigung übertragen. Damit kann jedes Teil-Netz mit seinem Netzpräfix (oder Netz-Maske) bekannt gemacht werden.
- Alle Router müssen einen Weiterleitungsalgorithmus implementieren, der auf der längsten möglichen Übereinstimmung basiert.
  Der VLSM-Einsatz bedeutet, dass es einige Netze geben kann, deren Präfixe sich nur auf den letzten Bit-Stellen unterscheiden. Eine Route mit einem längeren Netzpräfix beschreibt ein kleineres Netz (d.h. mit weniger Rechnern) als eine Route mit einem kürzeren Netzpräfix. Eine Route mit einem längeren Netzpräfix ist daher detaillierter als eine Route mit einem kürzeren Netzpräfix. Wenn ein Router die IP-Pakete weiterleitet, muss er die detaillierteste Route (d.h. mit dem längsten Netzpräfix) benutzen.
- Adresszuweisungen müssen die Netztopologie berücksichtigen
  Um ein hierarchisches Routing zu unterstützen und die Größe von Routing-Tabellen klein zu halten, sollte man die IP-Adressen so zuweisen, dass dabei die Netztopologie berücksichtigt wird. Hierbei sollte man nach Möglichkeit einen Bereich von mehreren Adressgruppen zusammenfassen und einer Region in der Topologie zuweisen, so dass nur eine Route zu diesem Bereich führt. Falls die IP-Adressen nicht unter Berücksichtigung der Netztopologie zugewiesen werden, lässt sich die Zusammenfassung von Adressbereichen nicht erreichen und die Reduzierung der Größe von Routing-Tabellen ist nicht möglich.

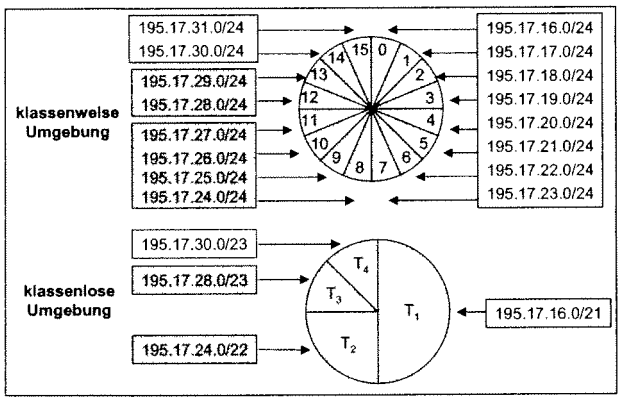


Bild. CIDR-Adresszuweisung

CIDR-Einsatz

Das rasante Wachstum des Internet hat unter den IETF-Mitgliedern ernsthafte Bedenken ausgelöst, ob das Routing-Konzept im Internet in der herkömmlichen Form mit dessen Wachstum noch Schritt halten kann. Anfang der 90er Jahre waren bereits folgende Probleme abzusehen:

- Der Klasse-B-Adressraum wird bald belegt sein.
- Die Routing-Tabellen im Internet-Backbone können unkontrolliert wachsen.
- Der 32-Bit IPv4-Adressraum wird bald zu knapp.

Um diese Probleme in den Griff zu bekommen, wurde das Konzept der Supernetze bzw. Classless Inter-Domain Routing (CIDR) entwickelt. Das CIDR-Konzept wurde offiziell im September 1993 in den RFCs 1517, 1518, 1519 und 1520 dargestellt.

Die wichtigsten CIDR-Besonderheiten sind:

- **CIDR bedeutet klassenlose IP-Adressierung**: CIDR eliminiert das traditionelle Konzept der Netzadressen der Klasse A bis C und ersetzt es durch die Netzpräfixnotation. Die Router benutzen das Netzpräfix anstelle der ersten drei Bits einer IP-Adresse, um festzustellen, welcher Teil der Adresse die Netznummer und welcher Teil die Rechnernummer ist. Damit kann der IPv4-Adressraum im Internet effizienter vergeben werden, bis das Protokoll IPv6 zur Verfügung steht.
- **Effiziente Adresszuweisung mit CIDR:** In einer klassenweisen Umgebung kann man nur /8-, /16- oder /24-Adressbereiche belegen; in einer CIDR-Umgebung hingegen können gerade die benötigten Adressbereiche belegt werden.
- **CIDR bedeutet VLSM-Einsatz im öffentlichen Internet:** CIDR erlaubt die Vergabe von IP-Adressblöcken einer beliebigen Größe, anstatt der 8-, 16- oder 24-Bit-Netznummer, die durch die Klassen vorgegeben werden. Beim CIDR-Einsatz wird mit jeder Route die Ziel-Subnetz-Maske (oder die Länge des Präfixes) bei der Verteilung der Routing-Information durch die Router angegeben. Mit der Präfixlänge wird angezeigt, wie viele Bits der Netzteil der Adresse umfasst. Eine Adresse, die beispielweise 20-Bit Subnetz-ID und 12 Bit Host-ID hat, wird mit einer Präfixlänge von 20 (/20) bekanntgegeben. Vorteilhaft ist hierbei, dass eine /20-IP-Adresse eine Adresse der Klasse A, B oder C sein kann. Die Router, die CIDR unterstützen, interpretieren nicht die ersten drei Bit der IP-Adresse, sondern benutzen ausschließlich das zusammen mit der Route empfangene Längenpräfix.
- **Aggregation von Routen mit CIDR:** CIDR unterstützt die Aggregation von Routen, so dass mehrere Routen als ein einziger Eintrag in der Routing-Tabelle repräsentiert werden können. Damit kann durch einen einzigen Routing-Eintrag der Verkehr zu vielen verschiedenen Subnetzen angegeben werden. Durch die Aggregation von Routen kann die Menge an Routen in großen IP-Netzen stark reduziert werden.

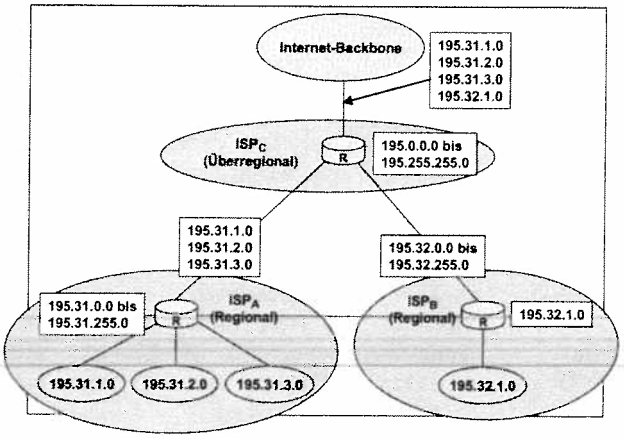


Bild: Klassenbasierte IP-Adressierung

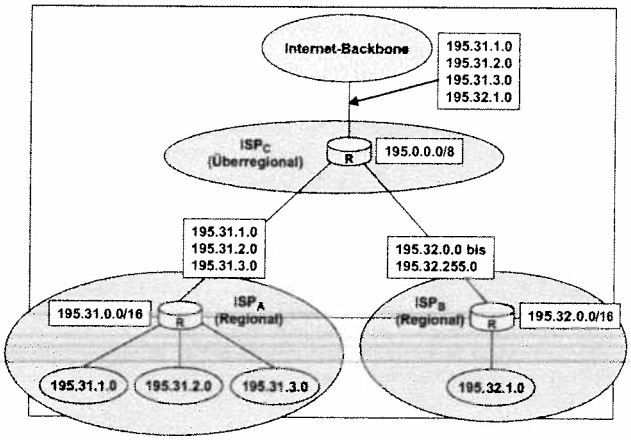


Bild: Klassenlose IP-Adressierung

Aggregation von Routen mit CIDR

Ein anderer wichtiger CIDR-Vorteil besteht in der Möglichkeit, das unkontrollierte Wachstum von Routing-Tabellen im Internet-Backbone zu verhindern.

Um die Menge der übertragenen Routing-Information zu reduzieren, wird das gesamte Internet in Routing-Domains aufgeteilt. Eine Domain repräsentiert ein entsprechend strukturiertes Netz, das von außen nur mit Hilfe ihres Netzpräfixes identifiziert wird. Innerhalb einer Routing-Domain können die interne Subnetze beliebig vernetzt werden. Die Routing Information über die einzelnen internen Subnetze aus der Domain werden nach außen unsichtbar gemacht. Zur gesamten Routing-Domain führt von außen nur eine aggregierte Route. Über eine aggregierte Route können viele Netze innerhalb einer Routing-Domain erreicht werden. Dies bedeutet, dass ein Weg zu vielen Netzadressen innerhalb der Routing-Domain mit Hilfe eines einzigen Eintrags in der Routing-Tabelle eines außerhalb liegenden Routers möglich ist.

Voraussetzungen für den effizienten CIDR-Einsatz

CIDR und VLSM sind im Grunde genommen die gleichen Konzepte. Sie ermöglichen, dass ein IP-Adressraum in kleinere Teile bedarfsgerecht aufgeteilt werden kann. VLSM unterscheidet sich von CIDR dadurch, dass die Aufteilung nach VLSM nur in dem einer Organisation zugeteilten Adressbereich erfolgt und damit für das öffentliche Internet nicht sichtbar ist. Beim CIDR dagegen kann die flexible Aufteilung eines Adressblocks von der Internet-Registrierung über einen großen ISP, von dort über einen mittleren und kleinen ISP bis zum Netz einer privaten Organisation erfolgen.

Der CIDR-Einsatz im öffentlichen Internet hat die gleichen Vorteile wie der VLSM-Einsatz innerhalb von privaten Netz-Infrastrukturen.

Ebenfalls wie beim VLSM setzt die erfolgreiche CIDR-Anwendung folgendes voraus:

- Beim Routing-Protokoll müssen die Netzpräfixe zusammen mit den Routen-Ankündigungen übermittelt werden. Routing-Protokolle wie RIP-2 und OSPFv2 erlauben den CIDR-Einsatz, indem sie das Netzpräfix bzw. die entsprechende Netzmaske mit den Routen-Ankündigungen übertragen.
- Alle Router müssen einen Weiterleitungsalgorithmus implementieren, der auf der längsten möglichen Übereinstimmung basiert.
- CIDR-Adresszuweisungen müssen die Netztopologie berücksichtigen.

Um ein hierarchisches Routing zu unterstützen und die Größe der Routing Tabellen möglich klein zu halten, sollte man die IP-Adressen so zuweisen, dass dabei die Internet-Topologie berücksichtigt wird. Hierbei soll ein Bereich von mehreren Adressgruppen so zusammengefasst und einer Region (Internet-Routing-Domain) in der Topologie zugewiesen werden, dass eine einzige Route zu diesem Bereich führt (Aggregation von Routen).

Im praktischen Einsatz: verteilte adaptive Routingalgorithmen
Dabei werden zwei grundlegende Algorithmen unterschieden

Distanz-Vektor-Algorithmen

- Distanz ist Routing-Metrik
- jedes System kennt die Distanz zu allen anderen Systemen im Netz
  hierzu werden die aktuellen Distanzen zwischen den Nachbarn ausgetauscht

Problem
– kürzerer langsamerer Weg wird längerem schnelleren Weg vorgezogen

Beispiele: Routing Information Protocol (RIP), Distance-Vector-Routing-Protocol (DVRP)

Link-State-Algorithmen

- Unterschiedliche Routing-Metriken möglich
- berücksichtigt die aktuellen Zustände der Netzanschlüsse
- jeder Router kennt die komplette Netztopologie und berechnet seine Routinginformation
- Link-State-Algorithmen konvergieren meistens schneller als Distanz-Vektor-Algorithmen
- für größere Netze sind sie potenziell besser geeignet

Beispiele: Open Shortest Path First (OSPF)
Intra-Domain Intermediate System to Intermediate System Routing Protocol (IS-IS)

Bild: Routing-Protokolle

Routing-Algorithmen

Die beiden Verfahren Distanz-Vektor-Routing und Link-State-Routing sind in der Praxis sehr wichtig.

- Beim Distanz-Vektor-Verfahren wird ein kürzester Weg gewählt, der die kleinstmögliche Anzahl von Zwischensystemen (hops) enthält.
- Beim Link-State-Verfahren ist jeder Teilstrecke (link) ein Gewicht nach einer festgelegten Metrik (z. B. Kosten, Distanz, Bandbreite und Auslastung) zugeordnet. Ein kürzester Weg ist dadurch gekennzeichnet, dass die Summe der Gewichte minimal ist.

Distanz-Vektor-Algorithmus

Der Distanz-Vektor-Algorithmus (auch als Bellman-Ford-Algorithmus bekannt) wird für die verteilte Berechnung von Routing-Tabellen verwendet. Dabei berechnet zunächst jeder Knoten für sich eine Routing-Tabelle. Die Einträge sind Tupel, die - wie der Name des Verfahrens andeuten soll - je eine Adresse (Vektor) und die zugehörige Distanz enthalten. Die Tupel werden den benachbarten Knoten mitgeteilt und zur Aktualisierung (update) deren Routing-Tabellen genutzt. Nach einiger Zeit (Konvergenzdauer) besitzen alle Knoten optimale Routing-Tabellen. Das Distanz-Vektor-Verfahren wird periodisch im Abstand weniger Sekunden durchgeführt. Damit kann der Ausfall einzelner Knoten oder Kanten (Übertragungsstrecken) berücksichtigt werden. Ein Knoten, der keine periodische Routing-Information liefert, gilt als ausgefallen. Die umgebenden Knoten ändern daraufhin ihre Routing-Tabellen so, dass der ausgefallene Knoten umgangen wird, soweit bestehende Pfade dies zulassen.

Beim Bellman-Ford-Algorithmus ist - wie beim Dijkstra-Algorithmus jede Kante des Graphen mit einem Gewicht belegt, die Distanz zum Ziel ist als Summe der Gewichte auf dem Weg zum Ziel definiert. Die Routing-Tabelle enthält für jeden Eintrag ein Feld, das die Distanz zum Zielknoten (auf einem Pfad entsprechend dem angegebenen Next Hop) enthält. Jeder Knoten sendet die ihm bekannten Wertepaare (Ziel, Distanz) an seine Nachbarn. Wenn ein Knoten eine Nachricht von seinem Nachbarn N erhält, prüft er alle Einträge und ändert seine Routing-Tabelle, falls der Nachbar einen kürzeren Pfad zu einem Ziel kennt.

Ein Vorteil des Distanz-Vektor-Algorithmus ist seine Einfachheit. Der größte Nachteil liegt in der langen Konvergenzdauer. Dies führt dazu, dass bei raschen Änderungen der Topologie Inkonsistenzen in den Routing-Tabellen entstehen, die zu großen Verzögerungen und zu Paketverlusten führen können.

Link-State-Routing, Dijkstra-Algorithmus

Das Link-State-Routing (auch Link-Status-Routing) wird in der Literatur auch als SPF-Routing (Shortest Path First) bezeichnet, obwohl andere Routing-Verfahren ebenfalls kürzeste Pfade ermitteln. Das Verfahren beinhaltet - wie das Distanz-Vektor-Verfahren - eine verteilte Berechnung des Routing. Die zwischen den Knoten ausgetauschten Nachrichten beinhalten den Status einer Verbindung zwischen zwei Knoten, der durch ein Gewicht in einer bestimmten Metrik ausgedrückt wird. Diese Nachrichten werden per Broadcast an alle Knoten gesendet. Damit besitzt jeder Knoten die globale und vollständige Zustandsinformation über das Netz. Jeder Knoten kann nun für sich einen Graphen für das Netz erstellen. Anschließend berechnet jeder Knoten seine Routing-Tabelle mit Hilfe des Dijkstra-Algorithmus. Das Routing kann also - wie beim Distanz-Vektor-Verfahren - an den aktuellen Zustand des Netzes adaptiert werden. Die Adaption findet schneller statt, da alle Knoten gleichzeitig über Statusänderungen informiert werden.

Die Grundidee des Link-State-Routing geht davon aus, dass zunächst die Topologie des Netzes ermittelt wird. Dazu sind die folgenden Schritte notwendig:

- Jeder Router kümmert sich selbst darum, seine Nachbarn und ihre Namen kennen zu lernen.
- Jeder Router bildet ein LSP (Link State Packet) mit den Namen seiner Nachbarn und den Gewichten der zugehörigen Links.
- Die LSP werden an alle Router verschickt, jeder Router speichert die zuletzt erhaltenen LSP aller anderen Router.
- Damit kennt jeder Router die vollständige Topologie des Netzes. Dies ermöglicht den einzelnen Routern die Berechnung von Pfaden zu jedem Ziel.

Der **Dijkstra-Algorithmus** wird nun zur Berechnung der kürzesten Wege ausgeführt. Der Algorithmus geht von einem bestimmten Knoten, dem Quellenknoten, aus und berechnet eine Routing-Tabelle für diesen Knoten. In der Routing-Tabelle sind für alle möglichen Zielknoten die nächsten Knoten, die in Richtung auf den Zielknoten zu durchlaufen sind, und die Distanz D von jedem Knoten zum Quellenknoten enthalten. Für jeden Knoten im Netz ist eine Routing-Tabelle zu ermitteln.

Für den Dijkstra-Algorithmus sind nebst der Beschreibung des Graphen einige Datenstrukturen erforderlich. Die Knoten werden von 1 bis n nummeriert, damit kann die Knotennummer als Index zum Datenzugriff verwendet werden. D ist ein Vektor, dessen i-te Komponente den aktuellen Wert der kürzesten Distanz vom Quellenknoten zum Knoten i enthält. Die i-te Komponente des Vektors R speichert den nächsten Knoten (next hop), der auf dem Weg zum Knoten i zu durchlaufen ist. Die Menge S der noch zu untersuchenden Knoten kann als doppelt verkettete Liste von Knotennummern gespeichert werden.

Der Dijkstra-Algorithmus lässt verschiedene Metriken zu. Im einfachsten Fall ist die Distanz gleich der Anzahl der durchlaufenen Zwischensysteme. Dazu werden alle Gewichte auf den Wert 1 gesetzt. In WANs kann die Bandbreite eines Link als Gewicht sinnvoll sein. Gewichte können auch die Ansicht des Netzadministrators (administrative policy) zu bevorzugten Pfaden widerspiegeln.

Pfad-Vektor-Algorithmus

Das Pfad-Vektor-Verfahren ist dem Distanz-Vektor-Verfahren ähnlich. Statt der Distanz zum Ziel wird jedoch ein Pfad zum Ziel angegeben. Dieser enthält eine Sequenz der zu durchlaufenden AS.

BGP (Border Gateway Protocol) ist ein Pfad-Vektor-Protokoll, das im Internet verwendet wird. Es verwendet keine Metriken. Durch die vordefinierten Pfade wird ein Policy-based Routing realisiert, das es dem Netzbetreiber erlaubt, bestimmte Pfade auszuschließen oder uninteressant zu machen.

OSPF (Open Shortest Path First Protocol, Version 2, RFC 2328) dient zum Austausch von Routing-Infonnationen zwischen IP-Routern, die zu einem AS gehören. OSPF nutzt einen Link-State-Algorithmus und setzt auf IP auf. Es soll das ältere RIP ersetzen. OSPF strukturiert ein Netz hierarchisch in Areas (Zusammenfassung mehrerer Subnetze), Backbones (Zusammenfassung mehrerer Areas, ein Backbone bildet selbst eine Area) und AS (Zusammenfassung mehrerer Backbones). Die Router haben demzufolge verschiedene Aufgaben und Bezeichnungen: interne Router (innerhalb einer Area oder eines Backbone), designierte Router (ein ausgewählter Router in einer Area, der stellvertretend für alle anderen Routing-Informationen mit anderen Areas austauscht), Area Border Router (verbindet zwei Areas oder eine Area mit einem Backbone) und AS Boundary Router (stellt die Verbindung zu anderen AS her). OSPF konvergiert schneller als RIP und benötigt weniger Bandbreite für die Übertragung von Routing-Informationen.

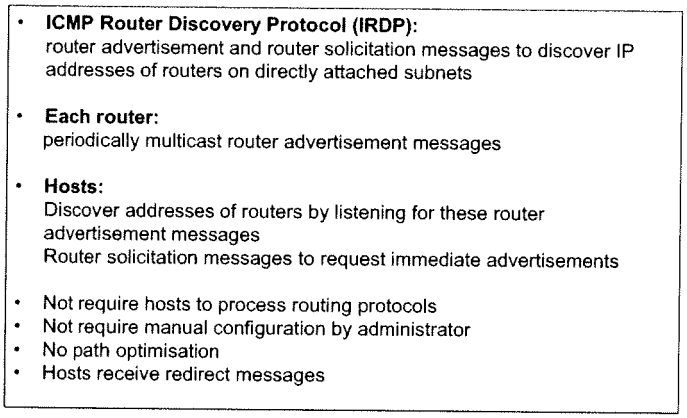


Bild: ICMP Router Discovery Protocol

Routing Information Protocol (RIP)

Das Routing-Protokoll RIP (Routing Information Protocol) ist ein sog. Distanzvektor-Routing-Protokoll. Dies bedeutet, dass die Entfernung zum Ziel in der Anzahl von Hops angegeben wird. Das Wort Distanzvektor verweist darauf, dass die Routing-Information zwischen den Routern in Form von Distanzvektoren ausgetauscht wird.

Zwischen dem RIP für das Protokoll IP (kurz RIP für IP) und dem RIP für das Protokoll IPX (Internetwork Packet eXchange) muss unterschieden werden. Im weiteren wird nur das Protokoll RIP für IP betrachtet und kurz als RIP bezeichnet. Die Ursprünge von RIP liegen in der XNS-Version (Xerox Network Services). Inzwischen ist das RIP ein weit verbreitetes Routing-Protokoll geworden. Neben dem Protokoll OSPF (Open Shortest Path First) ist das RIP das wichtigste Routing-Protokoll in IP-Netzen.

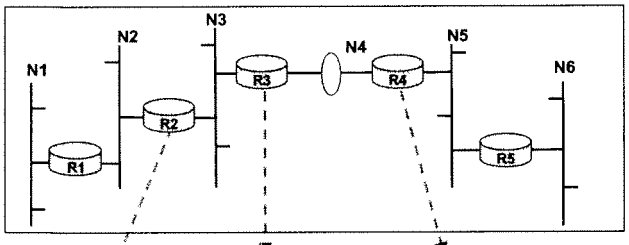
Es existieren zwei Versionen von RIP,

- RIP-Version 1 (RIP-1, RFC 1058),
- RIP-Version 2 (RIP-2, RFC 2453).

RIP ist zwar einfach und wird oft eingesetzt, doch hat es einige Schwächen, die an seinem ursprünglichen LAN-orientierten Konzept liegen. Der RIP-Einsatz in standortübergreifenden IP-Netzen über WANs ist mit großen Problemen verbunden. Daher eignet sich RIP hauptsächlich für kleine bis mittelgroße IP-Netze.

RIP verwendet die Anzahl von Hops als Entfernungsmaß (Metrik) für die in der Routing-Tabelle gespeicherte Routen. Dabei ist ein Hop als Sprung von einem Router ins Subnetz zu interpretieren. Ist die Anzahl von Hops zwischen dem Router A und einem Netzziel gleich x, so ist die Anzahl der Router auf dieser Strecke gleich x-1. Die Anzahl von Hops stellt daher die Anzahl der Router dar, die bis zum Erreichen des gewünschten Netzes unterwegs durchquert werden müssen.

Beim RIP wird die maximale Anzahl von Hops (Hoplimit) auf 15 begrenzt. Dies bedeutet, dass höchstens 15 Router zwischen Quelle und Ziel eines IP-Pakets liegen können. Ein Zielrechner in IP-Netzen, der um 16 oder mehr Hops vom Quellrechner entfernt ist, gilt als nicht erreichbar.



Router R2
Distanz-Vektor Tabelle

Netz	Next Hop	Metrik
N1	R1	2
N2	=	1
N3	=	1
N4	R3	2
N5	R3	3
N6	R3	4

Router R3
Distanz-Vektor Tabelle

Netz	Next Hop	Metrik
N1	R2	3
N2	R2	2
N3	=	1
N4	=	1
N5	R4	2
N6	R4	3

Router R4
Distanz-Vektor Tabelle

Netz	Next Hop	Metrik
N1	R3	4
N2	R3	3
N3	R3	2
N4	=	1
N5	=	1
N6	R5	2

Bild: Distanz-Vektor Routing

Jeder RIP-Router macht den Inhalt seiner Routing-Tabelle in allen an ihn angeschlossenen Subnetzen alle 30 Sekunden bekannt. Der Inhalt der Routing-Tabelle wird beim RIP-1 als Broadcast auf MAC-Ebene und beim RIP-2 als Multicast im Subnetz gesendet. Dies kann besonders beim Einsatz von WAN-Verbindungen zu Problemen führen, wo erhebliche Anteile der WAN-Übertragungskapazität zur Weiterleitung von RIP-Nachrichten verwendet werden müssen. Somit lässt sich RIP-basiertes Routing nicht leicht in großen IP-Netzen mit WAN-Anteilen einsetzen.

Die Hop-Anzahl bildet das einzige Kriterium zur Ermittlung der besten Route, d.h. je weniger Hops zum Netzziel in einer Route vorhanden sind, desto besser ist diese Route. Die Fähigkeiten von RIP sind sehr beschränkt, so gehen auch Leitungskapazität und Kosten nicht in die Berechnung der Route ein. Weitere Nachteile von RIP sind, dass lediglich eine aktive Route zwischen zwei Netzen genutzt werden kann und Aktualisierungen von Routing-Tabellen bei lokalen Topologie-Änderungen mittels Broadcast im gesamten Netz verteilt werden und damit das Netz unnötig belasten.

Der wichtige Vorteil von RIP ist die große Verfügbarkeit, denn fast jeder Rechner ist in der Lage, RIP zu verarbeiten. Einem Systemadministrator, der ein großes und standortübergreifendes Netz mit WAN-Anteilen zu verwalten hat, sind leistungsfähigere Protokolle, wie OSPF, zu empfehlen.

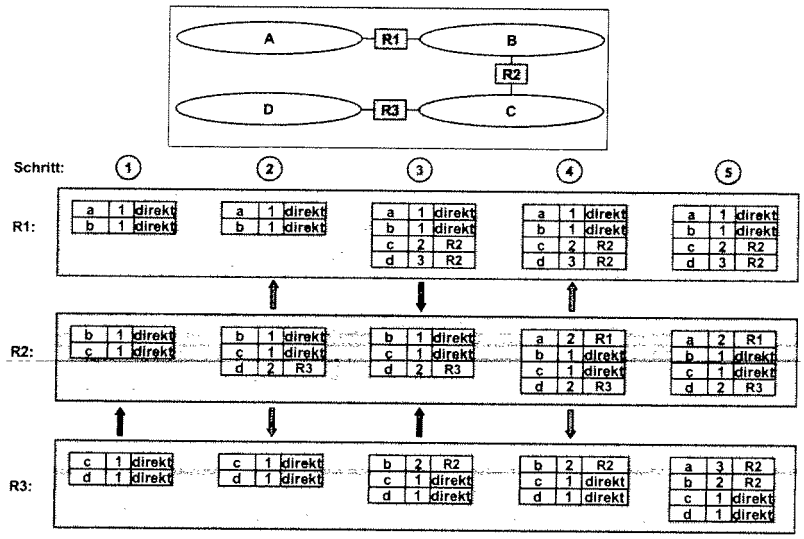


Bild: Routing Information Protocol (RIP)

Erlernen von Routing-Tabellen beim RIP

Die einzelnen RIP-Router werden in keiner Weise miteinander synchronisiert. Jeder Router versendet den Inhalt seiner Routing-Tabelle in alle an ihn angeschlossenen Subnetze. Empfängt ein Router eine RIP-Nachricht, so modifiziert er seine Routing-Tabelle.

Reduzierung der Konvergenzzeit

Es stellt sich nun die Frage, wie viel Zeit ein Router benötigt, um in seiner Routing-Tabelle die aktuellsten Routen-Angaben zu allen Subnetzen zu erlernen. Dafür muss die Routing-Tabelle oft mehrfach modifiziert werden.

Somit sind mehrere Modifikationsschritte notwendig und die Anzahl dieser Schritte bestimmt die Konvergenzzeit. Der Grenzwert 15 als maximale Hop-Zahl wurde u.a. eingeführt, um die Konvergenzzeit in sinnvollen Grenzen zu halten. Wie bereits erwähnt wurde, wird die Routing-Information in Abständen von 30 Sekunden verschickt. Wäre die maximale Hop-Anzahl nicht auf 15 eingeschränkt, könnte die Konvergenzzeit zu lange dauern. In einer zu langen Periode kann es vorkommen, dass einige Routen nicht mehr aktuell sind.

Um die Konvergenzzeit zu verringern, können beim RIP folgende Methoden verwendet werden:

- Split-Horizon-Methode (geteilter Horizont),
- Split-Horizon-Methode mit Poison-Reverse (geteilter Horzont mit vergiftetem Rückweg),
- Ausgelöste Router-Aktualisierungen (triggered updates).

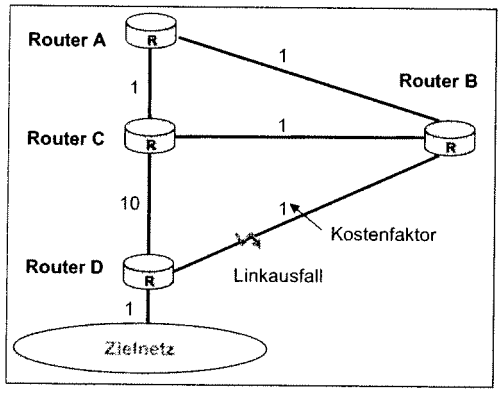


Bild: RIP: Netzbeispiel

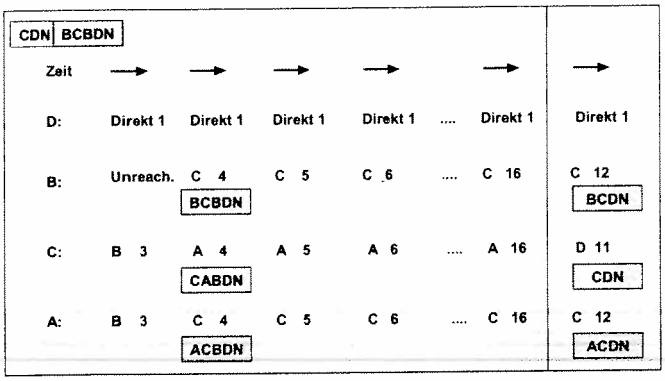


Bild: Routing Table Update: Counting to Infinity

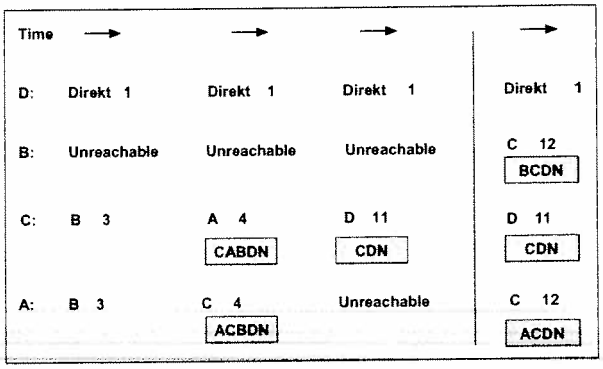


Bild: Routing Table Update: Split Horizon

Bei der **Split-Horizon-Methode** handelt es sich um die Router-Ankündigungen, die periodisch alle 30 Sekunden gesendet werden. Hierbei darf ein Router keine Subnetze auf dem Subnetz ankündigen, die er bereits aus diesem Subnetz erlernt hat. Anders ausgedrückt: Jeder Router kündigt auf einem Subnetz (d.h. über einen Port) nur diese Subnetze an, die er über andere Subnetze (d.h. über andere Ports) kennen gelernt hat. Die in RIP-Nachrichten gesendeten Angaben enthalten nur Subnetze, die sich jenseits des benachbarten Routers in entgegengesetzter Richtung befinden.

Die **Split-Horizon-Methode mit Poison-Reverse** unterscheidet sich von der einfachen Split-Horizon-Methode dadurch, dass alle Subnetze angekündigt werden. Diese Subnetze, die aus einer bestimmten Richtung erlernt wurden, werden jedoch mit der Entfernungsangabe von 16 Hops angekündigt und somit beim RIP als nicht erreichbar interpretiert. In einigen Fällen besitzt diese Methode einige Vorteile gegenüber dem einfachen Split-Horizon-Prinzip.

Durch **ausgelöste Router-Aktualisierungen** kann ein RIP-Router die Änderungen in seiner Routig-Tabelle beinahe umgehend ankündigen und muss nicht bis zur nächsten regelmäßigen Ankündigung (d.h. zum Ablauf des Timers 30 Sekunden) warten. Als Auslöser der Aktualisierung kann eine Metrik-Änderung in einem Eintrag der Routing-Tabelle sein. Beispielsweise lassen sich über eine ausgelöste Aktualisierung diese Subnetze, die nicht mehr verfügbar werden, mit der "Entfernung" 16 Hops ankündigen. Ausgelöste Aktualisierungen verbessern zwar die Konvergenzzeit, doch geschieht dies auf Kosten des zusätzlichen Broadcastverkehrs.

Besonderheiten von RIP-1

Das Routing-Protokoll RIP-1 (Version 1) wird in RFC 1058 spezifiziert. Da die Subnetz-Masken in RIP-1-Nachrichten nicht übermittelt werden, kann beim RIP-1-Einsatz nur eine Subnetz-Maske pro Netz verwendet werden. Beim RIP-1-Einsatz müssen daher die Subnetz-Masken innerhalb des gesamten Netzes gleich sein. Das RIP-1 wird hauptsächlich in kleinen und mittelgroßen IP-Netzen eingesetzt.

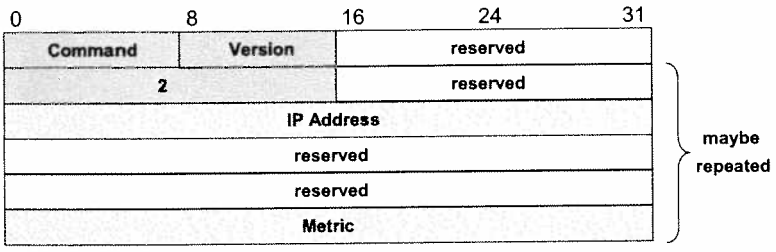


Bild: Struktur von RIP-1-Nachrichten

Struktur von RIP-1-Nachrichten

Für die Übermittlung von RIP-1-Nachrichten wird das Transportprotokoll UDP verwendet. Die RIP-1-Nachrichten werden über den UDP-Port 520 sowohl gesendet als auch empfangen. Beim Versenden einer RIP-1-Nachricht auf ein Subnetz wird die IP-Broadcastadresse im IP-Header als IP-Zieladresse genutzt. Jede RIP-1-Nachricht setzt sich aus einem Header und einer Vielzahl von Einträgen zusammen. In einem RIP-Eintrag (RIP-1 Entry) wird ein Subnetz angekündigt.

Der Header enthält die Felder:

- **Command (1 Byte).** Hier wird angegeben:
  - x'01'; es handelt sich um ein RIP-Request (Anforderung),
  - x'02'; es handelt sich um eine ein RIP-Response (Antwort).
- **Version (1 Byte).** Hier wird die RIP-Version angegeben. Beim RIP-1 enthält dieses Feld den Wert x'01'.

Die Request-Nachrichten werden bei der Initialisierung eines Routers gesendet. Beim Start kündigt der RIP-Router in allen lokal angeschlossenen Subnetzen die ihm bekannten Subnetze an. Der initialisierende Router sendet außerdem auf alle angeschlossenen Subnetze ein allgemeines RIP-Request. Dabei handelt es sich um eine besondere Nachricht, mit der alle benachbarten Router aufgefordert werden, ihm die Inhalte von ihren Routing-Tabellen in Form von Unicast-Nachrichten zukommen zu lassen. Auf der Basis dieser Antworten wird die Routing-Tabelle des initialisierenden Routers aufgebaut.

Eine Antwort kann als Reaktion auf eine Anfrage oder als regelmäßige bzw. ausgelöste Router-Ankündigung gesendet werden.

Ein RIP-1-Eintrag kann als ein 20-Byte-Behälter betrachtet werden, der folgende Angaben übermittelt:

- **Address Family Identifier (AFI, 2 Bytes):** Das RIP wurde ursprünglich für das Routing in heterogenen Netzen konzipiert, wo unterschiedliche Adressierungsarten verwendet werden. An dieser Stelle wird markiert, um welche Adressierungsart es sich handelt. Handelt es sich um die IP-Adressierung, so steht hier der Wert 2.
- **IPv4 Address (4 Bytes):** In diesem Feld wird das Netzziel angegeben. Dabei kann es sich um eine klassenlose Netzkennung, eine Subnetzkennung, eine IP-Adresse (für eine Host-Route) oder um 0.0.0.0 (für die Standard-Route) handeln. Bei einem allgemeinen RIP-Request wird als IPv4-Adresse 0.0.0.0 angegeben.
- **Metrik (4 Bytes):** Hier wird die Hop-Anzahl zum Netzziel angegeben. Dieser Wert beschreibt die Anzahl von Hops von dem Router, der diese RIP-Nachricht abgeschickt hat, die benötigt werden, um das betreffende Netzziel zu erreichen. In diesem Feld ist der zugelassene Höchstwert 16. Nach RIP sind maximal 15 Hops zwischen einem Router und einem Subnetz zulässig. Der Wert 16 hat eine besondere Bedeutung. Er weist darauf hin, dass ein betreffendes Netzziel unerreichbar für einen Router ist.

Die maximale Länge einer RIP-1-Nachricht (ohne UDP- und IP-Headers) beträgt 512 Bytes. Wenn der RIP-Router eine vollständige Liste aller Subnetze und aller möglichen Wege zu diesen Netzzielen speichert, kann die Routing-Tabelle so viele Einträge enthalten, dass sie in mehreren RIP-Nachrichten gesendet werden müssen. In einer einzigen RIP-Nachricht können nur 25 Einträge gesendet werden.

Routing-Tabelle bei RIP-1

Das Routing-Protokoll RIP-1 wurde für die Netze mit der klassenbasierten IP-Adressierung konzipiert, bei der die Netzkennung (Netz-ID) aus den Werten der ersten drei Bits der IP-Zieladresse bestimmt werden kann. In den RIP-1-Nachrichten wird die Netz-Maske (bzw. Subnetz-Maske) nicht übermittelt.

Die allgemeine Struktur der Routing-Tabelle beim RIP-1.

- Die erste Spalte enthält die Netzziele als Netz- bzw. Subnetzkennungen (Netz- bzw. Subnetz-IDs).
- Jedem physikalischen Port im Router muss eine IP-Adresse zugeordnet werden. In der zweiten Spalte Weiterleitungsadresse wird die IP-Adresse des Ausgangsports angegeben, über den das betreffende Paket abgesendet werden soll. Ist ein Router-Port ein LAN-Port (d.h. mit einer LAN-Adapterkarte), so wird ein IP-Paket über diesen Port in einem MAC-Rahmen gesendet. Auf der Basis der IP-Adresse des physikalischen LAN-Ports wird die Quellen-MAC-Adresse für den MAC-Rahmen mit dem IP-Paket bestimmt. Hier kommt das Protokoll ARP zum Einsatz.
- Die dritte Spalte enthält die IP-Adresse des nächsten Routers, falls das IP-Paket zu einem entfernten Ziel gesendet wird, bzw. die Identifikation eines lokalen Subnetzes, zu dem das IP-Paket direkt übergeben wird.
- Die vierte Spalte enthält die Metrik als Entfernung in Hops zum Zielsubnetz.
- In der letzten Spalte Timer wird die Zeitspanne seit der letzten Aktualisierung der Tabelle angegeben.

Die Bedeutung der Timer-Spalte soll nun kurz erläutert werden. Fällt ein Router aufgrund eines Stromausfalls oder eines Hardware- bzw. Softwarefehlers aus, besitzt er keine Möglichkeit, benachbarten Routern mitzuteilen, dass die über ihn erreichbaren Netzziele nicht mehr verfügbar sind. Um die Einträge mit nicht erreichbaren Zielen in Routing-Tabellen zu verhindern, besitzt jede von RIP erlernte Route standardmäßig eine maximale Lebensdauer von 3 Minuten. Wird eine Route in der Routing-Tabelle innerhalb von 3 Minuten nicht aktualisiert, so wird ihre Hop-Anzahl auf 16 gesetzt; und diese Route wird schließlich aus der Routing-Tabelle entfernt. Deshalb dauert es 3 Minuten bei Ausfall eines Routers, bis die benachbarten Router die von dem ausgefallenen Router erlernten Routen als nicht erreichbar markieren.

Schwächen von RIP-1

Das RIP-1 wurde im Jahre 1988 entwickelt, um in LAN-basierten IP-Netzen dynamisches Routing zu ermöglichen. Die LAN-Technologien wie Ethernet und Token-Ring unterstützen den Broadcast-Verkehr auf der MAC-Schicht, so dass ein einzelnes Paket von mehreren Rechnern empfangen und verarbeitet werden kann. Das RIP nutzt diese LAN-Eigenschaft.

Die wesentlichen Schwächen von RIP-1 sind:

- **Router-Ankündigungen als Broadcast auf der MAC-Ebene:** In modernen Netzen ist die Unterstützung von Broadcasts auf der MAC-Schicht nicht wünschenswert, weil sie zu großen Belastungen des Netzes führt.

- **Silent-RIP-Rechner:** Da die Router-Ankündigungen beim RIP-1 als MAC-Broadcast versendet werden, ermöglicht dies, sog. Silent-RIP-Rechner zu installieren. Ein Silent-RIP-Rechner verarbeitet RIP-Ankündigungen, kündigt jedoch seine eigenen Routen nicht an.
- **Keine CIDR-Unterstützung:** Das RIP-1 wurde zu einer Zeit entwickelt, als die IP-Netze aus schließlich Netz- und Subnetz-IDs verwendeten, die nur die klassenbasierte IP-Adressierung nutzten. Heute ist hingegen der Einsatz von CIDR (Classless Inter-Domain Routing) und Bildung der Subnetze mit Masken von variabler Länge für die bessere Ausnutzung des IP Adressraums beinahe unumgänglich.
- **Subnetzmaske wird in RIP-1-Nachrichten nicht übermittelt:** Das RIP-1 wurde für klassenbasierte IP-Netze entwickelt, in denen die Netzkennung aus den Werten der ersten drei Bits der IP-Adresse bestimmt werden kann. Da die Subnetz-Maske nicht übermittelt wird, muss der Router einfache Annahmen über die Subnetz-Masken selbst machen. Bei jeder Route, die in einer RIP-1-Nachricht enthalten ist, kann der Router bei der Bestimmung der Subnetz-Maske wie folgt vorgehen:
  - Wenn die Netzkennung zu einer Netzklasse A, B oder C passt, wird von der standardmäßigen klassenbasierten Subnetz-Maske ausgegangen.
  - Wenn die Netzkennung zu keiner Netzklasse A, B oder C passt, so kann man wie folgt vorgehen:
    - Wenn die Netzkennung zu der Subnetz-Maske der Schnittstelle passt, auf der die RIP-1-Nachricht empfangen wurde, so kann von der Subnetz-Maske dieser Schnittstelle ausgegangen werden, auf der gerade die RIP-1-Nachricht empfangen wurde.
    - Wenn die Netzkennung nicht zur Subnetz-Maske der Schnittstelle passt, auf der die RIP-1-Nachrichten empfangen werden, kann davon ausgegangen werden, dass es sich um eine Host-Route mit der Subnetz-Maske 255.255.255.255 handelt.

Routing-Protokoll RIP-2

Das Routing-Protokoll RIP Version 2 für IP (kurz RIP-2) ist ebenfalls wie RIP-1 ein Distanz-Vektor-Protokoll. Die letzte Spezifikation von RIP-2 ist in RFC 2453 dargestellt.

Ziele von RIP-2

Mit der Entwicklung des Protokolls RIP-2 wurde versucht, einige Schwächen von RIP-1 zu beheben, um folgende Ziele zu erreichen:

- das Verkehrsaufkommen durch Versenden von Router-Ankündigungen zu reduzieren;
- die Bildung von Subnetzen mit Masken variabler Länge und damit die Einsparung von IP-Adressen zu ermöglichen;
- die Router vor falsch oder böswillig konfigurierten Nachbar-Routern zu schützen;
- Abwärtskompatibilität zum RIP-1.

Die wichtigen RIP-2-Besonderheiten:

- Das Erlernen von Routing-Tabellen erfolgt beim RIP-2 nach den gleichen Prinzipien wie beim RIP-1.
- **Maximale Hop-Anzahl ist 15:** Beim RIP-2 (wie beim RIP-1) ist die maximale Anzahl von Hops auf 15 begrenzt. Die Hop-Anzahl 16 auf einer Route bedeutet, dass das Netzziel nicht erreichbar ist.
- Beim RIP-2 können die Methoden:
  - Split-Horizon,
  - Split-Horizon mit Poison-Reserve,
  - Ausgelöste Router-Aktualisierungen

  zum Vermeiden das Count-to-Infinity-Problems und auch zum Verringern der Konvergenzzeit verwendet werden.
- **Router-Ankündigungen als IP-Multicast:** Beim RIP-2 werden die Router-Ankündigungen nicht mehr als MAC-Broadcast versendet, sondern für das Versenden von Router-Ankündigungen wird die IP-Multicastadresse 224.0.0.9 im IP-Header als IP-Zieladresse gesetzt. Somit werden alle Nicht-RIP-Rechner von RIP-Router-Ankündigungen nicht beeinträchtigt.
- **Übermittlung von Subnetzmasken:** In RIP-2-Nachrichten wird die Subnetz-Maske zusammen mit dem Netzziel übermittelt. Das RIP-2 kann somit in VLSM-Umgebungen (Variable Length Subnet Masking) eingesetzt werden.
- **Authentifizierung:** Das RIPv2 ermöglicht die Authentifizierung, um den Ursprung eingehender Router-Ankündigungen zu überprüfen. Hierbei kann die Authentisierung durch Übermittlung des Kennworts erfolgen.
- **Abwärtskompatibilität von RIP-2 mit RIP-1:** Das RIP-2 ermöglicht die Abwärtskompatibilität mit RIP-1. Die RIP-2-Nachrichten werden so strukturiert, dass ein RIP-1-Router einige Felder der RIP-2-Nachricht verarbeiten kann. Wenn ein RIP-1-Router eine RIP-2-Nachricht empfängt, verwirft er sie nicht, sondern verarbeitet nur die RIP-1-relevanten Felder. Oft können die RIP-2-Router auch mit dem RIP-1-Router zusammenarbeiten. Ein RIP-2-Router sendet eine RIP-1-Response auf ein RIP-1-Request.

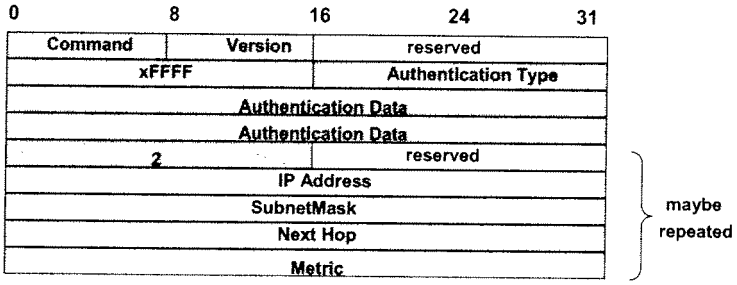


Bild: Struktur von RIP-2-Nachrichten

Struktur von RIP-2-Nachrichten

Für die Übermittlung von RIP-2-Nachrichten wird das Transportprotokoll UDP verwendet. Somit kann das RIP-2 im Schichtenmodell auch wie RIP-1 der Schicht 5 zugeordnet werden. Die RIP-2-Nachrichten werden ebenfalls wie RIP-1 über den UDP-Port 520 sowohl gesendet als auch empfangen. Somit kann der UDP-Port 520 als RIP-1/RIP-2-Port angesehen werden. Beim Versenden einer RIP-2-Nachricht auf ein Subnetz wird die IP-Multicastadresse 224.0.0.9 im IP-Header als IP-Zieladresse genutzt.

Jede RIP-2-Nachricht setzt sich aus einem Header und einer Vielzahl von RIP-2-Einträgen zusammen. Mit einem RIP-2-Eintrag (RIP-2 Entry) kann ein Router nur eine Route ankündigen. Der Header in der RIP-2-Nachricht enthält nur die beiden Angaben:

- **Command (1 Byte):** Hier wird angegeben, ob es sich um ein RIP-2-Request (x'01') oder um ein Response (x'02') handelt. Dieses Feld hat die gleiche Bedeutung wie beim Protokoll RIP-1.
- **Version (1 Byte):** Hier wird die RIP-Version angegeben (x'02').

Um sicherzustellen, dass die RIP-1-Router auch RIP-2-Nachrichten verarbeiten können, bleibt beim RIP-2 die Struktur von RIP-1-Nachrichten erhalten. Das RIP-2 nutzt diese Felder im RIP-Eintrag, die beim RIP-1 nicht verwendet und als "must be zero" definiert wurden. Die Felder "Command", "Address Family Identifier .. .. IPv4 Address" und "Metric" werden wie bei RIP-1 verwendet.

Mit einem RIP-Eintrag wird ein Router angegeben. Eine RIP-2-Nachricht darf maximal 25 Einträge je 20 Bytes enthalten. Sollen mehr als 25 Routen angekündigt werden, muss der Router mehrere Nachrichten senden.

Vergleicht man den RIP-1-Eintrag mit dem RIP-2- Eintrag, so stellt man fest, dass alle Felder, die beim RIP-1 nicht verwendet wurden, nun beim RIP-2 genutzt werden. Da ein RIP-Eintrag einen Router beschreibt, können zusätzliche Routen-Angaben beim RIP-2-gemacht werden.

Hierfür dienen folgende Felder im RIP-2- Eintrag:

- **Route Tag (2 Byte):** In diesem Feld kann die Routen-Markierung (Routen-Tag) angegeben werden. Die Möglichkeit wurde eingeführt, um zwischen RIP-basierten Routen (internal RIP routes) und Nicht-RIP-basierten Routen (external RIP routes) unterscheiden zu können. Das Routen-Tag kommt dann zum Einsatz, wenn die Kommunikation zwischen einem RIP-2-Router und einem BGP-Router (Border Gateway Protocol) unterstützt werden muss.
- **Subnet Mask (4 Byte):** Dieses Feld enthält die Subnetz-Maske des Netzziels im Feld IPv4-Address. Dadurch kann das RIP-2 in VLSM-Umgebungen eingesetzt werden. Daher ermöglicht das RIP-2 auch die CIDR-Unterstützung.
- **Next Hop:** Unter Verwendung dieses Felds kann ein Router eine Host-Route (d.h. eine Route direkt zu einem Rechner) ankündigen. In diesem Feld wird die IP-Adresse des Hosts eingetragen. Andere Router, die eine Ankündigung in diesem Netz empfangen, leiten die an den Host gerichteten Pakete direkt an diesen und nicht an den Router weiter.

Open Shortest Path First (OSPF)

OSPF (Open Shortest Path First) ist ein Routing-Protokoll innerhalb von Autonomen Systemen (AS), d.h. es ist ein Interior Gateway Protocol (IGP). Das OSPF wurde in den Jahren 1989-90 für IP-Netze entwickelt und ist eng mit dem OSI-Protokoll IS-IS (Intermediate System to Intermediate System) verwandt. Beide Protokolle können aber nicht direkt zusammenarbeiten. Das OSPF gehört zu der Gruppe der zustandsorientierten Routing-Protokolle im Unterschied zu RIP, das ein entfernungsorientiertes Routing-Protokoll ist. Die Routing-Information beim OSPF wird im Gegensatz zum RIP direkt in IP-Pakete eingebettet, d.h. ohne ein Transportprotokoll UDP zu nutzen, wie dies beim RIP der Fall ist. Hierfür wurde die Protokollnummer 89 dem OSPF im Header des IP-Pakets zugewiesen. Deshalb ist OSPF der Schicht 4 zuzuordnen. Die OSPF-Nachrichten sind klein, so dass man ohne Fragmentierung von IP-Paketen auskommen kann. OSPF unterstützt hierarchisches Routing. Zur Verwirklichung wird ein autonomes System in Bereiche eingeteilt

Zu unterscheiden ist zwischen:

- OSPF für IPv4 (OSPF for IPv4),
- OSPF für IPv6 (OSPF for IPv6).

Die aktuelle Spezifikation von OSPF für IPv4 ist OSPF Version 2 (OSPFv2, RFC 2328). Da man das Protokoll IPv6 Protocol Next Generation nennt, wird OSPF für IPv6 oft auch als OSPFng bezeichnet. Im weiteren wird unter der Abkürzung OSPF die Version für IPv4 verstanden. Die OSPF-Beschreibung entspricht OSPFv2.

Bei OSPF mit einer hierarchischen Struktur wird ein Autonomes System (AS) unterteilt in Areas, die über einem OSPF-Backbone miteinander gekoppelt sind. Zum Austausch der Routing-Information werden zwischen drei Routing-Bereichen und drei Router-Typen unterschieden.

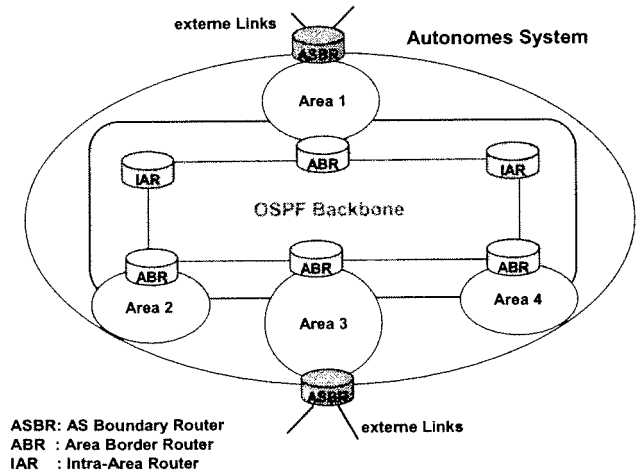


Bild: OSPF Netzstruktur

Drei Bereiche für die Routenwahl:

- Routing in einzelnen Bereichen (Intra-Area-Routing),
- Routing zwischen Bereichen (Inter-Area-Routing),
- Routing zwischen autonomen Systemen.

Drei Router-Typen:

- **Interner Router (Intra Area Router):** ein Router, der nur Nachbar-Router innerhalb eines Bereichs hat.
- **Bereichsgrenzen-Router (Area Border Router):** ein Router, dessen Nachbar-Router auch in anderen Bereichen sind. Über die Bereichsgrenzen-Router werden die Routing-Informationen zwischen den einzelnen Bereichen ausgetauscht.
- **AS-Grenzen-Router (AS Boundary Router):** Ein Router, der die autonomen Systeme miteinander verbindet.

Konzept: jeder Knoten verfügt über komplette Information der Netztopologie

Link-State-Algorithmus

- jeder Übertragungsabschnitt im Netz entspricht einem Eintrag in der Datenbasis
- jeder Knoten kann kürzesten Pfad zu allen anderen Knoten im Netz berechnen
- Link State Advertisements (LSAs) werden im gesamten AS geflutet

Flooding-Algorithmus

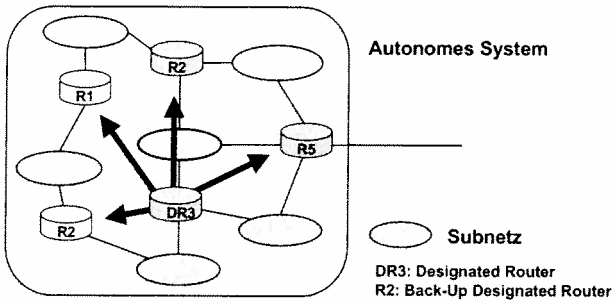
Beim Empfangen einer LSA-Meldung wird Eintrag in der Datenbasis überprüft:

- falls Eintrag noch nicht vorhanden, dann Hinzufügen und Broadcasten
- falls Eintrag vorhanden und neuer Wert niedriger als alter, dann Überschreiben und Broadcasten
- falls Eintrag vorhanden und neuer Wert höher als alter, dann Übertragen des bereits gespeicherten Wertes über Eingangslink
- falls beide Werte gleich sind, wird nichts unternommen

Bild: Protokoll OSPF

OSPF-Prinzip:

- Alle Ausgangsports eines Routers werden Kosten zugewiesen.
- Jeder Router erstellt aufgrund seiner Routenwahl-Datenbank einen Baum, in dem er selbst die Wurzel (Root) darstellt: SPF-Baum (Shortest Path First).
- Bereichs- und AS-Grenzen-Router verwalten zwei Topologie-Datenbanken (eins für den eigenen Bereich und eins für die über ihn erreichbaren Subnetze).
- Ferner gibt es einen Bereichsübergreifenden Topologiedatenbank.



Anstatt Fluten an alle Router (Aufwand: N x (N-1) Meldungen):

- Bestimmung eines Designated Router (DR) und Back-Up Designated Router
- Jeder Router sendet die LSA-Meldung an ein Designated Router (DR)
- Designated Router leitet die LSA-Meldung an alle andere Router weiter
- Aufwand: 2N + (N-1) = (3N –1) Meldungen

Um die Netze mit der Übermittlung von Routing-Informationen nicht zu stark zu belasten, wir in einem AS ohne Hierarchie oder in einem Area ein Designierter Router (DR) ausgewählt, der für die Übermittlung der Routing-Information (RI) in Form von Link State Advertisements (LSAs) zuständig ist.

Gleichzeitig wird ein Backup-DR bestimmt, der die Aufgabe hat, nach dem Ausfall des Designierter Routers dessen Funktionen zu übernehmen. Sind die Designierten Router bestimmt, werden die Nachbarschaften (Verbindungen vom DR zu anderen Routern) definiert, über die der Designierten Router die Routing-Information aus seiner RI-Datenbank an alle Router weitergeben kann.

Bild: Routing Informationsaustausch

Nur der Designierter Router (DR) ist für die Verteilung der Routing-Information (RI) zuständig, so dass Nicht-DR-Router untereinander nicht direkt kommunizieren können. Ein Designierter Router ist damit ein lokaler RI-Verteiler

- Common header

- Hello message
- Database description message

- Link state request message
- Link state update message
- Link state acknowledgement message

- Link state header

- Router links advertisement
- Network links advertisement
- Summary links advertisement
- External links advertisement

Nachrichten (Messages):

- Hello
- Link State Request
- Link State Update
- Link State Acknowledgement

Funktionen:

- Discovering neighbors
- Electing the designated router
- Initializing neighbors
- Propagating link state information
- Calculating route tables

Es existieren vier Nachrichtentypen: Hello sowie LS-Request, LS-Update und LS-Acknowledgment.

Es kommen zwei Gruppen von Funktionen vor: Erstens Erkennung der Routertopologie, Auswahl der beiden Designated Routers sowie die Einrichtung von Nachbarbeziehungen. Zweitens Verteilung von Routing Information als LSA (Link State Advertisment) und Berechnung der Routing-Tabellen (SPF-Baum) in jedem Router.

Alle vier Nachrichtentypen haben einen gemeinsamen Header. Ebenfalls gibt es vier Typen von LSAs (Router Links, Network Links, Summary Links und External Links) Auch die LSAs haben einen gemeinsamen Link-State-Header.

Bild: OSPF: Open Shortest Path First (Übersicht)

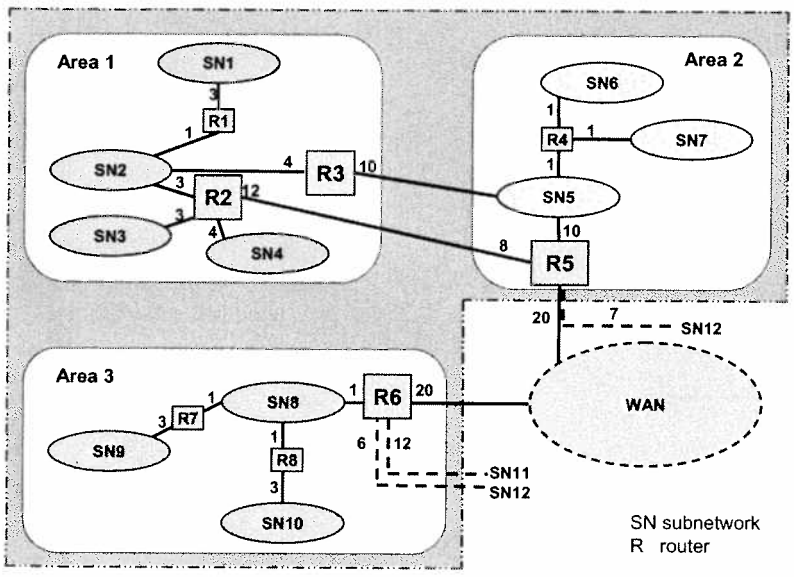


Bild: Autonomes System mit Areas

Um sich gegenseitig kennenzulernen, nutzen die Router ein Hello-Protokoll, das eine Hello-Nachricht zur Verfügung stellt. Diese Nachricht wird auch benutzt, um sowohl den DR als auch den Backup-DR zu bestimmen.

Die Hello-Nachricht besteht aus einem gemeinsamen Header und dem Hello-Teil. Hello-Nachrichten werden nur zwischen unmittelbaren Nachbarn ausgetauscht. Der Hello-Teil beinhaltet u.a. Zeitangaben über die Länge des Intervalls, in dem Hello-Nachrichten gesendet werden müssen (Hello-Intervall), und die Zeit, nach der ein Router seinen Nachbarn als ausgefallen erklären sollte (Ausfallentdeckungs-Intervall), nachdem von ihm keine Hello-Nachrichten mehr eingehen.

Ein OSPF-Router versucht zuerst, über sein Hello-Protokoll eine Verbindung mit seinen Nachbar-Routern einzurichten. Hierbei muss beachtet werden, dass bei WANs (z.B. X-25-Netz) die Adressen dieser Nachbar-Router im OSPF-Router eingetragen werden müssen. In einem LAN (Broadcast-Netz) verwendet der Router eine spezielle Broadcast-IP-Adresse (224.0.0.5), um einen OSPF-Router mit seinen Hello-Nachrichten anzusprechen.

Die Bereichsgrenzen-Router (inter-area) die Aufgabe, die über die Bereichsgrenzen hinweg erreichbaren Ziele ihrem eigenen Bereich bekannt zu geben, jedoch nur mit den entsprechenden Kosten und ohne die zugehörigen Topologieinformationen.

Die Kosten für die Ausgangsports müssen vom Netzmanager in allen Routern entsprechend eingestellt werden. Der Manager hat die Möglichkeit, bis zu 8 verschiedene Kostenarten (Metriken) zu konstruieren. Die Metrik wird aufgrund der Angaben im TOS-Feld (Type of Service) der Database-Description-Nachricht festgelegt. Werden mehrere Metriken unterstützt, muss jeder Router für jede Metrik einen SPF-Baum erstellen. Dies bedeutet, dass jede Metrik in jedem Router eine eigene Routing-Tabelle haben muss..

Den einzelnen Ausgangsports der Router wurden bestimmte Kosten zugewiesen, somit verursacht die Verbindung über das langsame WAN natürlich höhere Kosten als eine Kopplung über LANs. Zu beachten ist, dass immer nur die Kosten von einem Router zu einem Subnetz erfasst werden. Dies bedeutet, dass nur der Zugang zu einem Subnetz gewisse Kosten verursacht. Verbindungen von einem Netz zu einem Router werden mit dem Kostenwert 0 belegt.

Jeder Router muss für sich selbst eine Routing-Tabelle erstellen. Zu diesem Zweck baut er um sich - aufgrund der Eintragungen in seiner RI-Datenbank - einen überspannenden Baum auf, in dem er selbst die Wurzel (Root) darstellt und die Verzweigungen des Baumes die billigsten Wege zu allen möglichen Zielobjekten (Subnetze, Routern) sind. Ein solcher Baum wird als SPF-Baum (Shortest Path First) bezeichnet. Bestehen zwei Verbindungen zu einem Subnetz, wird immer der Pfad mit den niedrigeren Kosten verwendet. Der Pfad mit den höheren Kosten wird als redundanter Sekundärpfad angelegt und nur beim Ausfall des ersten Pfades genutzt. Sind die Kosten der Pfade gleich, wird der Datenstrom automatisch gleichverteilt.

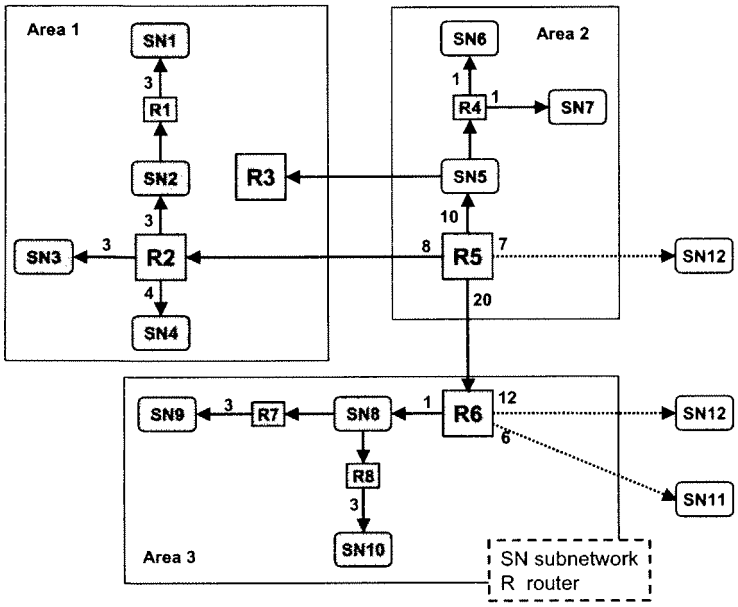


Bild: Shortest-Path-First Baum für Router R5

Im Bild ist der errechnete SPF-Baum für Router R5 gezeigt. Die Verbindungen und Kosten zu den einzelnen Objekten (Subnetze, Router) sind mit Richtungspfeilen dargestellt.

Da der Router R5 ein Bereichs- und AS-Grenzen-Router ist, muss er zwei Topologie-Datenbanken verwalten. Eine Datenbank für die Verbindungen und Kosten für den Bereich selbst sowie für Verbindungen und Kosten der über ihn erreichbaren Subnetze in den restlichen Bereichen, und eine Datenbank mit Verbindungen und Kosten von erreichbaren Objekten außerhalb des autonomen Systems.

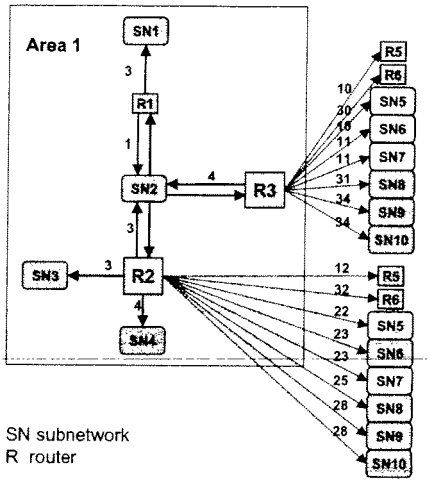
Ziel in AS	Nächste Router	Kosten
SN1	R2	14
SN2	R2	11
SN3	R2	11
SN4	R2	12
SN5	--	10
SN6	R4	11
SN7	R4	11
SN8	R6	21
SN9	R6	24
SN10	R6	24
R6	--	20

Ziel nicht in AS	Nächste Router	Kosten
SN11	R6	26
SN12	--	7

AS Autonomous System

Bild: Zwei Topologie-Datenbanken in Router R5

Daraus ergibt sich die folgende Information über die Innen - und Außen -Topologie für den Bereich 1.



Interior

Ziel in Area 1	Kosten über R2	Kosten über R3
SN1	6	7
SN2	3	4
SN3	3	7
SN4	4	8

Exterior

Ziel nicht in Area 1	Kosten über R2	Kosten über R3
R5	12	10
R6	32	30
SN5	22	10
SN6	23	11
SN7	23	11
SN8	25	31
SN9	28	34
SN10	28	34

Die Bereichsgrenzen-Router (inter-area) haben die Aufgabe, die über die Bereichsgrenzen hinweg erreichbaren Ziele ihrem eigenen Bereich bekannt zu geben, jedoch nur mit den entsprechenden Kosten und ohne die zugehörigen Topologieinformationen. In diesem Zusammenhang stellt sich die Frage, wie die Ziele in den einzelnen Bereichen über Bereichsgrenzen-Router von der Außenwelt erreichbar sind. Da die Routing-Informationen über erreichbare Bereichsziele in einer bereichsübergreifenden Topologie-Datenbank und somit von allen Bereichsgrenzen-Routern gelesen werden, können sie die Übertragungskosten zu allen Zielen außerhalb ihres eigenen Bereichs berechnen und die Kosten in die bereichsübergreifende Topologiedatenbank eintragen.

Bild: Innen- und Außen-Topologie Datenbank für Area 1

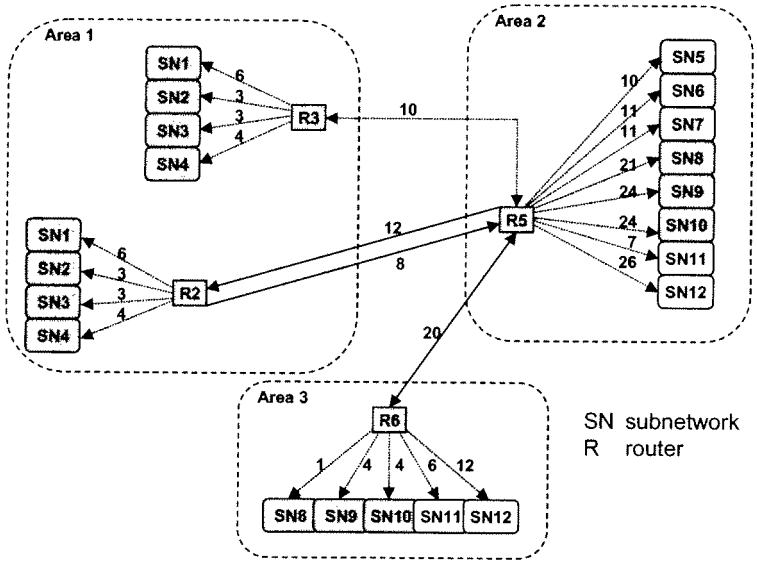


Bild: Topologie-Datenbanken in den zwei AS-Grenzen-Routern R5 und R6.

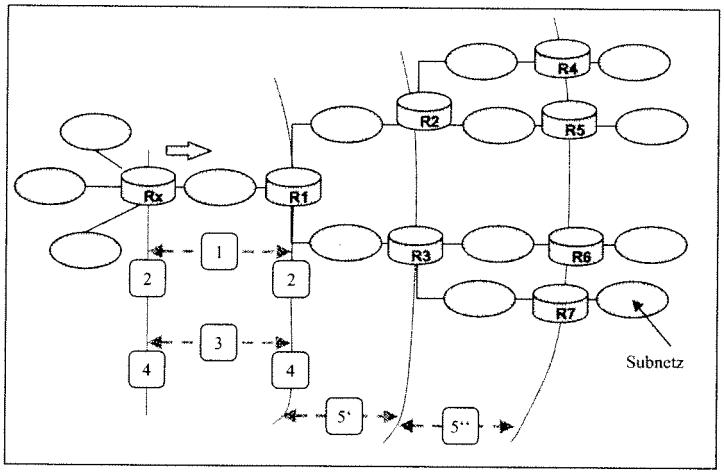


Bild: Hinzufügen eines neuen Routers

Hinzufügen eines Routers

Bei der Initialisierung eines neuen Routers in einem bereits bestehenden IP-Netz müssen die LSAs des neuen Routers an alle anderen Router übermittelt werden. Nach dem Empfang der LSAs vom neuen Router muss jeder andere Router im Netz die LSDB modifizieren, den SPF-Baum (nach dem Dijkstra-Algorithmus) für sich neu berechnen und neue Einträge in die Routing-Tabelle hinzufügen.

Das Hinzufügen eines neuen Routers führt zu den folgenden Schritten:

1. Der neue Router lernt die benachbarten Router kennen. Der neue Router Rx sendet eine Hello-Nachricht. Der benachbarte Router R1 antwortet ebenfalls mit der Hello-Nachricht. Die beiden Router Rx und R1 möchten nun eine Nachbarschaft aufbauen.
2. Der neue Router erstellt für sich die LSDB.
3. Der neue Router Rx muss für sich die LSDB erstellen. Hierfür tauschen die Router Rx und R1 die Database Description (DD)-Nachrichten aus. Ein DD-Nachricht von Rx enthält nur die eigene Routing-Information, d.h. die eigene Beschreibung. In den DD-Nachrichten übermittelt R1 die LSDB, in der die Routing-Information aller anderen Router außer R1 enthalten ist.
4. Synchronisation von LSDBs: Der neue Router Rx fordert mit dem Link-State-Request-Nachricht (LS) von dem benachbarten Router R1 bestimmte LSAs (z.B. die ihm noch fehlen). Router R1 sendet die angeforderten LSAs in den Paketen LS-Update. Der benachbarte Router R1 aktualisiert ebenfalls seine LSDB, so dass er mit der LS-Request-Nachricht vom neuen Router Rx bestimmte LSAs fordert. Router Rx sendet die von R1 angeforderten LSAs in den LS-Update-Nachrichten. Auf diese Weise haben die beiden Router Rx und R1, d.h. der neue Router und sein Nachbar, ihre LSDB synchronisiert. Nun besitzen sie eine aktuelle LSDB.
5. Der neue Router erstellt die Routing-Tabelle. Der Nachbar-Router aktualisiert seine Routing-Tabelle: Da die beiden Router Rx und R1 bereits die aktuellen LSDBs besitzen, berechnen sie ihre jeweiligen SPF-Bäume. Dann erstellt der neue Router Rx für sich die Routing-Tabelle, und sein Nachbar-Router R1 aktualisiert seine Routing-Tabelle. In der Routing-Tabelle beim R1 werden neue Netzziele hinzugefügt, die über den neuen Router Rx erreichbar sind.
6. Verteilung der Änderungen im Netz: Nachdem der Router R1 mit dem neuen Router Rx synchronisiert ist, verteilt R1 mit dem LS-Update_Nachricht die Änderungen im Netz an alle Nachbar-Router (R2 und R3), mit denen er eine Nachbarschaft unterhält. Die LS-Update-Nachricht enthält die von Rx erlernten LSAs. Nach Empfang der LSAs von R1 aktualisieren seine NachbarRouter R2 und R3 ihre LSDBs, berechnen ihre SPF-Bäume und aktualisieren ihre Routing-Tabellen.

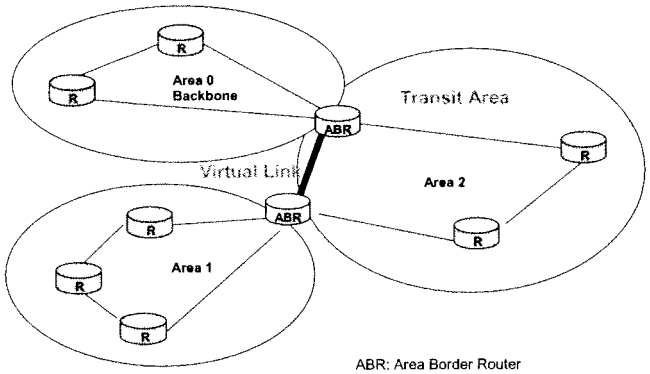


Bild: Virtual Link / Transit Area

Eine virtuelle Verbindung über den Transit-Bereich stellt eine logische Verbindung dar, die den Pfad mit den geringsten Kosten zwischen zwei AB-Router, die nicht über einem OSPF-Backbone verbunden sind und somit Router eines Transitbereichs verwenden müssen. Über die virtuelle Verbindung wird eine virtuelle Nachbarschaft gebildet, und die Routing-Infomationen in Form von LSAs werden ausgetauscht.

Wie bei physikalischen Nachbarschaften müssen vor dem erfolgreichen Aufbau einer virtuellen Nachbarschaft die Einstellungen in den virtuell verbundenen entsprechend übereinstimmen.

AS-übergreifendes Routing

Der AS-übergreifende Datenverkehr wird über einen AS-Grenzen-Router, kurz ASBR (Area Boundary Router), bzw. über mehrere ASBRs nach außen weitergeleitet. Eine Route, die zu einem Netzziel außerhalb eines AS führt, wird als externe Route bezeichnet. Eine externe Route ist definiert als beliebige Route, die sich nicht vollständig innerhalb eines OSPF-AS befindet.

Externe Routen werden durch einen oder mehrere ASBRs erlernt und im gesamten AS bekannt gemacht. Der ASBR kündigt die Verfügbarkeit externer Routen mit einer Reihe von LSAs für die externen Routen an. Die LSAs für die extemen Routen werden als eine Flut von OSPF-Nachrichten im gesamten AS (ausgenommen sog. Stub-Bereiche) gesendet. Die LSAs für die externen Routen gehen immer in die Berechnung von SPF-Bäumen und Routing-Tabellen ein. Der Datenverkehr zu externen Netzzielen wird innerhalb des AS gemäß den Routen mit den geringsten Kosten an den ASBR weitergeleitet.

Um die Menge der als Flut von OSPF-Nachrichten in Bereiche gesendeten Routing-Informationen noch weiter zu verringern, kann beim OSPF ein Bereich als sog. Stub-Bereich (Stub Area) eingerichtet werden. Ein Stub-Bereich kann einen bzw. mehrere ABRs haben, aber die externen Netzziele (d.h. Ziele in anderen autonomen Systemen) können nur über einen ABR erreicht werden.

Außerhalb des AS liegende Routen werden nicht als eine Flut von OSPF-Nachrichten in einen Stub-Bereich gesendet oder dort verbreitet. Das Routing auf alle außerhalb des AS liegenden Netze in einem Stub-Bereich erfolgt über eine Standardrou-

te (Zieladresse 0.0.0.0 mit der Netzmaske 0.0.0.0). Somit erfolgt das Routing zu allen außerhalb des AS liegenden Netzzielen mit Hilfe eines einzigen Eintrags in der Routing-Tabelle von Routem in einem Stub-Bereich.

Sicherheit
- Authentifizierung für Datenaustausch zwischen OSPF-Routern

Mehrere Pfade mit gleichen Kosten
- Verkehr kann bei gleichen Kosten über mehrere Pfade verteilt werden

Unterschiedliche Kostenmetriken für verschiedene Type-of-Service
- OSPF hat hierfür unterschiedliche Topologien und kann unterschiedliche Routen anbieten

Unterstützung von Unicast- und Multicastrouting
- Verwendet gleiche Topologie-Datenbasis

- Hierarchie innerhalb eines AS
AS kann in Areas untergliedert werden

Zwei Ebenen der Hierarchie
- Area
- Backbone (der die Areas verbindet)
  - Inter-Area-Kommunikation verläuft immer über Backbone-Area
  - Gehört ein Router zu zwei Areas, ist er automatisch ein Backbone-Router

Bild: Fortgeschrittene Funktionen in OSPF

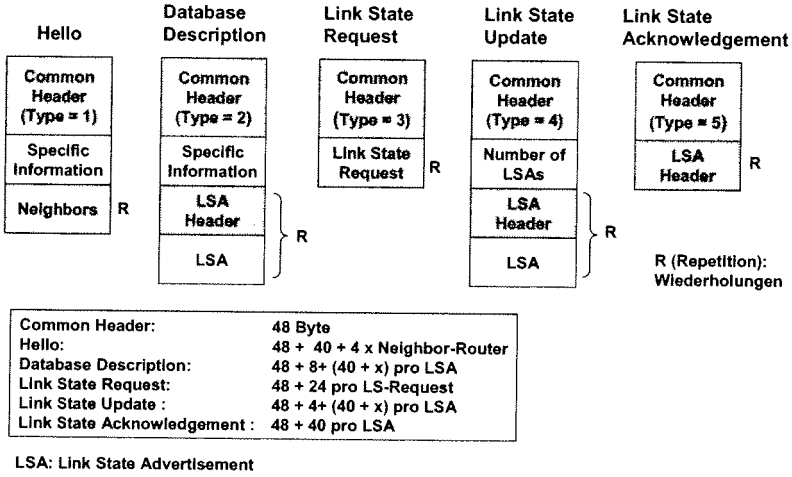


Bild: OSPF: Open Bild: OSPF-Nachrichten (Übersicht)

OSPF-Nachrichten

Um Routing-Tabellen in den Routern nach dem Protokoll OSPF zu erstellen und zu aktualisieren, müssen entsprechende OSPF-Nachrichten zwischen den Routern übermittelt werden. Beim OSPF sind folgende Typen von OSPF-Nachrichten definiert:

- Hello,
- Database Description,
- Link State Request,
- Link State Update,
- Link State Acknowledgment (Ack).

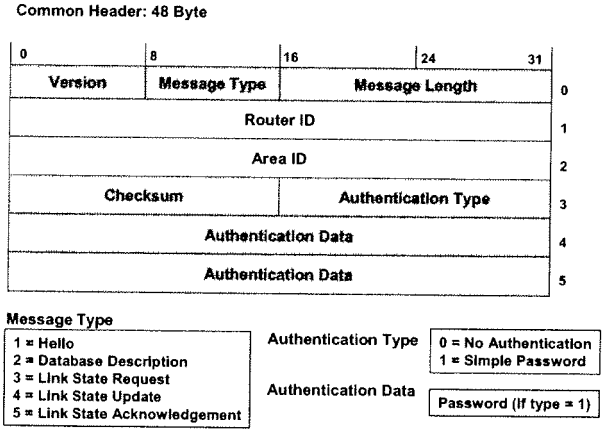


Bild: OSPF: Common Header

Aufbau von OSPF-Nachrichten

Jede OSPF-Nachricht setzt sich aus OSPF-Header und Nachrichteninhalt zusammen. Es werden hier nur Nachrichten des Protokolls OSPFv2 betrachtet.

Die Angaben im OSPF-Header sind:

- **Version:** Hier wird die Version des Protokolls OSPF angegeben (d.h. die Version 2).
- **Type (Nachrichtentyp):** Hier wird der Typ (d.h. die Bedeutung) der OSPF-Nachricht angegeben.
- **Message Length (Nachrichtenlänge):** Die Länge der gesamten Nachricht in Bytes (einschließlich des gemeinsamen Headers).
- **Router ID:** Die Identifikation (ID) des Routers, der die OSPF-Nachricht abgeschickt hat.
- **Area ID (Bereich-ID):** Die Identifikation des Bereichs, in dem die OSPF-Nachricht abgesesetzt wurde. Eine OSPF-Nachricht wird normalerweise einem Bereich zugeordnet. Wird eine Nachricht über eine virtuelle Verbindung (über mehrere Bereiche) gesendet, erhält er die Identifikation 0.0.0.0, d.h. die Identifikation des Backbone-Bereichs.

- **Checksum (Prüfsumme):** Diese Prüfsumme über den Nachrichteninhalt mit Ausnahme des Feldes Authentication soll es ermöglichen, Fehler in der Nachricht zu entdecken.
- **AuType (Authentication Type, Art der Authentisierung):** Hier wird angezeigt, welche Art der Authentisierung (Paßwort, Kryptografische Summe etc.) verwendet wird.
- **Authentication:** In diesem 64-Bit-Feld werden die Authentisierungsangaben gemacht.

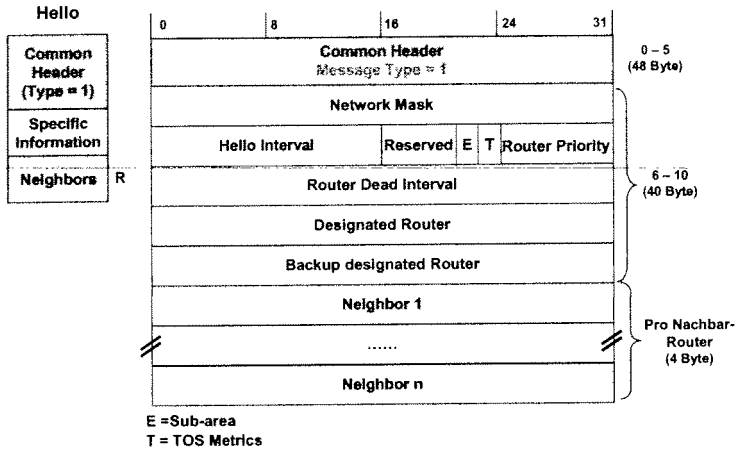


Bild: OSPF: Hello-Nachricht

Hello-Nachricht

Das Hello-Protokoll wird für die folgenden Funktionen verwendet:

- Um die benachbarten Router bei der Initialisierung eines neuen Routers bzw. beim Aufbau einer Nachbarschaft anzusprechen.
- Um in regelmäßigen Zeitabständen zu prüfen, ob die Verbindungen intakt sind.
- Um sowohl einen designierten Router (DR) als auch einen Backup-DR in broadcastorientierten Netzen zu bestimmen.

Die einzelnen Angaben in der Hello-Nachricht sind:

- **Network Mask (Netz-Maske):** Dieses Feld wird für das Subnetting verwendet. Hier wird die Netz-Maske (bzw. Subnetz-Maske) dieses Router-Interfaces (Schnittstelle) angegeben, über das die Hello-Nachricht abgeschickt wurde.
- **Hello Interval (Hello-Intervall):** Das Zeitintervall zwischen den regelmäßig gesendeten Hello-Nachrichten in Sekunden. Standardmäßig beträgt das Hello-Intervall 10 Sekunden.
- **Options (Optionen):** Mit Hilfe einzelner Bits in diesem Byte werden einige Router-Besonderheiten angegeben (z.B. ob der Router über dieses Interface, über das die Hello-Nachricht abgeschickt wurde, die AS-externen LSAs senden und empfangen kann).
- **Router Priority (Router-Priorität):** Hier wird die Router-Priorität angegeben. Sie ist von Bedeutung bei der Auswahl des designierten Routers.
- **Router Dead Interval (Ausfallentdeckungs-Intervall):** Die Anzahl von Sekunden, bis der Router einen Nachbar-Router als ausgefallen (tot) erklärt.
- **Designated Router (Designierter Router):** Falls der Router, der die Hello-Nachricht abgeschickt hat, ein designierter Router ist, wird hier die IP-Adresse des Interfaces angegeben, über das diese Nachricht gesendet wurde. Gegebenenfalls wird mit 0.0.0.0 angezeigt, dass es sich um keinen designierten Router handelt.
- **Backup D Designated Router (Backup-DR, Ersatz-DR):** Falls der Router, der die Hello-Nachricht abgeschickt hat, ein Backup-DR ist, wird hier die IP-Adresse des Interfaces angegeben, über das die Nachricht gesendet wurde. Gegebenenfalls wird mit 0.0.0.0 angezeigt, dass es sich um keinen Backup-DR handelt.
- **Neighbors (benachbarte Router):** Hier werden die IDs jener Router angegeben, von denen der Absender-Router der Hello-Nachricht bereits gültige Hello-Nachrichten empfangen hat.

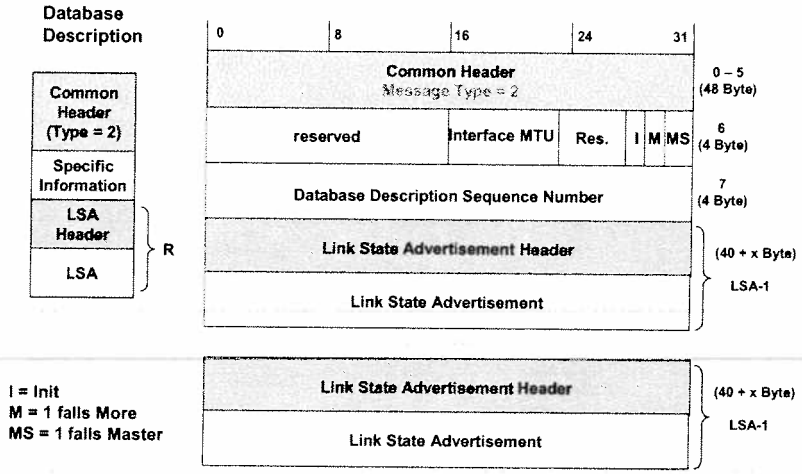


Bild: OSPF: Database-Description-Nachricht

Database-Description-Nachricht

Falls zwei benachbarte Router bereits eine Nachbarschaft aufgebaut haben, müssen sie ihre LSDBs synchronisieren (d.h. ständig abgleichen). Hierfür wird das Exchange Protocol verwendet. Dieses Protokoll funktioniert nach dem Anfrage/Antwort-Prinzip, so dass zuerst festgelegt wird, welcher Router ein Master-Router ist und welcher als Slave-Router funktioniert. Danach werden die Beschreibungen von LSDBs zwischen diesen Routern ausgetauscht. Hierbei werden die LSDB-Inhalte in den Database Description Nachrichten übermittelt. Der Master-Router fordert die LSDB-Inhalte an und antwortet darauf durch das Absenden eines bzw. von mehreren Database Description (DD) Nachrichten.

Eine DD-Nachricht setzt sich aus OSPF-Header und DD-Teil zusammen. Im DD-Teil sind bestimmte Steuerungsangaben und die LSDB-Beschreibung - in Form von LSA-Headern - enthalten.

Der DD-Teil enthält folgende Angaben:

- **Interface MTU:** Hier wird angezeigt, wie groß ein IP-Paket ohne Fragmentierung sein darf, das über das betreffende Router-Interface gesendet wird.
- **Options (Optionen):** Einige Bits in diesem Feld werden verwendet, um bestimmte Router-Besonderheiten anzuzeigen.
- **I-Bit (Init Bit):** Falls I = 1 ist, wird damit angezeigt, dass diese DD-Nachricht das erste innerhalb der Folge von DD-Nachrichten ist.
- **M-Bit (More Bit):** Mit M = 1 wird darauf verwiesen, dass nach dieser DD-Nachricht noch weitere DD-Nachricht aus einer Folge kommen.
- **MS-Bit (Master/Slave Bit):** Mit MS = 1 zeigt der Absender-Router an, dass er der Master-Router während des LSDB-Abgleichprozesses ist.
- **DD Sequence Number (DD: Database Description):** Die gesendeten DD-Nachrichten werden fortlaufend numeriert. Hier wird die Sequenznumnier der DD-Nachricht angegeben. Die Anfangsnummer ist eindeutig zu wählen.
- **LSA-Header (Link State Advertisement):** Die LSDB-Beschreibung wird in LSA-Headern übermittelt.

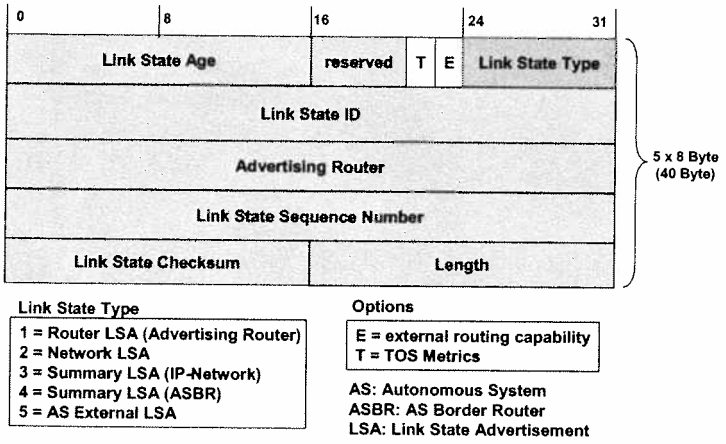
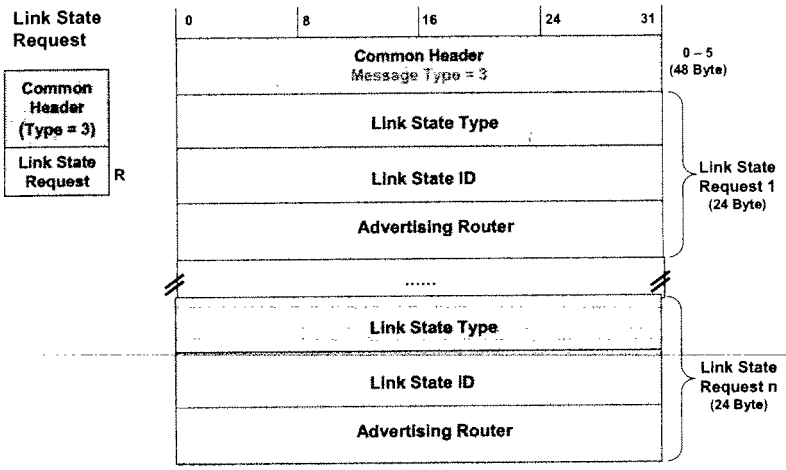


Bild: OPSF: Link State Message

Linkzustands-Nachrichten (Link-State-Messages)

Link-State-Messages tauschen die Routing-Information in Form von LSAs zwischen den benachbarten Routern aus. Hierzu gehören folgende OSPF-Nachrichten: Link State Request (LS-Request), Link State Update (LS-Update), Link State Ack (LS-Ack, Acknowledgment).



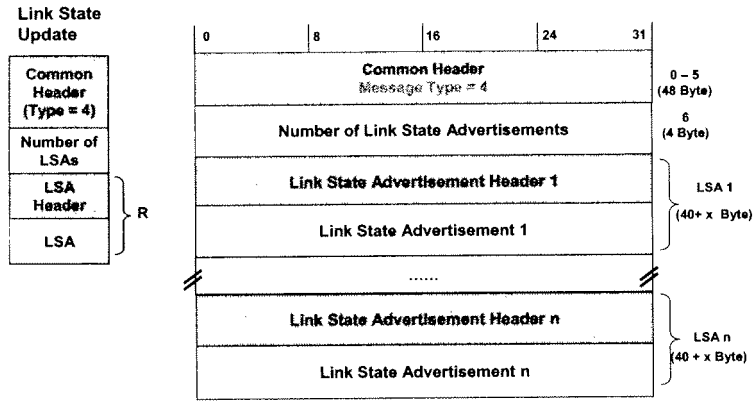
Ein Router kanl die veraltete Routing-Information in Form von LSAs von seinen benachbarten Routern mit einer LS-Request-Nachricht anfordern. Diese Nachricht kann mit dem LS-Update-Nachricht beantwortet werden. Hat sich beispielsweise die Routing-Tabelle in einem Router verändert, so sendet er die aktuelle Routing-Information in Form von LSAs in den LS-Update-Nachrichten an die benachbarten Router. Sie bestätigen ihm den Empfang von aktuellen LSAs mit den LS-Ack-Nachrichten.

Bild: OSPF: Link-State-Request-Message

Die LS-Request-Nachricht setzt sich aus dem OSPF-Header und einem LS-Request-Teil zusammen. Im LS-Request-Teil können mehrere LS-Anforderungen enthalten sein, und mit ihnen wird angezeigt, welche LSAs der Router haben möchte.

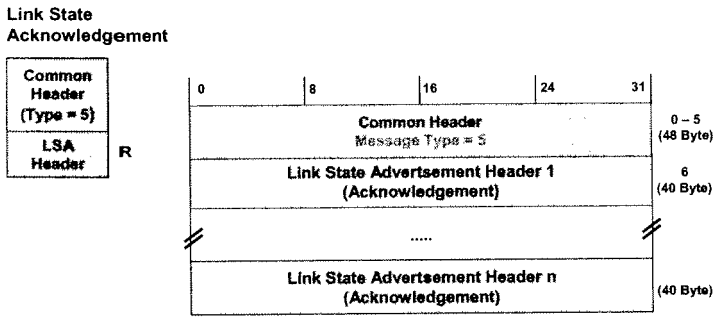
Die LS-Anforderung enthält folgende Angaben:

- **LS Type:** Hier wird angegeben, um welchen LS-Typ es sich handelt.
- **Link State ID:** Hier wird die LS-Identifikation (LS-ID) angegeben. Die LS-ID ist vom LS-Typ abhängig. Zum Beispiel stellt die IP-Adresse eines Interfaces in einem designierten Router die Identifikation eines Network-LSA dar.
- **Advertising Router:** Die Identifikation des Quell-Routers von LSA. Beim Einsatz eines designierten Routers wäre hier dessen Identifikation enthalten.



Die Veränderungen der Routing-Information werden in Form von LSAs in LS-Update-Nachrichten übermittelt. Sie werden auch als Antworten auf LS-Request-Nachrichten gesendet. Eine LS-Update-Nachricht enthält den OSPF-Header und einen LS-Update-Teil mit mehreren LSAs. Es wird die Anzahl von LSAs angegeben, die in der Nachricht enthalten sind. Die einzelnen LSAs bestehen aus dem Header und Informationsteil. Um das Verteilen von LSAs zuverlässig zu gestalten, werden die in den LS-Update-Nachrichten übertragenen

Bild: OSPF: Link State Update Message



LSAs durch ein LS-Ack quittiert. Die LS-Ack-Nachricht enthält die Liste von Headern dieser LSAs, deren Empfang bestätigt wird.

Bild: OSPF: Link State Acknowledgement Message

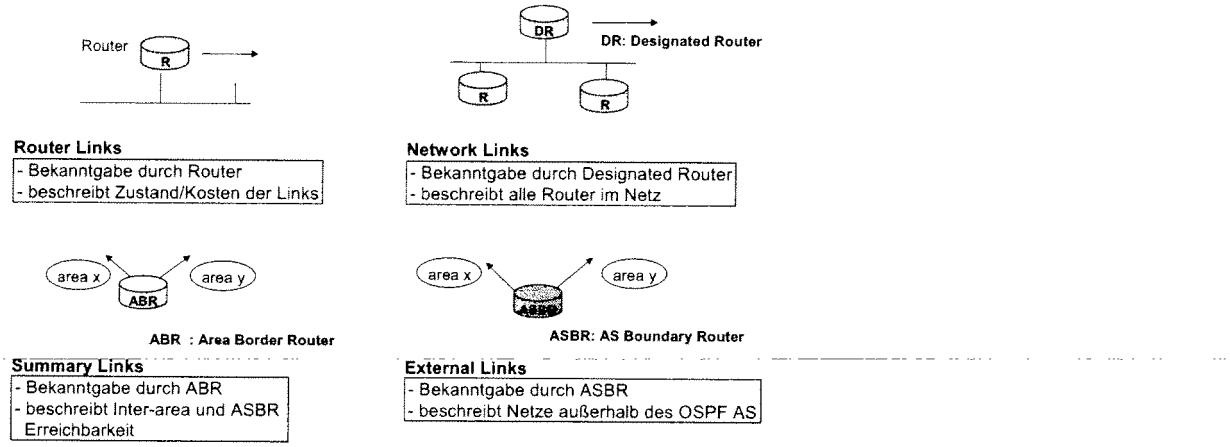


Bild: OSPF Link State Advertisements

Im allgemeinen definiert OSPF folgende LS-Typen:

- Network-LSAs verwendet man, um Broadcast-orientierte Netze (genauer gesagt Subnetze) im Hinblick auf das Routing zu beschreiben. Im Network-LSA eines Broadcast-orientierten Subnetzes wird angegeben:
- Subnetz-Maske
- ID (Identifikation) des designierten Routers,
- IDs aller Router, die am Subnetz angeschlossen sind.

Die Bedeutung von LSAs der Typen 3, 4 und 5.

- Eine Network-LSA vom Typ 3 (d.h. Summary-LSA, IP Network) wird vom Bereichgrenzen-Router (d.h. ABR) verwendet, um die erreichbaren Netzziele mit den Kosten in seinem Bereich im Backbone-Bereich ankündigen zu können. In einer LSA kann nur ein Netzziel (eine IP-Adresse) angegeben werden. Daher müssen für die Bekanntmachung mehrerer Ziele mehrere LSAs übermittelt werden.
- Eine Network-LSA vom Typ 4 (d.h. Summary-LSA, ASBR) wird vom ABR generiert, um die AS-Grenzen-Router (d.h. ASBR) und die mit ihnen verbundenen Kosten in seinem Bereich bekanntzumachen. In einer LSA kann nur ein ASBR angegeben werden.
- Eine Network-LSA vom Typ 5 (d.h. AS-external-LSA) wird vom ASBR generiert, um die außerhalb des eigenen AS liegenden Netzziele mit ihren Kosten in seinem AS bekannt zu machen. Der ASBR macht im eigenen AS die über ihn erreichbare AS-Außen-Netzziele bekannt.

Jede LSA setzt sich aus einem LSA-Header und aus LSA-Daten zusammen.

Die einzelnen Angaben im LSA-Header sind:

- **LSA Age:** Die Angabe der Zeit (in Sekunden), die seit der LSA-Generierung vergangen ist.
- **Options:** Dieses Feld enthält festgelegte Bits (z.B. E, MC, N/P, ...), mit denen einige Router-Besonderheiten angegeben werden. Ist hier beispielsweise E=1, bedeutet dies, dass der Advertising-Router einen Extenal Link hat, d.h. ein Interface zum anderen AS.
- **Link State ID (LS-ID, Identifikation):** Die LS-ID ist vom LSA-Typ abhängig und hat folgende Bedeutung:
  - LSA-Typ 1: LS-ID ist ID des Routers, der die LSA generiert hat.
  - LSA-Typ 2: LS-ID ist IP-Adresse des Interface des designierten Routers.
  - LSA-Typ 3: LS-ID ist IP-Adresse des Netzziels.
  - LSA-Typ 4: LS-ID ist ID des Routers (ASBR), der die LSA gesendet hat.
  - LSA-Typ 5: LS-ID ist IP-Adresse des Netzziels.
- **Advertising Router:** Hier wird die ID des Routers angegeben, der diese LSA generiert hat.
- **LS Sequence Number:** Hier werden die gesendeten LSAs fortlaufend nummeriert.
- **LS Checksum:** Hier ist eine Prüfsumme enthalten, mit der die ganze LSA ohne Feld LS Age überprüft wird.
- **Length:** Die LSA-Länge in Bytes.

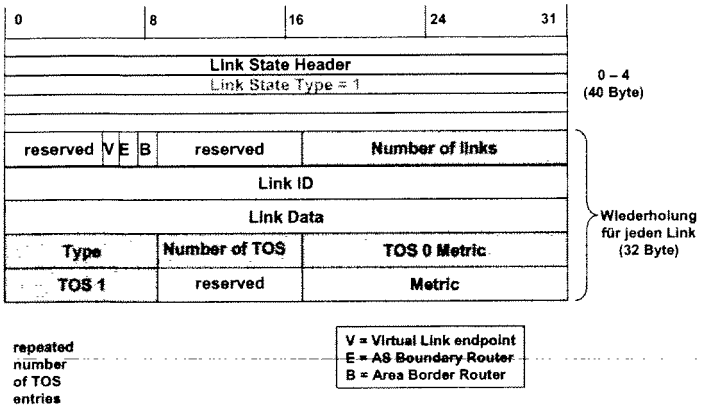


Bild: OSPF: Router Link State Advertisement

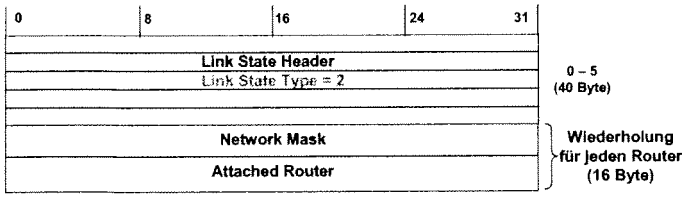


Bild: OSPF: Network Link State Advertisement

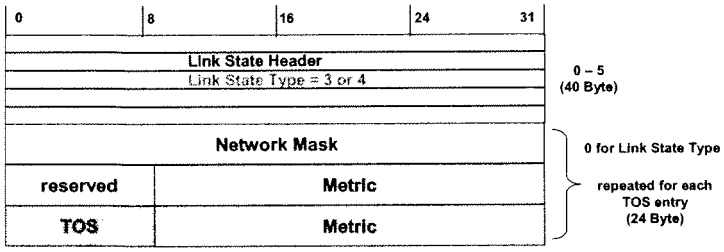


Bild: OSPF: Summary Link State Advertisement

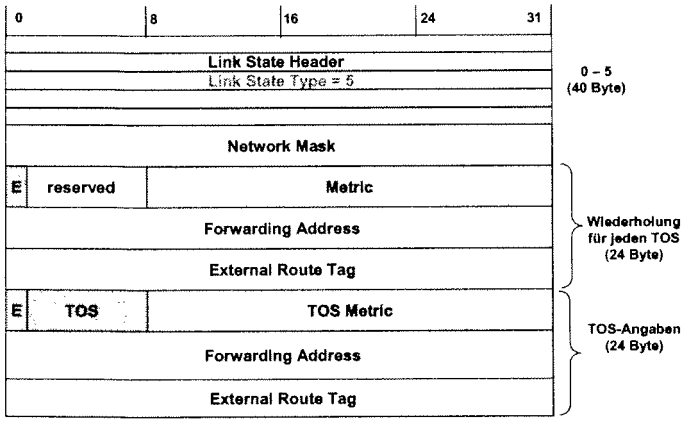


Bild: OSPF: External Link State Advertisement

LSA-Typen und -Angaben

Die Routing-Information nach dem OSPF wird in Form von LSAs (Link State Advertisements) zwischen den Routern so verteilt, dass jeder Router innerhalb eines Bereichs sich eine Datenbank mit der Routing-Information, d.h. eine LSDB, erstellen kann. Die LSDB dieser LSAs beschreibt daher den Zustand des Bereichs aus OSPF-Sicht. Wird ein autonomes System AS auf Bereiche nicht aufgeteilt, stellt das ganze AS einen Bereich dar

Die Router-LSAs beschreiben die aktiven Router-Interfaces und Verbindungen, über die der Router an den Bereich gebunden ist. Bei der Beschreibung des Router-Interfaces wird u.a. angegeben,

- um welche Link-Art es sich handelt (z.B. Point-to-Point-Link, Virtual Link,
- welche Metrik-Arten (d.h. Arten von Kosten) der Router unterstützt.

Die Router-LSAs werden nur innerhalb des betreffenden Bereichs verteilt.

Besonderheiten von OSPFv2

Einige OSPF-Besonderheiten sind:

- **Schleifenlose Routen:** Der designierte Router führt zur Synchronisation von einzelnen Routern innerhalb eines Bereichs. Dadurch entstehen keine logischen Schleifen bei der Berechnung von Routen, und somit tritt das Count-to-Infinity-Problem beim OSPF nicht auf, wie dies beim RIP der Fall war.
- **Schnellere Konvergenz als beim RIP:** OSPF kann Topologie-Änderungen schneller erkennen und übermitteln als RIP.
- **VLSM- bzw. CIDR-Unterstützung:** Beim OSPF wird die Präfixlänge (d.h. Länge der Subnetz-Maske) übermittelt. Dadurch ist die VLSM- bzw. CIDR-Unterstützung mit der Aggregation von Routen möglich.
- **Skalierbarkeit großer IP-Netze:** Beim OSPF werden autonome Systeme in Bereiche aufgeteilt. Dadurch lässt sich die Größe von Routing-Tabellen verringern. Zu einem Bereich kann eine aggregierte Route führen, die alle Routen zu den einzelnen Netzzielen innerhalb des Bereichs zusammenfaßt. Dadurch ist OSPF für große und sehr große IP-Netze geeignet, die beliebig erweiterbar sind.
- **Unterstützung für Authentisierung:** Der Informationsaustausch auf OSPF-Routen lässt sich authentisieren.
- **Unterstützung für externe Routen:** Routen innerhalb des OSPF-AS werden innerhalb des AS angekündigt, damit OSPF-Router die Route geringster Kosten zu externen Netzen berechnen kann.
- **Kompatibilität zum OSPFv1** durch TOS-Angaben.

Open Shortest Path First für IPv6

OSPF (Open Shortest Path First) für das Protokoll IPv6 stellt eine an die IPv6-Besonderheiten angepasste Form des OSPFv2 (d.h. OSPF der Version 2) für IP4 dar und wird im IETF-Standard RFC 2780 beschrieben. Da man das IPv6 als Protocol Next Generation bezeichnet, wird im folgenden die Abkürzung OSPFng für das OSPF für IPv6 verwendet. Die Version 2 des OSPFs für das Protokoll IP4 wird hier als OSPFv2 bezeichnet.

Die wichtigsten OSPFng-Besonderheiten sind:

- Beim OSPFng wird der Begriff "Link" für ein Subnetz bzw. ein Netz verwendet. Damit trifft die einheitliche OSPFng-Beschreibung sowohl für die broadcastorientierten LANs als auch für die verbindungsorientierten WANs zu.
- OSPFng ist hierarchisch über IPv6 angesiedelt. Die OSPFng-Nachrichten mit der Routing-Information werden direkt in die IP-Pakete eingebettet. Somit ist das OSPFng der Schicht 4 im Schichtenmodell zuzuordnen.
- OSPFng eignet sich für große Netze. Das OSPFng (ebenso wie OSPFv2) wurde insbesondere für den Einsatz in großen Netzen konzipiert, die auf eine Vielzahl von autonomen Systemen aufgeteilt werden können.
- OSPFng-Nachrichten: Um die Routing-Information in Form von LSAs (Link State Advertisements) zu übermitteln, verwendet das OSPFng (ebenfalls wie OSPFv2 für IPv4) folgende Pakete:
  - Hello
  - Database Description
  - Link State Request
  - Link State Update
  - Link State Acknowledgment
- Link State Database (LSDB) und die Routing-Tabelle (RT): Die LSDB und RT werden beim OSPFng identisch aufgebaut wie LSDB und RT beim OSPFv2.
- Bildung einer Nachbarschaft: Beim OSPFng werden die Nachbarschaften zwischen den Routern nach den gleichen Prinzipien wie beim OSPFv2 gebildet.
- Einsatz eines designierten Routers: Beim OSPFng wird (wie beim OSPFv2) ein designierter Router für die Verteilung der Routing-Information in Broadcast-Netzen und in NBMA-Netzen eingesetzt.
- Erstellung der Routing-Tabelle beim OSPFng: Die Routing-Tabelle beim OSPFng wird nach den gleichen Prinzipien wie beim OSPFv2 erstellt; d.h. die Berechnung des SPF-Baums (Shortest Path First) erfolgt beim OSPFng nach dem Algorithmus von Dijkstra.

Routing-Protokoll EGP

EGP (Exterior Gateway Protocol) ist ein externes Routing-Protokoll, das entworfen wurde, um Unternehmensnetze mit Hilfe eines Routers an das Internet-Kernnetz anzuschließen. Es ist durchaus auch als internes Routing-Protokoll geeignet, so dass auf den Einsatz weiterer Routing-Protokolle verzichtet werden kann. EGP erledigt drei wesentliche Aufgaben, die zur Lösung des Routenwahlproblems ausreichen und mit Hilfe von zehn verschiedenen Nachrichtentypen erfüllt werden.

Die Hauptaufgaben sind:
- Verbindungen zu Nachbar-Routern herstellen,
- Erreichbarkeit der Nachbar-Router feststellen,
- Routing-Informationen austauschen.

Das EGP-Protokoll ist ein statisches Protokoll, d.h. die Routing-Tabelle an andere EGP-Router muss vom jeweiligen Systemadministrator manuell vorgegeben werden.

Die Felder des Headers haben die Bedeutung:

- **Versionsnummer:** Beinhaltet die Konstante 1.
- **Type:** Als Nachrichtentypen kommen in Frage: - Type 1: Liste der Verbindungen (Routenwahl-Informationen; Update) - Type 2: Aktualisierungsanforderung (Poll) - Type 3: Verbindungsauf- u. abbau: Request, Confirm, Refuse, Cease, Cease-Acknowledge - Type 5: Hello (Code = 0) I heard you - Type 8: Error
- **Code:** Der Inhalt ist abhängig vom Typ.
- **Status:** Inhalt unterschiedlich, er hängt vom Nachrichtentyp ab.
- **Prüfsumme:** Als Prüfwert wird ein 2 Byte großes Einerkomplement für die EGP-Nachricht - beginnend ab der Versionsnummer - gebildet.
- **Nummer des autonomen Systems:** Hier steht die Nummer des autonomen Systems, an dem der Router angeschlossen ist, der diese Nachricht versendet hat. - Sequenznummer: Mit dieser Nummer können die Antworten den Anfragen eindeutig zugeordnet werden.

Zur Verbindungsaufnahme mit seinen Nachbarn sendet ein Router zunächst eine Request-Nachricht (Typ = 3; Code = 0). In diesem Request teilt er die bei seiner Konfiguration eingestellte Nummer des ihm angeschlossenen autonomen Systems mit. Der Nachbar kann daraufhin die Verbindung mit Confirm-Nachricht (Type = 3; Code = 1) bestätigen, die Verbindung mit Refuse-Nachricht (Type = 3; Code = 2) zurückweisen oder eine Error-Nachricht (Type = 8; Code = 0) senden.

Wird die Verbindung von dem Nachbar-Router nicht gewünscht, sendet er ein „Refuse" dort steht im Statusfeld die 1, wenn der Übermittler keinen Tabellenplatz mehr hat; die 2, wenn der Administrator die Nachbarschaftsbeziehung verboten hat.

Zur Aufrechterhaltung der Verbindung tauschen benachbarte Router in periodischen Abständen Hello-Nachrichten (Type = 5; Code = 0) und I-heard-you-Nachrichten (Type = 5; Code=1) aus. Im Fehlerfall wird eine Hello-Nachricht mit einer entsprechenden Fehlermeldung beantwortet.
- Type = 5
- Code: 0 für „Hello" und 1 für „I heard you"
- Status: Kann hier vier verschiedene Werte annehmen: 1 = erreichbar; 2, 3, 4 = unerreichbar.
- Sequenznummer: Mit Hilfe dieser Nummer können die Antworten eindeutig den Anfragen zugeordnet werden.
- Min. Zeitintervall: Intervall in Minuten, wie lange der Ziel-Router auf Lebenszeichen warten soll.
- Letzte Update-Nummer: Hier steht die Nummer der Zuletzt empfangenen Routing-Informationen.

Die Besonderheit von EGP besteht darin, dass ein EGP-Router nicht die Kosten zu einem Ziel bekannt gibt, sondern nur, ob das Ziel erreichbar ist oder nicht. Dies setzt - wie eingangs erwähnt - eine schleifenfreie Topologie voraus.

Das Senden der Routing-Informationen erfolgt mit der Update-Nachricht (Typ = 1, Code = 0) und findet nur nach vorheriger Anfrage Poll (Typ = 2, Code = 0) statt, die ebenfalls in periodischen Abständen gesendet wird.

Update-Nachricht. Die Routing-Information besteht u. a. aus der IP-Adresse des Netzes, das über den sendenden Router erreichbar ist, der Anzahl der internen Router dieses Netzes und der Anzahl der externen Router, auf die sich das Paket bezieht. Es folgt die Liste von Routeradressen und die Zugehörige Anzahl der Netze, die ihrerseits die Adresse und Entfernung zu jedem angeschlossenen Netz enthalten.

Die Bedeutungen der einzelnen Angaben in der Update-Nachricht sind:
- **U-Flag:** 0 = Antwort auf ein Poll, 1 = unaufgeforderte Routing-Information,
- Nummer des autonomen Systems: Hier steht die Nummer des autonomen Systems, an dem der Router angeschlossen ist, der diese Nachricht versendet hat.
- **Fragmentnummer:** Dies Feld enthält eine Identifikationsnummer, falls eine Routing-Information nicht in ein komplettes Paket passt (also fragmentiert werden musste). Ansonsten steht hier der Wert 0.

- **Nummer des letzten Fragment:** Beinhaltet die Nummer des letzten Fragments. Wenn keine Fragmentierung vorgenommen wurde, dann steht hier 0.
- **ID-Nummer:** Mit dieser Nummer kann einem Poll eindeutig die Update-Nachricht zugeordnet werden.
- **IP-Nummer des Quell-Netzes:** Hier wird lediglich der Netzanteil der IP-Adresse angegeben, d.h. bei einer Klasse-B-Adresse ist das zweite und dritte Byte auf 0 gesetzt.
- **Anzahl interne Router:** Gibt die Anzahl der internen Router an, für die diese Routing-Information gilt.
- **Anzahl externe Router:** Gibt die Anzahl der externen Router an, für die diese Routing-Information gilt.
- Anzahl der Netze: Gibt die Anzahl der Netze an, die in den nachfolgenden Feldern vom Router n erreicht werden können.
- **IP-Nummer von Router n:** Hier steht nur der Host-Anteil der IP-Adresse, d.h. je nachdem, welche IP-Klasse angesprochen wird, entfällt das erste, zweite und dritte Byte der IP-Adresse.
- **Zielnetz m:** Hier steht der Netzanteil der IP-Adresse des Zielnetzes, das über den Router n erreicht werden kann. Die Länge ist somit wieder abhängig von der IP-Klasse, also ein, zwei oder drei Bytes groß.
- **Entfernung:** Gibt an, wieviele Router zwischen Router n und dem Zielnetz m liegen.

Die Felder Zielnetz und Entfernung geben somit die Anzahl der erreichbaren Netze über einen Router an. Für jeden Router werden Anzahl der Netze, IP-Nummer von Router n, Zielnetz m und Entfernung so oft wiederholt, bis sämtliche Erreichbarkeitsinformationen aufgeführt wurden.

Besonderheit der Entfernungsangabe in der Update-Nachricht: EGP zieht die in den Routing-Informationen enthaltenen Entfernungsangaben nicht zu eigenen Berechnungen heran. Relevant ist lediglich, ob ein Nachbar erreichbar ist oder nicht.

Einstellung einer EGP-Nachbarschaftbeziehung: Eine Verbindungseinstellung zwischen benachbarten Routern wird durch Cease (Typ = 3; Code = 3) initiiert, das vom Nachbarn mit Cease-Acknowledgement (Typ = 3; Code = 4) bestätigt oder mit einer Fehlermeldung Error (Typ = 8, Code = 0) beantwortet wird.

IS-IS: Intermediate System to Intermediate Systeme

Ursprünglich im Rahmen von ISO/OSI für das dortige verbindungslose Netzprotokoll (CLNP: Connectionless Network Protocol) entworfen

Verwendung für IP möglich, aber nicht maßgeschneidert wie OSPF
Heute wird IS-IS noch von großen Providern eingesetzt

Eigenschaften
- Link-State-Protokoll
- Funktionalität ähnlich wie OSPF

Vorteile, die Befürworter nennen:
- Bessere Handhabung in sehr großen Areas
- Robusteres Protokoll für das Fluten
- Teilweise qualitativ bessere Implementierung in manchen Routern

Derzeit: Weiterentwicklung und Verbesserungen für den Einsatz mit IP in der IS-IS Working Group der IETF

Bild: IS-IS

Border Gateway Protocol (BGP-4)

BGP (Border Gateway Protocol) ist ein Protokoll für das Routing zwischen autonomen Systemen. Es wurde im Jahre 1989 eingeführt und seitdem mehrfach verbessert. Die aktuelle BGP Version 4 (BGP-4, RFC 1771) ist seit 1993 im Einsatz. Das Wesentliche beim BGP-4 ist die Unterstützung des CIDR-Konzeptes (Classless Interdomain Routing, RFC 1519). Um dies zu erreichen, wird das Netz-Präfix beim BGP-4 übermittelt. BGP-4 ermöglicht auch die Aggregation von Routen. Es löst das ältere EGP (Exterior Gateway Protocol, RFC 904) ab. BGP ist ein komplexes Routing-Protokoll. Es erfordert leistungsfähige Router und erhebliche Bandbreite für die Übertragung der Routing-Tabellen. Es wird deshalb von ISPs eingesetzt und von Firmen, die ein Netz aus mehreren AS betreiben. Firmennetze mit mehreren Zugängen zu ISPs können ebenfalls BGP nutzen.

- **Beziehung zwischen Routern und AS**
- **Nachrichtenaustausch**

- **Message Header**
- **Open Message**
- **Update Message**
- **Notification Message**
- **Keepalive Message**

- **Austausch von NLRI**
- **Pfad-Attribute**
- **Austausch von Withdrawn-Routes**

AS: Autonomous System
NLRI: Network Layer Reachability Information

Bild: BGP: Border Gateway Protocol

Die BGP-4-Router senden ihre Nachrichten mit Hilfe des verbindungsorientierten Protokolls TCP über den Port 179, wodurch Informationen auch über fehleranfällige Verbindungen sicher ausgetauscht werden können. Zwischen zwei benachbarten BGP-Routern wird daher für den RI-Austausch eine TCP-Verbindung aufgebaut. Die beiden benachbarten Router werden als Peers bzw. als BGP-Speaker bezeichnet. Die Verbindung zwischen Peers nennt man auch Peer-Verbindung.

Beim BGP-4 senden die Router eine Liste der autonomen Systeme, die auf dem entsprechenden Pfad zu einem Ziel liegen und von Nachbar-Routern sortiert und ausgewertet werden

Pfad-Vektor-Protokoll
- Erweiterung zum Distanz-Vektor
- BGP verbreitet keine Metriken wie Kosten etc. sondern Pfade
- Pfade garantieren Schleifenfreiheit
- Policies sind hier ausschlaggebend für die Wegewahl

Kommunikation zwischen BGP-Instanzen (gesichert durch TCP)

OPEN
- Aufbau einer Verbindung zum Kommunikationspartner Authentisierung

UPDATE
- Bekanntgabe eines neuen Pfads oder Zurücknahme eines alten Pfads

KEEP ALIVE
- Hält Verbindung aufrecht in Abwesenheit von UDPATE-Dateneinheiten Quittung zu einem OPEN-Request

NOTIFICATION
- Mitteilung von Fehlern in vorangegangenen Dateneinheiten Abbau einer Verbindung

Bild: Inter-AS-Routing: BGP

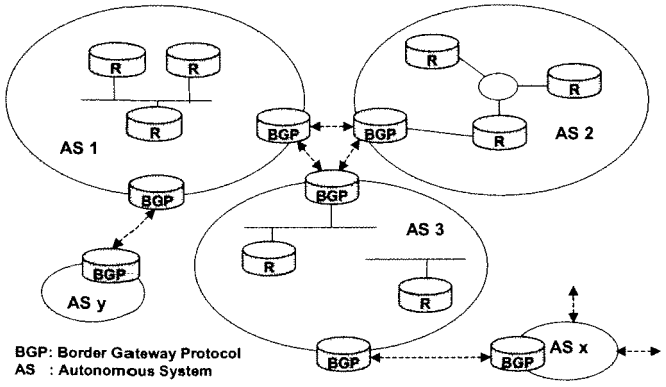


Bild: BGP-Router und Autonome Systeme

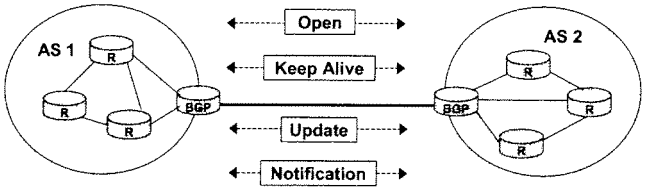
BGP kann auch innerhalb eines autonomen Systems für die Kommunikation zwischen zwei AS-Grenzen-Routern (d.h. zwischen ASBR) eingesetzt werden. Somit kann eine Peer-Verbindung sowohl innerhalb eines AS als auch zwischen unterschiedlichen AS aufgebaut werden.

In diesem Zusammenhang bezeichnet man das BGP
- als internes BGP,
- als externes BGP.

Die Peer-Verbindung zwischen zwei Routern, die zu unterschiedlichen AS gehören, wird als external Link (externe Verbindung) bezeichnet. Dagegen nennt man die Peer-Verbindung innerhalb eines AS internal Link (interne Verbindung).

Das Protokoll BGP-4 wird zwischen zwei AS-Grenzen-Routern ASBR (AS Border Router) eingesetzt. Ein ASBR macht mit Hilfe von BGP die Routen zu seinem AS bekannt. Hierfür muss sowohl jedes AS als auch jeder ASBR eine eindeutige Identifikation haben. Die Routing-Information zwischen Peers wird in Form von BGP-Nachrichten ausgetauscht. Die BGP-Peers bauen zuerst eine TCP-Verbindung für den Austausch der Routing-Infortnation. Dies bedeutet, dass ein TCP-Header jeder zu übertragenden BGP-Nachricht vorangestellt wird. Nachdem die TCP-Verbindung zwischen BGP-Peers aufgebaut wurde, wird zwischen ihnen eine BGP-Nachbarschaft verknüpft. Die Nachbarschaft zwischen BGP-Peers kann als die gegenseitige Bereitschaft, die Routing-Information zu tauschen, angesehen werden.

Um eine Nachbarschaft aufzubauen, sendet jeder Router eine BGP-Open-Nachricht, in der die Identifikation des eigenen autonomen Systems (MyAS) und des Absender-Routers angegeben wird. Hierbei stellt der BGP-Identifier in der Open-Nachricht eigentlich die Router-ID dar. Die Open-Nachricht wird von der Gegenseite mit der BGP-Keep-Alive-Nachricht bestätigt. Während des Aufbaus der Nachbarschaft stellen sich die beiden Router gegenseitig vor.



• OPEN:	Aufbau einer TCP-Verbindung zum Austausch von Routing-Informationen zwischen zwei BGP-Routern
• UPDATE	enthält Pfad-Attribute und Liste erreichbarer Netze
• NOTIFICATION:	Mitteilung von Fehlern
• KEEP ALIVE:	Erreichbarkeitskontrolle der Nachbar-BGP-Router

Bild: BGP Message Flow

Nach dem Aufbau der Nachbarschaft kündigen die BGP-Peers die ihnen bekannten Routen zu ihren autonomen Systemen mittels Update-Nachrichten an. Der Name Update ist hier damit zu begründen, dass es sich im Laufe der Zeit überwiegend um die Aktualisierungen (Updates) von Routing-Tabellen handelt.

Jede Update-Nachricht enthält die Identifikation des Quell-AS (als MyAS) und die Angabe der aggregierten Route zum AS. Die Update-Nachricht enthält u.a. eine Liste von Netzzielen in der Form <Länge, Präfix>, die über den Absender-Router erreicht werden können.

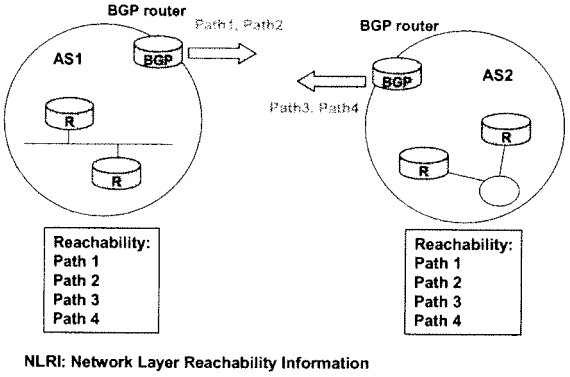


Bild: BGP Austausch von NLRI

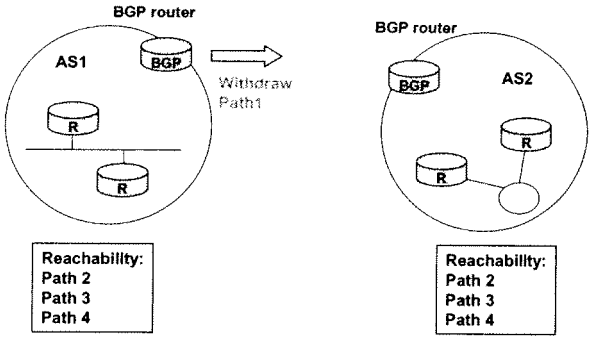
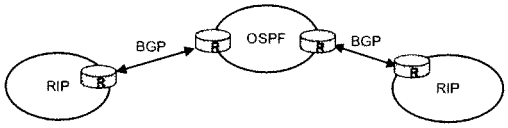


Bild: BGP Austausch von Withdrawn Routes

Falls sich die Lage im AS ändert, z.B. ein Subnetz plötzlich nicht erreichbar ist oder eine bessere Route zur Verfügung steht, informiert ein BGP-Router mit sogenannten NLRI (Network Layer Reachability Information) seinen Nachbarn darüber, dass die ungültig gewordene Route (Withdrawn Route) zurückgezogen und eventuell durch die neue Route ersetzt bzw. vollkommen entfernt werden soll. Die ungültig gewordenen Routen können in einer Update-Nachricht angegeben werden.



- OSPF und RIP nur innerhalb der einzelnen Domänen (begrenzte Skalierbarkeit !)
- BGP als separates Routing-Protokoll zwischen Domänen (= autonome Systeme, AS)
  - Inter-Domain-Routing-Protokoll
  - Distanz-Vektor-Prinzip
  - Skalierbarkeit durch Hierarchiebildung
  - Berücksichtigung von administrativen Beschränkungen (Policies)

Bild: Border Gateway Protocol (BGP)

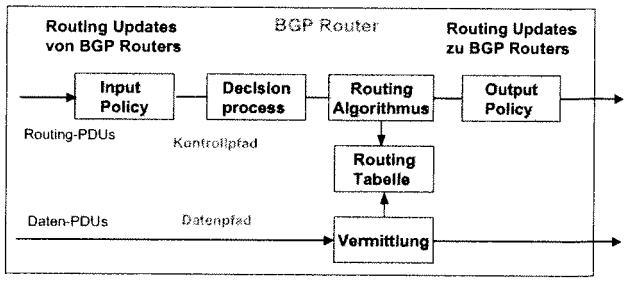


Bild: BGP Process and Routing Policies

Pfad-Attribute

Die Pfad-Attribute (Path Attributes) werden verwendet, um die Eigenschaften von Routen näher spezifizieren zu können. Ein Pfad-Attribut ist folgendes Triplet <Attribut-Typ, Attribut-Länge, Attribut-Wert>.

Der Attribut-Typ ist ein zwei Byte langes Feld, bei dem das erste Byte die Attribut-Kategorie und das zweite Byte den Attribut-Code darstellt. Es sind folgende Attribut-Kategorien zu unterscheiden:

- **Well-known mandatory:** Dieses Attribut muss in der Update-Nachricht vorhanden sein. Es muss von allen BGP-4-Implementierungen erkannt sein.
- **Well-known discretionary:** Dieses Attribut muss von allen BGP-4-Implementierungen erkannt und kann optional in der Update-Nachricht enthalten sein.
- **Optional transitive:** Es handelt sich um ein optionales Attribut. Falls es durch eine BGP-4-Implementierung nicht erkannt wird, sollte es nicht ignoriert, sondern an andere BGP-Router eventuell weitergeleitet werden.
- **Optional non-transitive:** Es handelt sich um ein optionales Attribut. Falls es durch eine BGP-4 Implementierung nicht erkannt wird, sollte es ignoriert und an keinen anderen BGP-Router weitergeleitet werden.

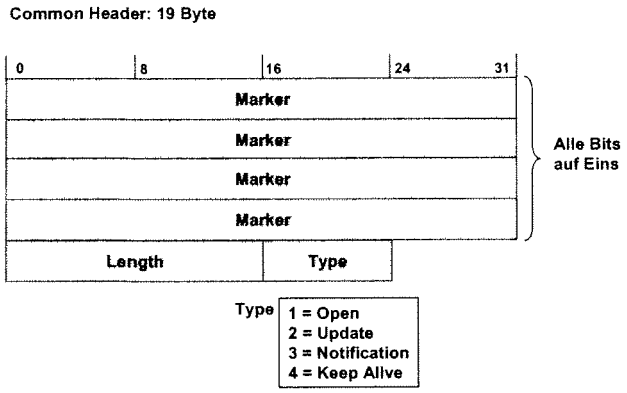


Bild: BGP: Nachrichten-Header

Jede BGP-4-Nachricht setzt sich aus einem gemeinsamen BGP-Header, der 19 Bytes lang ist, und einem Nachrichtenteil zusammen. Die BGP-Nachrichten können maximal 4098 Bytes haben. Die kleinste ist die BGP-Keep-Alive-Nachricht. Sie besteht nur aus dem BGP-Header und ist daher nur 19 Bytes lang.

Der BGP-Header enthält folgende Angaben:

- **Marker:** Dieses Feld dient zur gegenseitigen Authentisierung der BGP-Peers während der Nachbarschaft. Falls es sich um die Open-Nachricht handelt d.h. es besteht noch keine Nachbarschaft -, sind alle Marker-Bits gleich 1. In anderen BGP-Nachrichten wird der Marker durch einen Teil der Authentisierungsdaten festgelegt.
- **Length:** Hier wird die Länge in Bytes der ganzen BGP-Nachricht (inkl. Header) angegeben.
- **Type:** Hier wird der Typ (d.h. die Bedeutung) der Nachricht festgelegt.

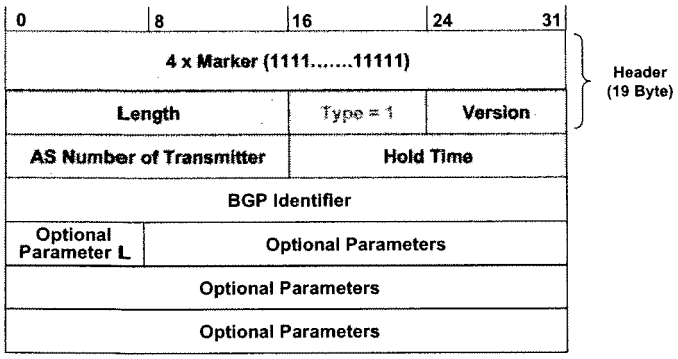


Bild: BGP: Open-Nachricht

Open-Nachricht

Eine wichtige Funktion des Protokolls BGP-4 besteht im Aufbau von Nachbarschaften zwischen BGP-Peers. Dies ist die Voraussetzung für den Austausch der Routing-Information. Falls eine TCP-Verbindung zwischen den BGP-Peers bereits besteht, kann die Nachbarschaft mit Hilfe von Open-Nachrichten aufgebaut werden.

Die einzelnen Angaben innerhalb der Open-Nachricht haben folgende Bedeutung:

- **Version (von BGP):** Dieses Feld enthält 4, d.h. es handelt sich um das BGP-4.
- **My Autonomous System (MyAS):** Hier wird die Identifikation (Nummer) des Quell-AS angegeben.
- **Hold Time:** Die maximale Zeitspanne in Sekunden, die zwischen dem Empfang von darauffolgenden Keep-Alive- - bzw. Update-Nachrichten verstreichen darf. Das ist die maximale Wartezeit auf eine neue Keep-Alive- bzw. Update-Nachricht vom Nachbar-Router. Nach Ablauf dieser Zeit wird der Nachbar-Router für ausgefallen erklärt.
- **BGP Identifier:** Hier wird die Identifikation des Absender-Routers angegeben. Somit handelt es sich um die Router-ID.
- **Optional Parameter Length:** Hier wird die Länge von optionalen Parametern angezeigt. Der Wert 0 weist darauf hin, dass keine optionalen Parameter vorhanden sind.

- **Optional Parameters:** Hier werden die optionalen Parameter angegeben. Jeder Parameter wird in Form des Triplets: <Parametertyp (1 Byte), Parameterlänge (1 Byte), Parameterwert> repräsentiert. Der optionale Parameter vom Typ 1 ist Auhentification Information.

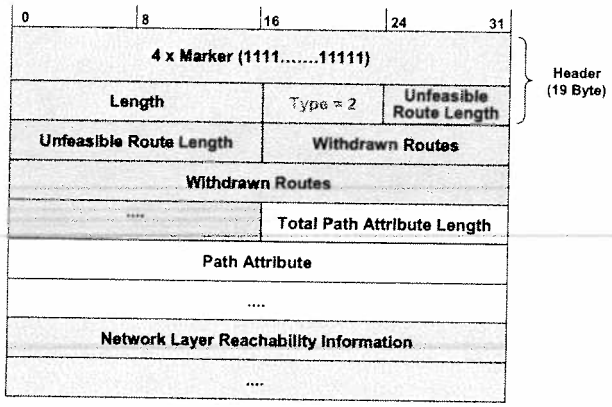
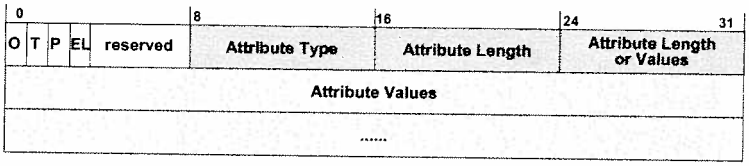


Bild: BGP: Update-Nachricht

Update-Nachricht

Nach der Übermittlung der Open-Nachricht werden zunächst die gesamten Routing-Informationen mittels Keep-Alive-Nachrichten zwischen den BGP-Peers ausgetauscht. Die Änderungen, die im Laufe der Zeit im Netz auftreten (z.B. ein neues Subnetz wurde eingerichtet), werden durch die Übermittlung von Update-Nachrichten bekanntgemacht.



O : Optional
T : Transitive
P : Partial
EL: Extended Length

Bild: BGP: Path Attributes

Die Update-Nachricht besteht aus den folgenden Angaben:

- **Unfeasible Route Length:** Hier wird die Länge in Bytes des nächsten Felds Withdraw Routes angezeigt. Es handelt sich hier um die Länge des Feldes, in dem die ungültig gewordenen (zurückgezogenen) Routen aufgelistet werden.
- **Withdraw Routes:** Hier werden die ungültig gewordenen Routen aufgelistet. Diese Routen müssen aus der Routing-Tabelle entfernt werden. Withdraw Routes werden durch das Tupel <Länge, Präfix> repräsentiert. Das Tupel <17, 131.42.128.0> bedeutet beispielsweise, dass die Route 131.42.128.0/17 (im CIDR-Format) zurückgezogen werden soll.
- **Total Path Attribute Length:** Hier wird die Länge des nächsten Felds Path Attributes angegeben.
- **Path Attributes (Pfad-Attribute):** In diesem Feld sind die routenspezifischen Informationen (sog. Pfad-Attribute) enthalten. Ein Pfad-Attribut ist ein Triplet der Form <Attribut-Typ, Attribut-Länge, Attribut-Wert>.
- **NLRI (Network Layer Reachability Information):** Die NLRI ist ein Mechanismus zur Unterstützung von CIDR (Classless Interdomain Routing). Das NLRI-Feld enthält eine Liste der Netzziele, über die ein BGP-Router seinen Nachbar-Router informieren möchte. Das NLRI-Feld besteht aus mehreren NLRI-Instanzen der Form <Length, Prefix>.

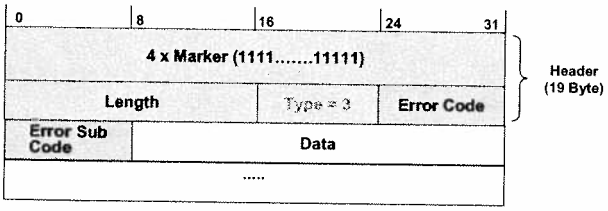


Bild: BGP: Notification-Nachricht

Notification-Nachricht

Ein Router sendet eine BGP4-Notification-Nachricht, um seinem Nachbar-Router eine Fehlermeldung zu signalisieren. Sie wird immer nach der Entdeckung eines Fehlers verschickt und kann zum Abbruch der Peer-Verbindung (Nachbarschaft) führen. Falls ein Router eine Verbindung abbauen möchte, sendet er die Notification-Nachricht, in der er gleichzeitig den Grund für den Abbau der Verbindung angibt.

Error Code	Error Subcode (Beispiele)
1: Message header error	1: Connection not synchronized 2: Bad message length
2: OPEN message error	1: Unsupported version number 2: Bad peer AS 3: Bad BGP identifier
3: UPDATE message error	1: Malformed attribute list 2: Unrecognised well-known
4: Hold timer expired	Kein subcode
5: Finite state machine error	Kein subcode
6: Cease (Beenden)	Kein subcode

Bild: BGP: Notification-Nachricht (Error Codes)

Eine Notification-Nachricht besteht aus dem BGP-Header, der Fehlerangabe (Error Code und Error Subcode) und aus einem variablen Feld Data, in dem der Fehler gegebenenfalls weiter beschrieben werden kann. Der Error Code verweist auf den Fehlertyp. Mit dem Error-Subcode wird der Fehler näher spezifiziert.

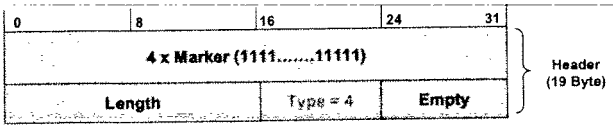


Bild: BGP: Keep-Alive-Nachricht

KEEP-ALIVE-Nachricht

Die BGP4-Keep-Alive-Nachricht besteht nur aus einem 19 Byte langen BGP-4-Header und enthält keine weiteren Angaben. Keep-Alive wird u.a. als eine Bestätigung verwendet.

Falls keine neuen Routing-Informationen (keine Veränderungen) vorliegen, werden die Keep-Alive-Nachrichten zwischen den BGP-Peers periodisch gesendet, um der Gegenseite die Funktionsbereitschaft zu signalisieren. Dadurch lässt sich feststellen, ob der Nachbar-Router erreichbar ist.

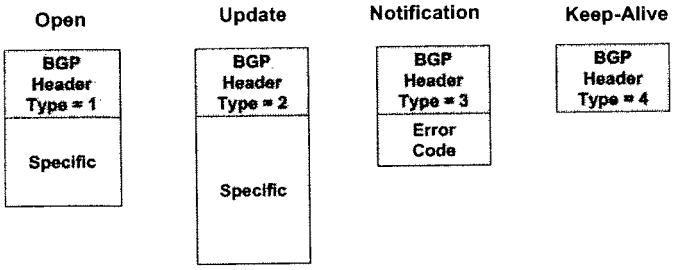


Bild: BGP-Nachrichten

Routing-Algorithmen

Die beiden Verfahren Distanz-Vektor-Routing und Link-State-Routing sind in der Praxis sehr wichtig.

- Beim Distanz-Vektor-Verfahren wird ein kürzester Weg gewählt, der die kleinstmögliche Anzahl von Zwischensystemen (hops) enthält.
- Beim Link-State-Verfahren ist jeder Teilstrecke (link) ein Gewicht nach einer festgelegten Metrik (z. B. Kosten, Distanz, Bandbreite, Auslastung) zugeordnet. Ein kürzester Weg ist dadurch gekennzeichnet, dass die Summe der Gewichte minimal ist.

Distanz-Vektor-Algorithmus

Der Distanz-Vektor-Algorithmus (auch als **Bellman-Ford-Algorithmus** bekannt) wird für die verteilte Berechnung von Routing-Tabellen verwendet. Dabei berechnet zunächst jeder Knoten für sich eine Routing-Tabelle. Die Einträge sind Tupel, die - wie der Name des Verfahrens andeuten soll - je eine Adresse (Vektor) und die zugehörige Distanz enthalten. Die Tupel werden den benachbarten Knoten mitgeteilt und zur Aktualisierung (update) deren Routing-Tabellen genutzt. Nach einiger Zeit (Konvergenzdauer) besitzen alle Knoten optimale Routing-Tabellen. Das Distanz-Vektor-Verfahren wird periodisch im Abstand weniger Sekunden durchgeführt. Damit kann der Ausfall einzelner Knoten oder Kanten (Übertragungsstrecken) berücksichtigt werden. Ein Knoten, der keine periodische Routing-Information liefert, gilt als ausgefallen. Die umgebenden Knoten ändern daraufhin ihre Routing-Tabellen so, dass der ausgefallene Knoten umgangen wird, soweit bestehende Pfade dies zulassen.

Beim Bellman-Ford-Algorithmus ist jede Kante des Graphen mit einem Gewicht belegt, die Distanz zum Ziel ist als Summe der Gewichte auf dem Weg zum Ziel definiert. Die Routing-Tabelle enthält für jeden Eintrag ein Feld, das die Distanz zum Zielknoten (auf einem Pfad entsprechend dem angegebenen Next Hop) enthält. Jeder Knoten sendet die ihm bekannten Wertepaare (Ziel, Distanz) an seine Nachbarn. Wenn ein Knoten eine Nachricht von seinem Nachbarn erhält, prüft er alle Einträge und ändert seine Routing-Tabelle, falls der Nachbar einen kürzeren Pfad zu einem Ziel kennt.

Ein Vorteil des Distanz-Vektor-Algorithmus ist seine Einfachheit. Der größte Nachteil liegt in der langen Konvergenzdauer. Dies führt dazu, dass bei raschen Änderungen der Topologie Inkonsistenzen in den Routing-Tabellen entstehen, die zu großen Verzögerungen und zu Paketverlusten führen können.

Link-State-Routing, Dijkstra-Algorithmus

Das Link-State-Routing (auch Link-Status-Routing) wird auch als **SPF-Routing** (Shortest Path First) bezeichnet, obwohl andere Routing-Verfahren ebenfalls kürzeste Pfade ermitteln. Das Verfahren beinhaltet - wie das Distanz-Vektor-Verfahren - eine verteilte Berechnung des Routing. Die zwischen den Knoten ausgetauschten Nachrichten beinhalten den Status einer Verbindung zwischen zwei Knoten, der durch ein Gewicht in einer bestimmten Metrik ausgedrückt wird. Diese Nachrichten werden per Broadcast an alle Knoten gesendet. Damit besitzt jeder Knoten die globale und vollständige Zustandsinformation über das Netz. Jeder Knoten kann nun für sich einen Graphen für das Netz erstellen. Anschließend berechnet jeder Knoten seine Routing-Tabelle mit Hilfe des Dijkstra-Algorithmus. Das Routing kann also - wie beim Distanz-Vektor-Verfahren - an den aktuellen Zustand des Netzes adaptiert werden. Die Adaption findet schneller statt, da alle Knoten gleichzeitig über Statusänderungen informiert werden.

Die Grundidee des Link-State-Routing geht davon aus, dass zunächst die Topologie des Netzes ermittelt wird. Dazu sind die folgenden Schritte notwendig:

- Jeder Router kümmert sich selbst darum, seine Nachbarn und ihre Namen kennen zu lernen.
- Jeder Router bildet ein LSP (Link State Packet) mit den Namen seiner Nachbarn und den Gewichten der zugehörigen Links.
- Die LSP werden an alle Router verschickt, jeder Router puffert die zuletzt erhaltenen LSP aller anderen Router.
- Damit kennt jeder Router die vollständige Topologie des Netzes. Dies ermöglicht den einzelnen Routern die Berechnung von Pfaden zu jedem Ziel.

Der **Dijkstra-Algorithmus** wird nun zur Berechnung der kürzesten Wege ausgeführt. Der Algorithmus geht von einem bestimmten Knoten, dem Quellenknoten, aus und berechnet eine Routing-Tabelle für diesen Knoten. In der Routing-Tabelle sind für alle möglichen Zielknoten die nächsten Knoten, die in Richtung auf den Zielknoten zu durchlaufen sind, und die Distanz D von jedem Knoten zum Quellenknoten enthalten. Für jeden Knoten im Netz ist eine Routing-Tabelle zu ermitteln. Für den Dijkstra-Algorithmus sind nebst der Beschreibung des Graphen einige Datenstrukturen erforderlich. Die Knoten werden von 1 bis n nummeriert, damit kann die Knotennummer als Index zum Datenzugriff verwendet werden. D ist ein Vektor, dessen i-te Komponente den aktuellen Wert der kürzesten Distanz vom Quellenknoten zum Knoten i enthält. Die i-te Komponente des Vektors R puffert den nächsten Knoten (next hop), der auf dem Weg zum Knoten i zu durchlaufen ist. Die Menge S der noch zu untersuchenden Knoten kann als doppelt verkettete Liste von Knotennummern gepuffert werden.

Der Dijkstra-Algorithmus lässt verschiedene Metriken zu. Im einfachsten Fall ist die Distanz gleich der Anzahl der durchlaufenen Zwischensysteme. Dazu werden alle Gewichte auf den Wert 1 gesetzt. In WANs kann die Bandbreite eines Link als Gewicht sinnvoll sein. Gewichte können auch die Ansicht des Netzadministrators (administrative policy) zu bevorzugten Pfaden widerspiegeln.

Pfad-Vektor-Algorithmus

Das Pfad-Vektor-Verfahren ist dem Distanz-Vektor-Verfahren ähnlich. Statt der Distanz zum Ziel wird jedoch ein Pfad zum Ziel angegeben. Dieser enthält eine Sequenz der zu durchlaufenden Routing-Bereiche (sogenannte Autonomous Systems, (AS). Das Verfahren BGP (Border Gateway Protocol) ist ein Pfad-Vektor-Protokoll, das im Internet verwendet wird. Es verwendet keine Metriken. Durch die vordefinierten Pfade wird ein Policy-based routing realisiert, das es dem Netzbetreiber erlaubt, bestimmte Pfade auszuschließen oder uninteressant zu machen.

Dijkstra's Shortest Path Algorithmus

- Realisierung durch Forward Search Algorithmus
- Jeder Router hält 2 Listen:
  Confirmed und Tentative mit Einträgen der Form {Ziel, Kosten, nächster Knoten}

- Algorithmus
  1. Initialisierung von Confirmed mit eigenem Eintrag (Kosten 0)
  2. Selektiere den zuletzt in Confirmed aufgenommenen (Next)
  3. Berechne die Kosten zu allen Nachbarn von Next:
     Kosten = Kosten zu Next + Kosten von Next zu Nachbar
     a. Falls Nachbar nicht in Confirmed:
        Aufnahme des Nachbars in Tentative
     b. Falls Nachbar in Tentative: Ersetzen des Eintrags,
        falls die Kosten niedriger als die des Eintrags sind.
  4. Stop, falls Tentative leer ist, sonst: übernehme in Confirmed
     den Eintrag mit den geringsten Kosten aus Tentative,
     gehe zu 2

Bild: Dijkstra's Shortest Path Algorithmus

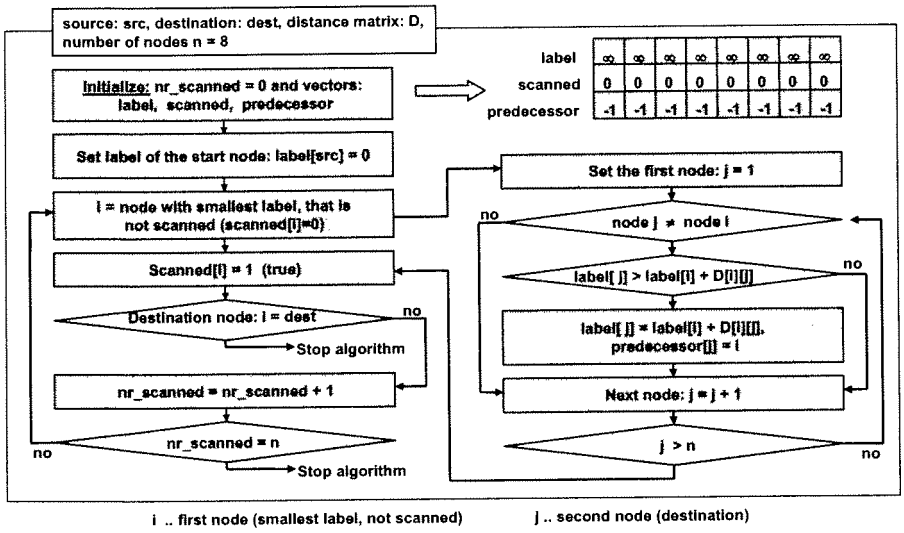


Bild: Shortest Path Algorithm (Dijkstra)

```
#define MAX_NODES 1020                  /* maximum number of nodes */
#define INFINITY 1000000000             /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];      /* dist[i][j] is the distance from i to j */

void shortest_path (int s,int t, int path[])
{struct state {                                 /* the path being worked on */
        int predecessor;                        /* previous node */
        int length;                             /* length from source to this node */
        enum {permanent, tentative} label;      /* label state*/
} state[MAX_NODES];
```

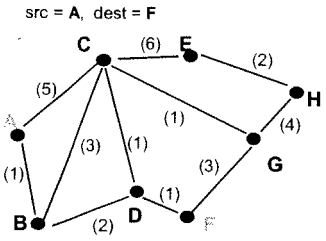
```
int i, k, min;
struct state *p;
for (p =& state[0];
p < & state[n] ;p++) {          /* initialize state */
        p  > predecessor = -1;
        p -> length = INFINITY;
        p -> label = tentative;
} state[t].length = 0;  state[t].label = permanent;
k = t;                          /* k is the initial working node */
```

Bild: Dijkstra's Shortest Path Algorithmus

```
do{                                             /* is there a better path from k? */
   for (i = 0; i < n; i++)                      /* this graph has n nodes */
      if (dist[k][i] != 0 && state[i].label == tentative) {
         if (state[k].length + dist[k][i] < state[i].length) {
            state[i].predecessor = k;
            state[i].length = state[k].length + dist[k][i];
         }
      }
   /* Find the tentatively labeled node with the smallest label. */
   k = 0; min = INIFINITY;
   for (i = 0; i < n; i++)
      if (state[i].label == tentative && state[i].length < min {
         min = state[i].length;
         k = i;
      }
   state[k].label = permanent;
} while (k != s);

i = 0; k = s;           /* copy the path into the output array. */
do {path[i++] = k; k = state[k].predecessor;  } while (k  >= 0);
}
```

Bild: Dijkstra's Shortest Path Algorithmus

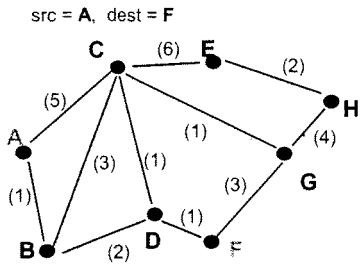


1.

	A	B	C	D	E	F	G	H
label	0	∞	∞	∞	∞	∞	∞	∞
scanned	0	0	0	0	0	0	0	0
predecessor	-1	-1	-1	-1	-1	-1	-1	-1

smallest not scanned label (0) is scanned next
predecessor and label are replaced if the new label is smaller

Bild: Dijkstra's Shortest Path Algorithmus (1)



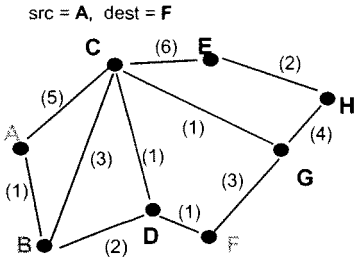
1.

	A	B	C	D	E	F	G	H
label	0	∞	∞	∞	∞	∞	∞	∞
scanned	0	0	0	0	0	0	0	0
predecessor	-1	-1	-1	-1	-1	-1	-1	-1

2.

	A	B	C	D	E	F	G	H
label	0	1	5	∞	∞	∞	∞	∞
scanned	1	0	0	0	0	0	0	0
predecessor	-1	A	A	-1	-1	-1	-1	-1

Bild: Dijkstra's Shortest Path Algorithmus (2)



1.

	A	B	C	D	E	F	G	H
label	0	∞	∞	∞	∞	∞	∞	∞
scanned	0	0	0	0	0	0	0	0
predecessor	-1	-1	-1	-1	-1	-1	-1	-1

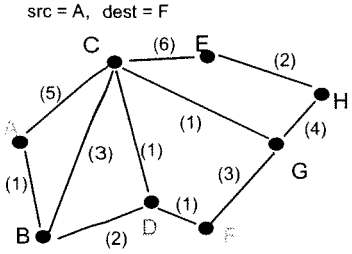
2.

	A	B	C	D	E	F	G	H
label	0	1	5	∞	∞	∞	∞	∞
scanned	1	0	0	0	0	0	0	0
predecessor	-1	A	A	-1	-1	-1	-1	-1

3.

	A	B	C	D	E	F	G	H
label	0	1	4	3	∞	∞	∞	∞
scanned	1	1	0	0	0	0	0	0
predecessor	-1	A	B	B	-1	-1	-1	-1

Bild: Dijkstra's Shortest Path Algorithmus (3)



1.

	A	B	C	D	E	F	G	H
label	0	∞	∞	∞	∞	∞	∞	∞
scanned	0	0	0	0	0	0	0	0
predecessor	-1	-1	-1	-1	-1	-1	-1	-1

2.

	A	B	C	D	E	F	G	H
label	0	1	5	∞	∞	∞	∞	∞
scanned	1	0	0	0	0	0	0	0
predecessor	-1	A	A	-1	-1	-1	-1	-1

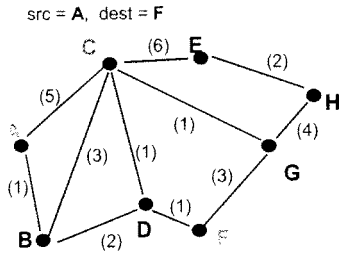
4.

	A	B	C	D	E	F	G	H
label	0	1	4	3	∞	4	∞	∞
scanned	1	1	0	1	0	0	0	0
predecessor	-1	A	B	B	-1	D	-1	-1

3.

	A	B	C	D	E	F	G	H
label	0	1	4	3	∞	∞	∞	∞
scanned	1	1	0	0	0	0	0	0
predecessor	-1	A	B	B	-1	-1	-1	-1

Bild: Dijkstra's Shortest Path Algorithmus (4)



1.

	A	B	C	D	E	F	G	H
label	0	∞	∞	∞	∞	∞	∞	∞
scanned	0	0	0	0	0	0	0	0
predecessor	-1	-1	-1	-1	-1	-1	-1	-1

2.

	A	B	C	D	E	F	G	H
label	0	1	5	∞	∞	∞	∞	∞
scanned	1	0	0	0	0	0	0	0
predecessor	-1	A	A	-1	-1	-1	-1	-1

3.

	A	B	C	D	E	F	G	H
label	0	1	4	3	∞	∞	∞	∞
scanned	1	1	0	0	0	0	0	0
predecessor	-1	A	B	B	-1	-1	-1	-1

4.

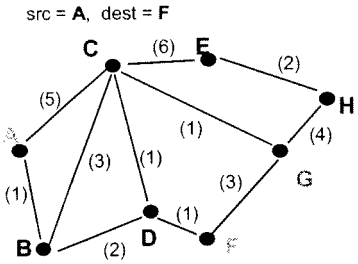
	A	B	C	D	E	F	G	H
label	0	1	4	3	∞	4	∞	∞
scanned	1	1	0	1	0	0	0	0
predecessor	-1	A	B	B	-1	D	-1	-1

5.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	∞
scanned	1	1	1	1	0	0	0	0
predecessor	-1	A	B	B	C	D	C	-1

F has the smallest not scanned label!

Bild: Dijkstra's Shortest Path Algorithmus (5)

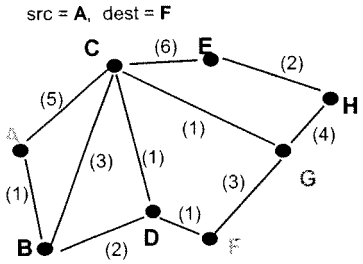


6.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	∞
scanned	1	1	1	1	0	1	0	0
predecessor	-1	A	B	B	C	D	C	-1

Complete table for source A

Bild: Dijkstra's Shortest Path Algorithmus (6)



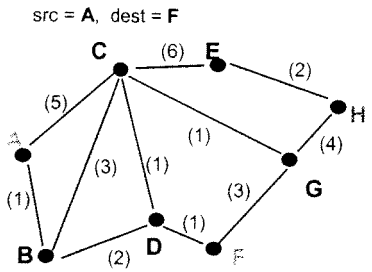
6.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	∞
scanned	1	1	1	1	0	1	0	0
predecessor	-1	A	B	B	C	D	C	-1

7.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	9
scanned	1	1	1	1	0	1	1	0
predecessor	-1	A	B	B	C	D	C	G

Bild: Dijkstra's Shortest Path Algorithmus (7)



6.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	∞
scanned	1	1	1	1	0	1	0	0
predecessor	-1	A	B	B	C	D	C	-1

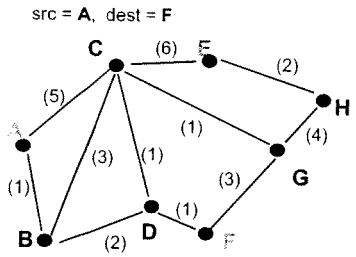
7.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	9
scanned	1	1	1	1	0	1	1	0
predecessor	-1	A	B	B	C	D	C	G

8.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	9
scanned	1	1	1	1	0	1	1	1
predecessor	-1	A	B	B	C	D	C	G

Bild: Dijkstra's Shortest Path Algorithmus (8)



6.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	∞
scanned	1	1	1	1	0	1	0	0
predecessor	-1	A	B	B	C	D	C	-1

7.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	9
scanned	1	1	1	1	0	1	1	0
predecessor	-1	A	B	B	C	D	C	G

9.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	9
scanned	1	1	1	1	1	1	1	1
predecessor	-1	A	B	B	C	D	C	G

8.

	A	B	C	D	E	F	G	H
label	0	1	4	3	10	4	5	9
scanned	1	1	1	1	0	1	1	1
predecessor	-1	A	B	B	C	D	C	G

Shortest path:FDBA ⟶ ABDF

Bild: Dijkstra's Shortest Path Algorithmus (9)

3.2c Internet-Referenzmodell: Internetschicht – MPLS, QoS

Version: Dez: 2003

Inhalt

- MPLS (Multi Protocol Label Switching): Netzstruktur, Vermittlungsformat, Vermittlungspfade
- QoS (Quality-of-Service, Servicequalität)
- Integrated Services
- RSVP (Resource Reservation Protocol)
- Differentiated Services

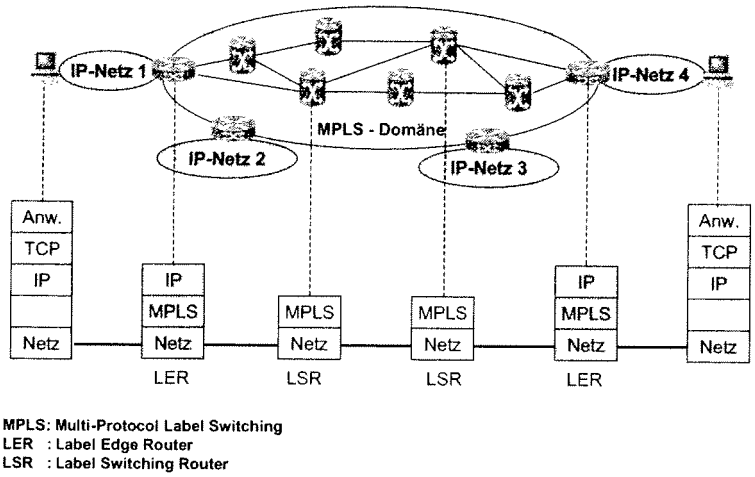


Bild: IP-Vernetzung mit MPLS

MPLS (Multi-Protocol Label Switching) ist ein Verfahren zur Forwarding von IP-Paketen. Dies erfolgt nicht mehr auf Basis deren IP-Adresse, sondern jedem Paket wird zusätzlich ein **Label** (Marke) mitgegeben. Ein Label ist ein kurzer, nicht weiter strukturierter Identifier fester Länge, der von einem Router sehr schnell ausgewertet werden kann. Dadurch lässt sich der Router-Durchsatz (Anzahl der pro Zeiteinheit weitergereichten Pakete) wesentlich erhöhen. Die Umformung von IP- zu MPLS-Paketen und umgekehrt geschieht am Rand der MPLS-Domäne in den sogenannten Label Edge Routern (LER). Im Innern sorgen Label Switching Router (LSR) für die Weiterleitung der MPLS-Pakete.

- A label is a short, fixed length, locally significant identifier which is used to identify a FEC

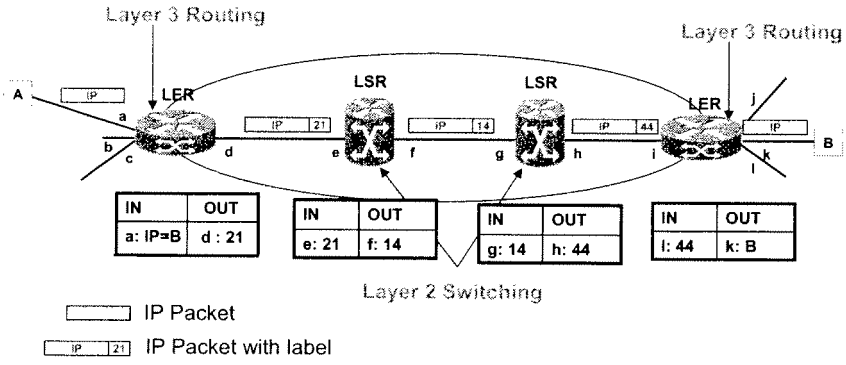


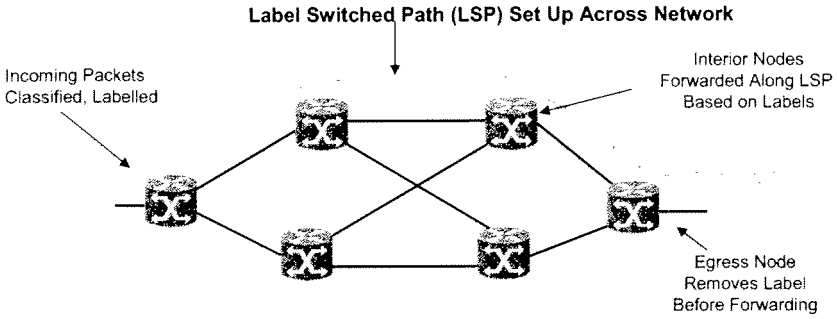
Bild: MPLS Concepts

Im Gegensatz zu einer IP-Adresse hat ein Label nur lokale Bedeutung für eine Teilstrecke (Link). Das Forwarding geschieht durch Label Swapping, d. h. der Router trägt für die nächste Teilstrecke ein neues, passendes Label ein. Die Labels werden von den Routern lokal ermittelt und den Nachbarn mitgeteilt. Basis für die Ermittlung der Labels ist nach wie vor das IP-Routing, das mit den üblichen Verfahren durchgeführt wird. MPLS lässt sich in das OSI-Modell nicht genau einordnen, seine Funktion ist jedoch mit den Schichten 2 und 3 verbunden.

Multiprotocol Label Switching (MPLS)

MPLS stellt ein Verfahren dar, um IP-Pakete u.a. in Frame-Relay- und ATM-Netzen effektiv übermitteln zu können. Auch die Übermittlung der IP-Pakete in zukünftigen optischen Netzen auf Basis der WDM-Technik (Wavelength Division Multiplexing) wird mit MPLS verlaufen. Nach dem MPLS-Konzept wird jedem zu übertragenden Paket ein Label vorangestellt. Anhand von Labeln können IP-Pakete in Netzknoten effizient weitergeleitet werden, ohne dabei den komplexen IP-Header auswerten zu müssen. Das MPLS-Konzept kann als eine Art IP-Hardware-Switching interpretiert werden.

Beim MPLS werden zwei Arten von sog. Label Switching Routern (LSR) definiert, nämlich Edge-LSR (E-LSR) am Rande und Core-LSR (C-LSR) im Kernbereich des Netzes. Die Router sind über permanente logische Verbindungen vernetzt, so dass ein logisches Netz entsteht, in dem die C-LSR als Knoten und die E-LSR als Endkomponenten dienen. Ein solches Netz stellt ein logisches Routing-Netz oberhalb der physikalischen Netzstruktur dar. Die E-LSR klassifizieren die zu übertragenden IP-Pakete und versehen sie mit Labeln. Die Netzknoten leiten die IP-Pakete anhand der Label weiter. Die Label-Informationen werden nach dem Protokoll LDP (Label Distribution Protocol) ausgetauscht.



Two types of Label Switched Paths:

- Hop-by-hop
- Explicit Routing

Bild: Das Konzept Label Switched Path

- **Label Edge Router (LER)**
  - Ingress LER examines inbound packets, classifies packet, adds MPLS header and assigns initial label
  - Egress LER removes the MPLS header and routes packet
  - Receives a "labelled" packet and routes the packet to the destination
  - Determines where and how a packet travels
  - Assigns a label
  - Passes the "labelled" packet to the next LSR/LER
- **Label Switching Router (LSR)**
  - Transit switch that forwards packets based on MPLS labels
  - Receives a "labelled" packet
  - Determines the next "hop" based on label
  - Assigns a new label
  - Passes the "labelled" packet to the next LSR/LER

Bild: LER & LSR

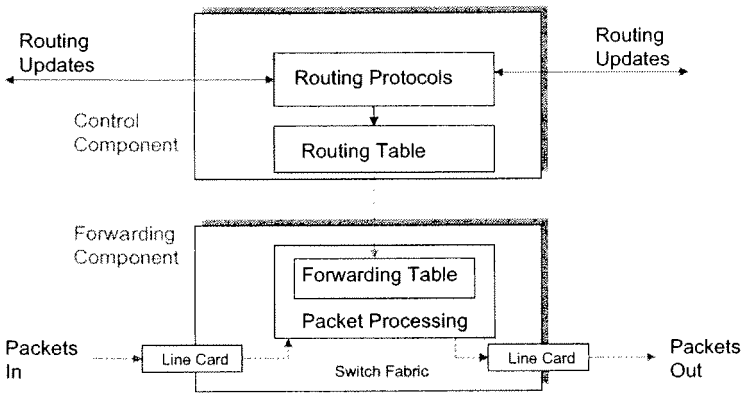


Bild: MPLS Architektur

Notwendigkeit und Idee von MPLS

Die klassischen IP-Netze wie z.B. das heutige Internet funktionieren nach dem Datagramm-Prinzip. Dies bedeutet, dass keine Verbindung für die Übermittlung der IP-Pakete zwischen den kommunizierenden Rechnern aufgebaut wird, sondern die einzelnen IP-Pakete in Routern (als Internet-Knoten) individuell nach der aktuellen Lage im Netz weitergeleitet werden. Aus diesem Grund bezeichnet man die klassischen IP-Netze auch als verbindungslos.

In verbindungslosen IP-Netzen werden die einzelnen IP-Pakete vom Quell- zum Zielrechner meist über unterschiedliche Wege transportiert. Infolgedessen sind die Verzögerungen von einzelnen IP-Paketen in der Regel unterschiedlich. Dies ist die Ursache dafür, dass es schwierig ist, die steigenden QoS-Anforderungen (Quality of Service) in verbindungslosen IP-Weitverkehrsnetzen zu erfüllen (RFC 3031).

Die QoS-Anforderungen lassen sich nur dann besser und einfacher erfüllen, wenn die zusammengehörenden (z.B. einer Dienst-Klasse) IP-Pakete im Netz zwischen zwei kommunizierenden Rechnern über den gleichen Weg übermittelt werden.

Um dies zu erreichen, muss zuerst eine virtuelle Verbin dung über das IP-Netz aufgebaut werden. Hierfür wurde gerade das Konzept MPLS entwickelt.

Idee von MPLS

Die Idee von MPLS besteht darin, dass zuerst ein Pfad als virtuelle Verbindung über das IP-Netz zwischen den kommunizierenden Rechnern für die Übermittlung der IP-Pakete aufgebaut wird. Dadurch werden die einzelnen IP-Pakete über die gleichen Netzknoten übermittelt. Dieses Prinzip entspricht vollkommen den virtuellen Verbindungen in sog. verbindungsorientierten Netzen mit Paketvermittlung (wie z.B. X.25-, Frame-Relay- bzw. ATM-Netze).

Um die IP-Pakete genauso wie z.B. Pakete in einem Frame-Relay- bzw. ATM-Netz zu übermitteln, müssen die IP-Pakete um eine spezielle Angabe ergänzt werden, die der Angabe eines logischen Kanals entspricht. Beim MPLS wird hierfür jedem zu übertragenden IP-Paket ein Zusatzfeld mit einem Label vorangestellt. Das Label kann als Identifikation des IP-Pakets angesehen werden. Anhand von Labeln können die IP-Pakete in den Netzknoten effizient nach den gleichen Prinzipien wie in Frame-Relay- bzw. ATM-Netzen weitergeleitet werden, ohne dabei den komplexen IP-Header auswerten zu müssen.

Beim MPLS werden sog. Label Switching Router (LSR) eingeführt. Sie können als Funktions-Module angesehen und softwaremäßig realisiert werden. Man unterscheidet zwischen:

- einem Edge-LSR (E-LSR),
- einem Core-LSR (C-LSR).

Die Funktion eines E-LSR wird in einem klassischen Router am Rande des Netzes untergebracht. Der E-LSR klassifiziert die zu übertragenden IP-Pakete und versieht sie mit. Labeln. Ein E-LSR wird manchmal auch als Label Edge Router (LER) bezeichnet und stellt einen MPLS-Randknoten (MPLS Edge Node) dar.

Ein C-LSR wird als Funktionsmodul im Netzknoten implementiert. Die Hauptaufgabe des C-LSR besteht in der Bestimmung von optimalen Routen und in der Verteilung der Label-Information nach dem Protokoll LDP (Label Distribution Protocol).

E-LSR und C-LSR werden über permanente logische Verbindungen miteinander vernetzt. Dadurch entsteht ein logisches MPLS-Routing-Netz oberhalb eines physikalischen Netzes (z.B. eines ATM-, Frame-Relay-Netzes), in dem die C-LSR als Knoten und die E-LSR als Endkomponenten fungieren. Ein solches Netz stellt eine Routing-Ebene oberhalb des physikalischen Layer-2-Switching-Netzes dar. Dieses Switching-Netz wird im weiteren als MPLS-Switching-Netz bezeichnet.

Im allgemeinen stellt das MPLS ein Konzept für eine verteilte Integration des Routing (Layer 3) mit einem Layer-2-Switching-Netz (z.B. Frame Relay- bzw. ATM-Netz) dar. Beim MPLS unterscheidet man zwischen zwei Netz-Layers:

- MPLS-Routing-Netz auf Layer 3 und
- MPLS-Switching-Netz auf Layer 2.

Das MPLS-Switching-Netz bildet das physikalische Layer-2-Switching Netz.

Die Weiterleitung der IP-Pakete über das MPLS-Switching-Netz erfolgt anhand der den Paketen vorangestellten Labeln. Hierfür wird über das Switching-Netz eine virtuelle Ende-zu-Ende-Verbindung aufgebaut. Eine solche Verbindung wird als Label Switched Path (LSP) bezeichnet. Ein LSP stellt eine gerichtete virtuelle Verbindung dar. Für eine Vollduplex-Verbindung müssen somit zwei entgegengerichtete LSPs aufgebaut werden.

Ein LSP kann automatisch so bestimmt werden, dass zuerst eine Route über das MPLS-Routing-Netz zwischen den kommunizierenden E-LSRs mit Hilfe eines klassischen Routing-Protokolls (z.B. OSPF, RIP) ermittelt wird. Dann verläuft der LSP über diese Switches (d.h. im MPLS-Switching-Netz), deren LSR sich auf der Route innerhalb des Routing-Netzes befinden. Der LSP-Verlauf über das Switching-Netz kann auch manuell konfiguriert werden.
Bemerkung: Ein IP-Netz nach dem MPLS ist ein verbindungsorientiertes Netz und die Übermittlung der IP-Pakete erfolgt nach den gleichen Prinzipien, die in anderen verbindungsorientierten Netzen mit Paketvermittlung (z.B. X.25-, Frame-Relay und ATM-Netze) angewandt werden.

MPLS als Integration von Routing und Switching

Für die Integration von Routing und Switching beim MPLS enthält jeder Knoten im Netz zwei Funktionskomponenten:
o einen Layer-2-Switch, wo die Weiterleitung der IP-Pakete auf Basis der Label-Switching-Tabelle (LST) stattfindet, und o ein Router-Modul mit MPLS-Unterstützung, das eine C-LSR darstellt.

In einem LSR wird die klassische Routing-Funktion (z.B. nach dem Protokoll OSPF bzw. RIP) unterstützt. Zusätzlich realisiert der LSR die Verteilung der Label-Informationen innerhalb des logischen MPLS-Routing Netzes nach dem Protokoll LDP.

Am Eingang zum Netz wird zuerst jedes zu übertragende IP-Paket einer bestimmten Klasse, die man FEC (Forwarding Equivalence Class) nennt, zugeordnet und jeder FEC wiederum ein Label zugewiesen. Somit kann ein Label als FEC-Identifikation angesehen werden.

Die Routing-Tabelle und die Tabelle mit den Label/FEC-Zuordnungen dienen als Basis"-Informationen für die Instanz NHLFE (Next Hop Label Forwarding Entry). Diese Instanz enthält sämtliche Angaben, die man benötigt, um die IP-Pakete nach dem MPLS-Prinzip weiterzuleiten. Den Kern der Instanz NHLFE bildet die Label-Switching-Tabelle (LST), in der angegeben wird, wie die einzelnen IP-Pakete im Switch weitergeleitet werden sollen.

Die LST kann auch "verteilt" implementiert werden. Wie im weiteren gezeigt wird, ist es sinnvoll, dass eine LST jedem Eingangs-Interface im Switch zugeordnet wird. Eine LST eines Interfaces enthält die Angaben, wie die an diesem Interface empfangenen IP-Pakete weitergeleitet werden müssen.

Logisches Modell von MPLS
Nach dem MPLS können die unterschiedlichen Klassen der IP-Pakete über eine physikalische Leitung parallel übertragen werden. Label-Raum Die Datenübertragung über eine physikalische Leitung nach dem MPLS-Konzept kann als eine Verbindung zweier statistischer Multiplexer interpretiert werden. Die Ports im Multiplexer stellen Sende-/Empfangs-Puffer in, Speicher dar und werden hierbei mit Hilfe von Labeln identifiziert. Die beiden Multiplexer im Verbund müssen immer die gleiche Anzahl von Ports aufweisen. Im E-LSR und im C-LSR ist somit jeder Leitung eine Anzahl von Labeln zuzuordnen. Diese Anzahl von Labeln bezeichnet man als Label-Raum pro physikalisches Interface (per-interface label space).

Da einer Klasse von zu übertragenden IP-Paketen ein Label im E-LSR zugeordnet wird, bedeutet dies, dass Pakete in derselben Klasse im E-LSR zum Absenden immer am gleichen Port vor dem Multiplexer abgespeichert werden. Das Label, das einer Klasse von IP-Paketen zugeordnet wurde, dient gleichzeitig als Identifikation dieses Ports des Multiplexers, in dem die IP-Pakete dieser Klasse zum Absenden abgespeichert werden.

Nach dem MPLS-Konzept können mehrere Klassen von IP-Paketen parallel über eine physikalische Leitung übertragen werden. Hierbei wird jede Klasse mit einem Label markiert. Auf diese Weise kann eine physikalische Leitung auf eine Vielzahl von logischen Kanälen aufgeteilt werden. In diesem Zusammenhang stellt ein Label die Identifikation des logischen Kanals dar.

Prinzip von Label-Switching
Das Prinzip von Label-Switching besteht im allgemeinen darin, dass ein empfangenes IP-Paket (z.B. aus der Leitung x und mit dem Label a) mit einem (im allgemeinen) anderen Label auf eine andere Leitung (z.B. auf die Leitung y und mit dem Label b) weitergeleitet wird.

Die Aufgabe von Label-Switching ist es, virtuelle Verbindungen als einen sog. Label Switched Path (LSP) durch die Kopplung der logischen Kanäle in Switches zu realisieren. Hierfür müssen die Label-Werte mit Hilfe einer Label-Switching-Tabelle (LST) umgesetzt werden, so dass eine korrekte Verknüpfung der logischen Kanäle in IP-Netzknoten erfolgen kann. Ein LSP stellt eine Kette von logischen Kanälen in den einzelnen unterwegs liegenden physikalischen Leitungen dar. Hierbei wird ein logischer Kanal mit einem Label identifiziert. Ein Label kann auch als Nummer eines Ports angenommen werden.

Ein IP-Paket wird mit Label a vom Port a im Router A zum Port a im Switch übermittelt. Gemäß der Label-Switching Tabelle im Switch wird zuerst eine physikalische Ausgangsleitung für das Absenden dieses IP-Pakets und dann ein Port bestimmt, an dem diese Ausgangsleitung anliegt. Dem zu sendenden IP-Paket wird dadurch eventuell ein neuer Label-Wert, der dem neuen Port entspricht, vorangestellt.

Ein Label ist nur lokale Bedeutung hat, d.h. nur mit einer physikalischen Leitung verbunden. Auch ist es möglich, dass das Label in allen auf dem Pfad liegenden Switches nicht verändert wird. In einem solchen Fall wird den IP-Paketen auf allen Leitungen das gleiche Label vorangestellt. Andererseits soll hervorgehoben werden, dass die Übermittlung der IP-Pakete über einen bereits bestehenden Pfad zwischen den kommunizierenden Rechnern nur auf den vorangestellten Labeln basiert.

Logische Struktur der MPLS-Switching-Netze
Ein MPLS-Switching-Netz lässt sich als Geflecht logischer Kanäle verstehen. Eine virtuelle Ende-zu-Ende-Verbindung im MPLS-Switching-Netz stellt einen Label Switched Pfad (LSP) dar, der als eine Kette von logischen Kanälen innerhalb von physikalischen Leitungen angesehen werden kann. Hierbei werden die logischen Kanäle mit Hilfe von Labeln identifiziert. Mit einer Leitung ist immer ein Label-Raum verbunden. Über einen LSP wird eine Klasse FEC (Forwarding Equivalence Class) von IP-Paketen übermittelt. Die Zuordnung der zu übertragenden Pakete zu einer bestimmten Klasse erfolgt im Quell-E-LSR.

Über einen LSP werden die IP-Pakete nur in eine Richtung transportiert. Für eine virtuelle Vollduplex-Verbindung sind zwei entgegengerichtete LSPs nötig.

Bemerkung: Eine virtuelle Vollduplex-TCP-Verbindung setzt sich aus zwei entgegengerichteten, unidirektionalen TCP-Teil-Verbindungen zusammen. Jede gerichtete TCP-Teilverbindung kann somit auf der Basis eines LSP eingerichtet werden. Für eine Vollduplex-TCP-Verbindung sind daher zwei LSPs notwendig.

Die IP-Pakete werden über einen LSP anhand von Labeln in den Switches weitergeleitet. Da die Pakete auf einem LSP immer über die gleiche virtuelle Übertragungsstrecke verlaufen, wird die Reihenfolge der übermittelten IP-Pakete im MPLS-

Switching-Netz nicht verändert. Dies ist ein großer Vorteil im Vergleich zu den klassischen IP-Netzen, die verbindungslos sind.

Bildung der Klassen von IP-Paketen und MPLS Einsatz

Im E-LSR werden die zu übertragenden IP-Pakete klassifiziert d.h. einer sog. Forwarding Equivalence Class (FEC) zugeordnet. Jede Klasse von IP-Paketen wird wiederum über einen virtuellen Pfad LSP im Netz übermittelt. Da die Klassen von IP-Paketen nach unterschiedlichen Kriterien gebildet werden können, ergibt sich dadurch ein breites Spektrum von MPLS-Einsatzmöglichkeiten.. Beispielsweise kommen folgende Kriterien für die Zuordnung von IP-Paketen zu den FECs im E-LSR in Frage:

- FEC: alle IP-Pakete zu einem Ziel-Subnetz. In diesem Fall wird ein virtueller Pfad vom Quell-E-LSR zum Ziel-E-LSR aufgebaut, um die IP-Pakete von einem IP-Subnetz zu einem anderen zu übermitteln. Zwei entgegengerichtete LSPs entsprechen in diesem Fall einer virtuellen Vollduplexleitung zwischen den Subnetzen.
- FEC: alle IP-Pakete zu einem Ziel-Rechner. Hierbei wird ein virtueller Pfad zwischen einem Quell-E-LSR und einem Ziel-E-LSR aufgebaut, um die IP-Pakete an einen Zielrechner zu über mitteln. Diesen Fall könnte man als eine Variante von IP-Switching interpretieren.
- FEC: alle IP-Pakete zwischen zwei Routern, über die zwei Standorte eines Unternehmens angeschlossen sind.

Werden zwei entgegengerichtete virtuelle Pfade über ein MPLS-Netz bei einer derartigen Zuordnung der IP-Pakete zur FEC aufgebaut, so könnte man diese Pfade mit einer virtuellen Standleitung zwischen zwei Standorten eines Unternehmens vergleichen. Über diese virtuelle Standleitung kann ein virtuelles privates Netz auf IP-Basis aufgebaut werden.

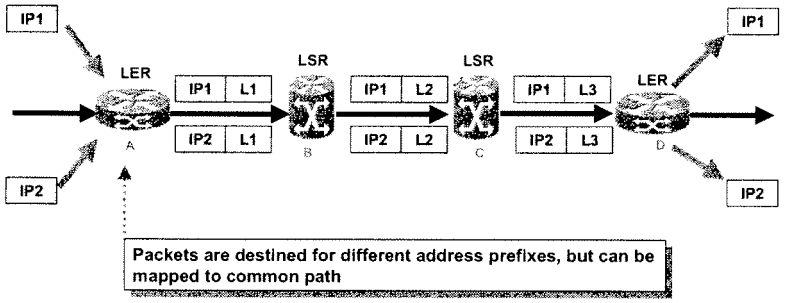


Bild: Forwarding Equivalence Classes (FEC)

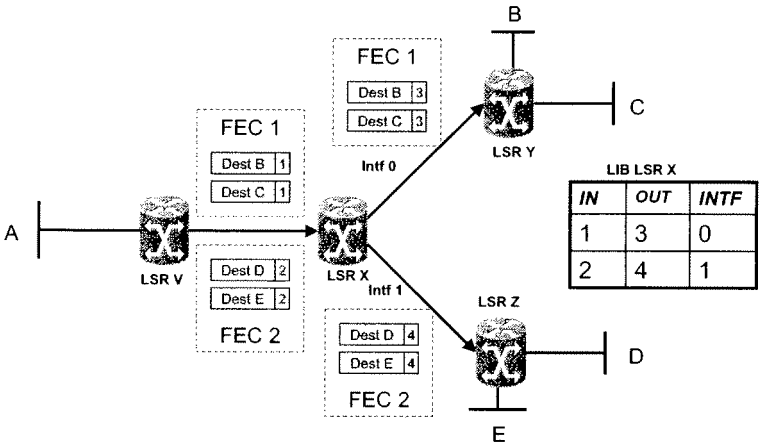


Bild: Beispiel für Forwarding Equivalence Classes (FEC)

1. Expedited Forwarding Service [RFC2598]
   - low loss, low latency, low jitter, assured bandwidth
   - code point for the EF PHB is 101110

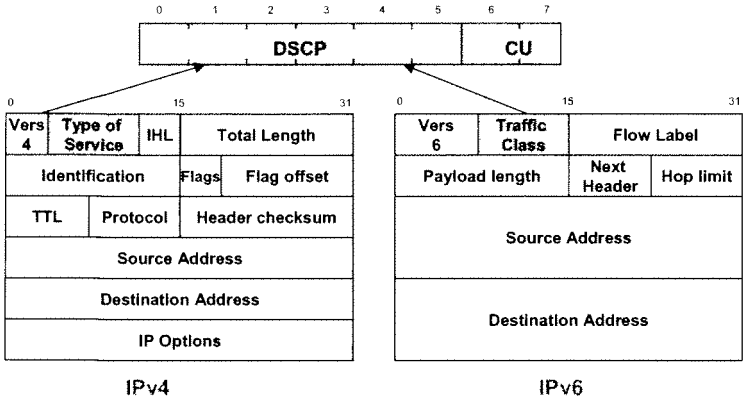
2^6 = 64 possible classes
14 classes – actually defined

2. Assured Forwarding Service [RFC2597]
   - no bandwidth guarantee, but packets labelled with high priority

Dropping Precedence	*Class 1*	*Class 2*	*Class 3*	*Class 4*
Low	AF_{11} = 001010	AF_{21} = 010010	AF_{31} = 011010	AF_{41} = 100010
Medium	AF_{12} = 001100	AF_{22} = 010100	AF_{32} = 011100	AF_{42} = 100100
High	AF_{13} = 001110	AF_{23} = 010110	AF_{33} = 011110	AF_{43} = 100110

3. Best-Effort Service
   - no guarantee for QoS
   - code point 000000 – default codepoint, default PHB

Bild: DiffServ Classes



IPv4 IPv6

- An octet specifying the PHB class
- Inserted in the ToS octet of IPv4 or Traffic Class octet in IPv6
- Is backward compatible with existing IP packets
- 6-bit codepoint with 2-bit unused

Bild: DSCP in the IPv4 and IPv6 Headers

- Objective: mechanisms that ensure that each service class receives the proper PHB at each LSR in the LSP

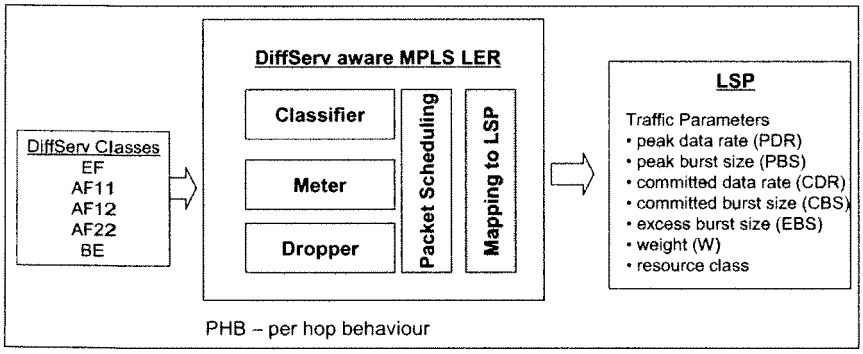
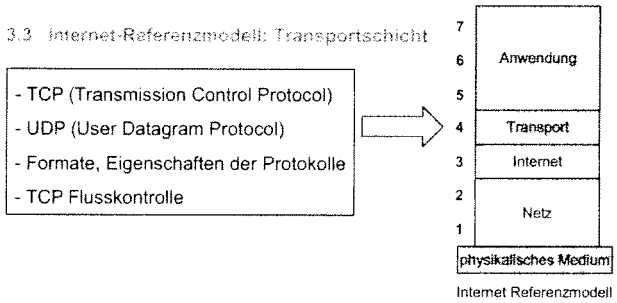


Bild: DiffServ over MPLS Traffic Engineering

3.3 Internet-Referenzmodell: Transportschicht



TCP (Transmission Control Protocol) leistet eine zuverlässige Übertragung, die verbindungs- und bytestromorientiert ist. Dafür ist erheblicher Aufwand zur Abarbeitung des TCP-Protokolls erforderlich, der zu einem geringen Durchsatz führt.

UDP (User Datagram Protocol) hingegen überträgt verbindungslos und unzuverlässig, erreicht dafür aber einen höheren Durchsatz.

Transmission Control Protocol (TCP)
- verbindungsorientiert
- Bytestrom-orientiert
- unterstützt nur 1:1-Kommunikation
- zuverlässig

User Datagram Protocol (UDP)
- verbindungslos
- Nachrichten-orientiert
- Multicast-Unterstützung
- unzuverlässig
  - optionale Fehlererkennung
  - keine Fehlerbehebung
  - keine Reihenfolgeerhaltung

Bild: Transportprotokolle im Internet

- **Ziel:** Kommunikation zwischen Prozessen
- Prozess-ID des Betriebssystems ist zur eindeutigen Identifikation von Prozessen nicht geeignet.
- vollständige Transportadresse = IP-Adresse + Port-Nummer
- Port = Kommunikationsendpunkt

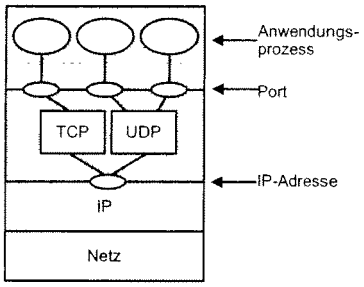
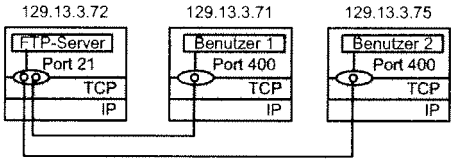


Bild: Transport-Adressen im Internet



- Identifikation von TCP-Diensten über Ports (vergleichbar mit SAPs)
- reservierte Portnummern bis 255 für häufig benutzte Dienste (well-known ports)
- Socket: IP-Adresse + Port (vergleichbar mit Verbindungsendpunkt CEP innerhalb von SAP
- **Beispiel**
  - Verbindung mehrerer Benutzer mit Port-Nummer 400 zu 1 FTP-Server
  - Identifizierung der Verbindungen über IP-Adresse und Port-Nummer

Bild: Adressierung

TCP (Transmission Control Protocol) hat als Protokoll der Transportschicht die Aufgabe eine zuverlässige Verbindung (keine fehlenden, falschen, duplizierten oder vertauschten Segmente beim Anwendungsprozess im Empfänger) zwischen genau zwei Endpunkten zu gewährleisten.

TCP stellt eine virtuelle Verbindung zwischen den TCP-Instanzen zweier Endsysteme her. Die TCP Software sorgt dafür, dass die Anwendung eine zuverlässige Verbindung vorfindet. TCP hat folgende Eigenschaften: Vollduplex-Übertragung, Fehlerbehandlung durch Go-Back-N und Flusssteuerung mittels sliding window. TCP ist byteorientiert (stromorientiert), d. h., es transportiert einen zuverlässigen Byte-Strom, Sequenz- und Quittungsnummern beziehen sich auf Bytes.

Vor Beginn des Datenaustauschs wird eine Verbindung aufgebaut, falls beide Teilnehmer dem Aufbau zustimmen. Dabei findet keine Interaktion mit vorangehenden Verbindungen statt. Beim Verbindungsabbau werden noch ausstehende Dateneinheiten (TCP-Segmente) zuverlässig übermittelt.

Transport-Protokolle TCP und UDP

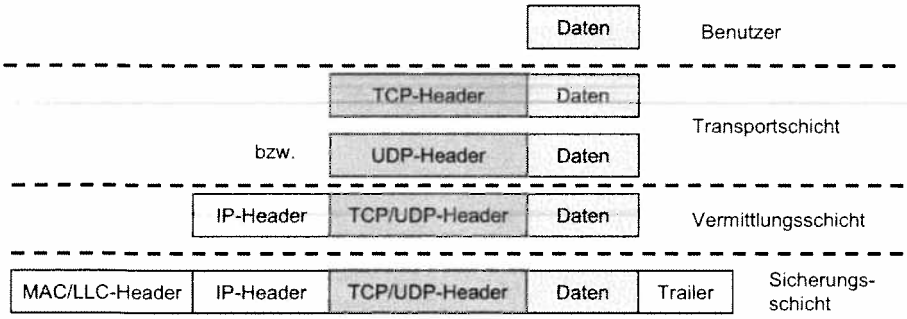
Die TCP/IP-Protokollfamilie stellt für die Übertragung der Nutzdaten zwei sehr unterschiedliche Transportprotokolle auf IP-Basis zur Verfügung:

- Transmission Control Protocol (TCP) mit einer gesicherten, verbindungsorientierten Vollduplex-Kommunikation zwischen Sender und Empfänger.
- User Datagram Protocol (UDP) mit einer ungesicherten, verbindungslosen Kommunikation zwischen Sender und Empfänger.

Neben diesen beiden Protokollen umfasst die TCP/IP-Protokollfamilie noch weitere Transportprotokolle, die jedoch auf TCP und UDP aufsetzen und sich deren jeweilige Eigenschaften zunutze machen:

- Übertragung der ISO-Transportprotokolle TP0 bis TP4 (RFC 905, 1240 und 2126),
- NetBIOS über TCP/UDP (RFC 1001/1002),
- Real Time Protocol (RTP) über UDP (RFC 1889),
- IBM's Protokoll SNA (System Network Architecture) über UDP (RFC1538).

Während TCP im Laufe der Entwicklung mit seinen Kontrollmechanismen zusehends verfeinert wurde, ist das UDP Protokoll aufgrund seines sehr schmalen Funktionsumfangs im wesentlichen unverändert geblieben, sieht man von den Portmapper-Diensten ab, die seinen Einsatz stark vereinfachen.



Die Verschachtelung von Dateneinheiten wird am Beispiel TCP/IP in Erinnerung gerufen.

Bild: PDU Verschachtelung

Sicherungsprotokolle	Transportprotokolle
Instanzen sind über einen Link verbunden	Instanzen sind in nicht benachbarten Systemen
eher feste, kleine Umlaufzeiten der Segmente (round trip times, RTTs)	stark variierende RTTs, dadurch hohes Bandbreiten- Verzögerungs-Produkt
keine Reihenfolgevertauschungen	Mögliche Reihenfolgevertauschungen
Link limitiert Senderate	möglicher Engpass auf einem Zwischen-Link

Bild: Sicherungs- vs. Transportprotokolle

Die Gemeinsamkeiten und Unterschiede zwischen Sicherungs- und Transportprotokollen ist im Tabelle zusammengefasst.

RTT = Round Trip Time

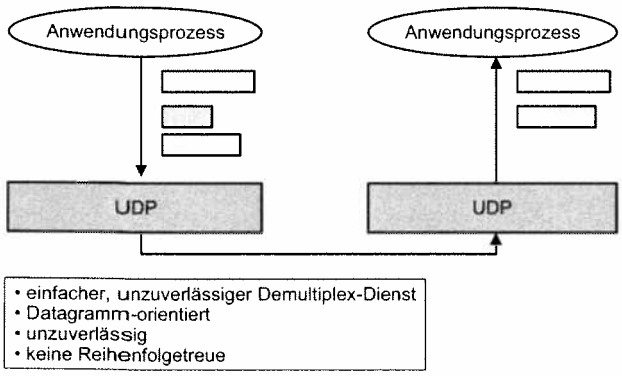


Bild: User Datagram Protocol (UDP)

UDP (User Datagram Protocol, RFC 768) ist ein verbindungsloses, unzuverlässiges Transportprotokoll. UDP kann für verschiedene Protokolle der Anwendungsschicht genutzt werden, die durch eine Portnummer identifiziert werden **(Port-Multiplexing).** Insbesondere werden TFTP (Portnummer 69), DNS (53), SNMP (161) und RPC (111) über UDP abgewickelt. Portnummern für TCP und UDP können verschieden sein, jedoch wird für Dienste, die sowohl mit TCP als auch mit UDP erreichbar sind, eine einheitliche Nummer festgelegt.

TFTP: Trivial Transfer Protocol
DNS: Domain Name System
SNMP: Simple Network Protocol
RPC: Remote Procedure Call

Der Header von UDP ist sehr einfach. Die Portnummern für Sender und Empfänger werden mit je 16 Bit codiert. Das Feld Länge gibt die gesamte Länge des Datagramms (inklusive Header) in Byte an. Der Minimalwert ist 8, entsprechend der Länge des Headers. Die Prüfsumme ist optional, ein Wert von 0 zeigt an, dass sie nicht verwendet wurde. Der Algorithmus für die Prüfsumme ist derselbe wie in IP. Da IP jedoch keine Prüfsumme über das Datenfeld überträgt, ist es nicht ratsam, die UDP-Prüfsumme wegzulassen. Die Berechnung der Prüfsumme erfolgt über den UDP-Header, die Daten und den Pseudoheader. Letzterer besteht aus 12 Byte, die die IP-Quelle und Zieladresse beinhalten. Dadurch wird der Empfänger in die Lage versetzt zu prüfen, ob das Datagramm an der richtigen Adresse (IP-Adresse und Protokollnummer) angekommen ist.

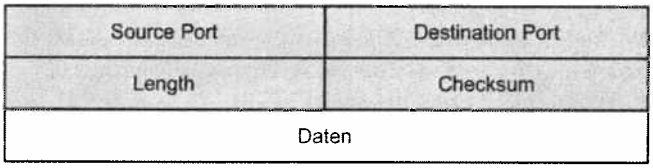
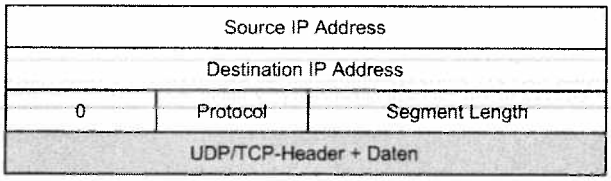


Bild: UDP-Header

Der UDP-Header besteht aus den Source und Destination Ports, die Länge der UDP-Dateneinheit (inklusive Header) in Byte und eine Prüfsumme, die mit Hilfe eines sogenannten Pseudo-Headers gebildet wird.



Pseudo-Header

Teile des IP-Headers der UDP-Header bilden zusammen den UDP Pseudo-Header.

Im Gegensatz zum TCP erfordert das UPD-Protokoll (RFC 768) nicht notwendigerweise die Berechnung einer Prüfsumme; auch wenn dies aktuelle UDP-Implementierungen in der Regel leisten.

- Definition eines Pseudo-Headers
- Berechnung der Prüfsumme (optional in UDP über IPv4) über
  - Pseudo-Header
  - TCP/UDP-Header
  - Daten
- Absicherung von IP-Protokollinformation
- Erlaubt Schutz gegen fehlgeleitete UDP-Pakete
- Ist Prüfsummenfeld = 0, dann wird keine Prüfsummenberechnung gewünscht
  Bei berechneter Prüfsumme 0 wird 0xFFFF übertragen

Bild: Prüfsummenberechnung (Pseudo-Header)

Keine Verbindungsaufbauphase
Verzögerung bis zum Aufbau einer Verbindung entfällt. Daten können sofort gesendet werden.

Kein Verbindungszustand
- Im Endsystem müssen keine verbindungsrelevanten Informationen gehalten werden (z.B. Flusskontrollfenster, Staukontrollfenster, Sequenznummern)
- Server kann mittels UDP typischerweise mehr aktive Clients unterstützen als mit

Geringer Overhead in der Dateneinheit
- Lediglich Adressen sowie Längenfeld und Prüfsumme

Unreguliertes Senden
UDP kann Daten so schnell senden wie sie von der Anwendung geliefert werden und wie sie vom Netz abgenommen werden (Verlust bei Sender möglich)

Vorteil: Sehr einfaches Protokoll mit äußerst geringem Overhead

Bild: Eigenschaften von UDP

Durch UDP können Anwendungen Datagramme (selbständige Datenblöcke) senden und empfangen. UDP bietet - ähnlich dem IP - keine gesicherte Übertragung und keine Flusskontrolle. Ob die Dateneinheiten auch wirklich beim Empfänger ankommen, ist nicht sichergestellt.

Das Bild fasst die Eigenschaften von UDP zusammen.

20	FTP (Data), File Transfer Protocol	(TCP)
21	FTP (Control)	(TCP)
23	TELNET, Terminal Emulation	(TCP)
25	SMTP, Simple Mail Transfer Protocol	(TCP)
53	DOMAIN, Domain Name Server	(UDP)
67	BOOTPS, Bootstrap Protocol Server	(UDP)
68	BOOTPC, Bootstrap Protocol Client	(UDP)
69	TFTP, Trivial File Transfer Protocol	(UDP)
80	HTTP Hypertext Transfer Protocol (default port)	(TCP)
111	SUN RPC, Run Remote Procedure Call	(TCP)
161	SNMP, Simple Network Management Protocol	(UDP)

Bild: Well-Known Ports

- < 512: reserved for particular services (one per host)
- 512 ... 1023: privileged port (Unix superuser only)
- 1024 ... 65535: available for applications
- some services offered both as UDP and TCP *may* have same port number

7	echo
9	discard
19	character generator
37	time
53	domain name service
123	Network Time Protocol (NTP)

Bild: UDP Port Numbers

Die Bedeutung von UDP liegt in seinem Transportdienst für andere wichtige Internet-Protokolle. Hierzu zählen u.a.:

- Trivial File Transfer Protocol (TFTP),
- Remote Procedure Calls (RPC),
- Network File System (NFS),
- Domain Name Services (DNS),
- Simple Network Management Protocol (SNMP),
- BOOT Protocol,
- Lightweight Directory Access Protocol (LDAP),
- TIME und DAYTIME Nachrichten,
- Versenden von Broadcast-Nachrichten.

	TCP	UDP
Verbindungen	x	-
Datenfluss am Dienstzugangspunkt	**Bytestrom**	**Dateneinheiten**
Demultiplexen	x	x
Reihenfolgeerhaltung	x	-
Fehlererkennung	x	**optional**
Fehlerbehebung	x	-
Flusskontrolle	x	-
Staukontrolle	x	-

Bild: Vergleich zwischen TCP und UDP

In der Tabelle sind die Eigenschaften von TCP und UDP vergleichend zusammengestellt.

Funktion des Protokolls TCP

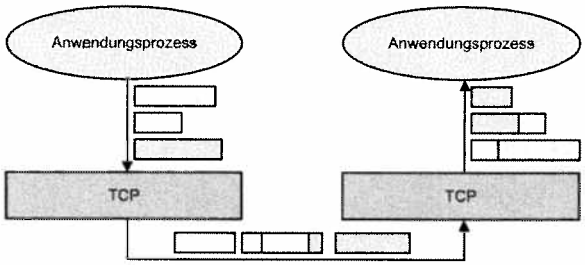
Das Protokoll TCP ist ein Transportprotokoll mit der Aufgabe den Datenaustausch zwischen zwei kommunizierenden Rechnern auf einer sicheren Basis zu gewährleisten, d.h. Datenverluste und -verfälschungen während der Übertragung zu entdecken und dementsprechend eine wiederholte Übertragung zu veranlassen.

Es werden sogenannte Ports den aktiven TCP/IP-Anwendungen als Anwendungsprozessen zugeordnet. Diese Ports stellen die individuellen Kommunikationspuffer einzelner Anwendungsprozesse dar. Die Ports werden den Anwendungsprozessen nach Bedarf (auch dynamisch) zugeordnet. Die Port-Nummern sind 16 Bits lang. Die Ports mit den Nummern 0 bis 1023 sind weltweit eindeutig für Standarddienste (sog. Well-Known Services) wie z.B. TELNET, FTP, HTTP von vornherein reserviert. Die reservierten Ports von Standardanwendungen werden auch als Well-Known Ports bezeichnet. Unter den Nummern im Bereich von 1024 bis 65 535 können sie im Rechner den Anwendungsprozessen frei zugeteilt werden.

Das Protokoll TCP ist auch als ein logischer Multiplexer von Anwendungsprozessen zu sehen. Das Protokoll TCP ermöglicht es, dass mehrere Anwendungsprozesse auf die Dienste des Protokolls IP gleichzeitig zugreifen können. Jedes Sicherungsprotokoll ist verbindungsorientiert. Dies setzt voraus, dass eine Beziehung zwischen zwei entsprechenden Kommunikationspuffern in den kommunizierenden Rechnern zustande kommen muss. Um die Datenübermittlung nach dem Protokoll TCP sicher zu machen, muss ebenfalls eine logische Verbindung aufgebaut werden.

- Verbindungsaufbau zwischen zwei Sockets (Verbindungsendpunkte)
- Datenübertragung
  - Vollduplex über virtuelle Verbindung
  - Unterstützung von Prioritäten
  - Reihenfolgetreue
  - Fluss-/Staukontrolle mit Fenstermechanismus
  - Fehlerkontrolle
    - Folgenummern
    - Prüfsumme
    - Quittierungsnummern
    - Übertragungswiederholung
- Gesicherter Verbindungsabbau

Bild: Transmission Control Protocol (TCP)



- unterstützt reihenfolgetreuen und zuverlässigen Byte-Strom
- Segmentgröße richtet sich nach MTU-Größe des lokalen Links oder Segment wird nach Push-Operation bzw. Timeout erzeugt

Bild: TCP-Datenaustausch

Stellt zuverlässigen, verbindungsorientierten Punkt-zu-Punkt-Dienst zur Verfügung
Arbeitet an der Dienstschnittstelle mit einem Bytestrom
(RFC 793, September 1981)

Ist in der Regel über IP implementiert
Die zu IP weitergereichte Dateneinheit wird als TCP-Segment bezeichnet
Logische Sicht auf TCP-Ebene

Neue / optimierte Funktionen
- Verbesserung des Timeout-Mechanismus (Zeitstempel im Segment-Kopf)
- Erhöhen der Fenstergröße (Basiseinheit kann skaliert werden)

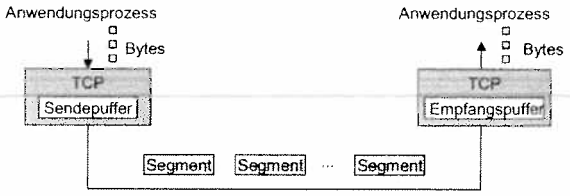


Bild: TCP-Datenaustausch

TCP-Verbindungen sind vollduplex. Man kann jede Verbindung als ein Paar von gegenseitig gerichteten unidirektionalen Verbindungen interpretieren. Der Verbindungsaufbau zwischen zwei Rechnern, die das TCP-Protokoll benutzen, erfolgt immer mit Hilfe des Three-Way-Handshake-Verfahrens, das für eine Synchronisation der Kommunikationspartner sorgt und sicherstellt, dass die TCP-Verbindung in jede Richtung korrekt initialisiert wird (vollduplex). An dieser Stelle ist hervorzuheben, dass der Anwendungsprozess im Quellrechner mit dem TCP über einen wahlfreien Port kommuniziert, der dynamisch (aber im Quellenrechner nur einem gleichzeitig) zugewiesen wird.

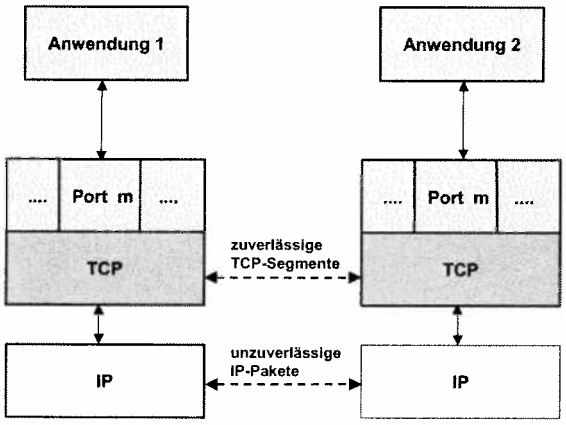


Bild: TCP-Verbindung

Socket
Das Tupel aus IP-Adresse und Port-Nummer wird als Socket bezeichnet.

Im Protokollkopf gibt es Felder, die als Source Port und Destination Port bezeichnet werden. Die Port-Nummern dienen in Senderichtung (also der Destination Port) zur Identifikation des Dienstes oberhalb TCP oder UDP, üblicherweise werden die sogenannten Well-Known Ports im Bereich 0 ... 1023 benutzt (Dienst TELNET die Port-Nummer 23 und das Hyper Text Transfer Protocol (HTTP) die Port-Nummer 80).

In Empfangsrichtung dient der Port (also der source port) zur Auswahl für die richtige Instanz auf dem richtigen Client, denn nur so ist auch möglich, dass z. B. mit einem Browser mehrere Web-Sessions gleichzeitig durchgeführt werden können.

Es ist also der Port, über den die Antwort erwartet wird, üblicherweise ist das ein sogenannter short lived port im Bereich ab 1024 aufwärts. Um also eine Anwendung auf einem bestimmten Rechner anzusprechen wird eine IP-Adresse benötigt - sie wählt den Rechner (Host) aus - und eine Port-Nummer - sie wählt die Anwendung, genauer die richtige Instanz eine Anwendung aus. Und genau

Bei Windows-Rechnern gibt es der Begriff der Winsock - es handelt sich dabei um das Programm, das die Protokollschichten IP, TCP und UDP behandelt und der Anwendung, z.B. dem Browser, den Socket zur Verfügung stellt.

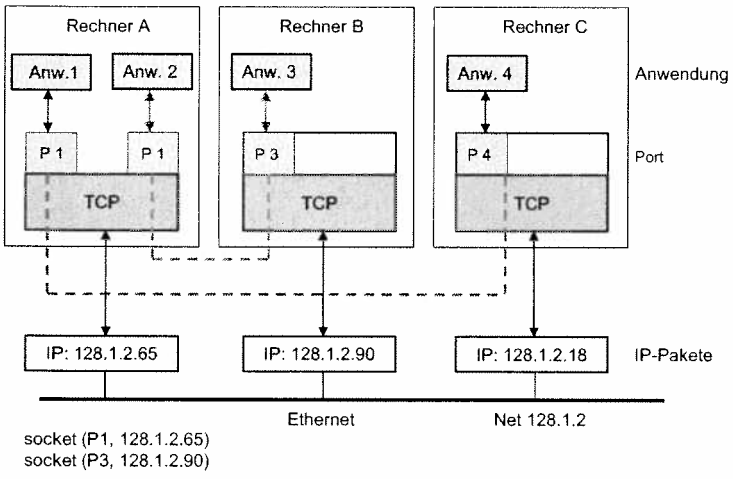


Bild: TCP-Verbindungen

TCP-Ports sind entweder

- **reserviert** (auch: privilegierte Portnummern 1-255 für TCP/IPAnwendungen, 256-1023 für bestimmte UNIX-Anwendungen),
- **registriert** (Nummern zwischen 1024 und 49151 werden von IANA registriert) oder
- **privat, dynamisch** (Portnummern zwischen 49152 und 65535).

Ein Client verwendet die reservierte Portnummer (sofern diese existiert) einer gewünschten Anwendung als Zielport, als Quellenport wählt er eine nicht reservierte Portnummer, die er selbst im Augenblick noch nicht benutzt.

Anwendung	Anwendungs-protokoll	Verwendetes Transportprotokoll
Email	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
Dateitransfer	FTP	TCP
Entfernter Fileserver	NFS	i.d.R. UDP
Streaming multimedia	Proprietär	i.d.R. UDP
Internet-Telefonie	Proprietär	i.d.R. UDP
Netzmanagement	SNMP	i.d.R. UDP
Intra-Domain-Routing	RIP	i.d.R. UDP
Inter-Domain-Routing	BGP	TCP
Namensübersetzung	DNS	i.d.R. UDP

Bild: Anwendungsprotokolle mit TCP oder UDP

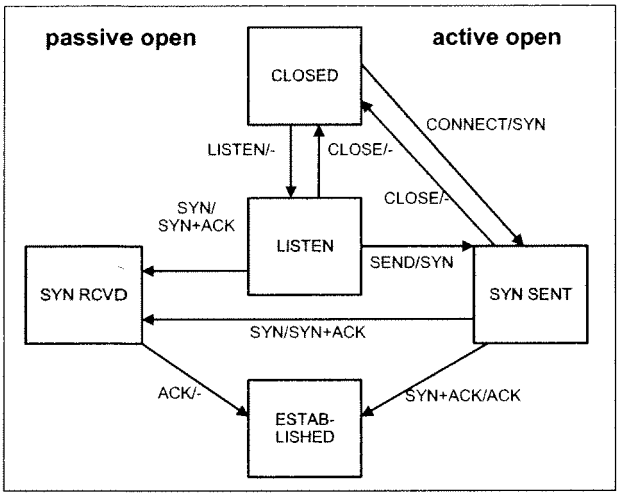


Bild: Zustandsübergangsdiagramm beim TCP-Verbindungsaufbau

Hosts A and B müssen sich über die Wahl eines ISN (Initial Sequence Number) einigen.
- Verwendung eines 3-Way-Handshake

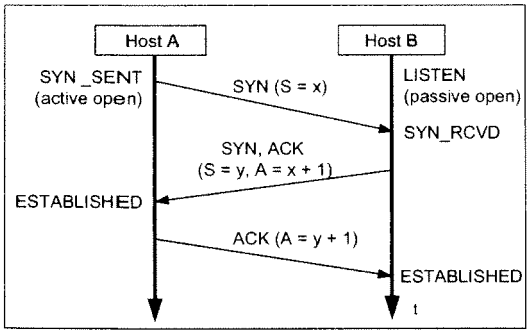


Bild: TCP-Verbindungsaufbau

Das TCP-Modell geht von einer sog. Zustandsmaschine aus.

Eine TCP-Instanz befindet sich immer in einem wohldefinierten Zustand. Die Hauptzustände sind hierbei
- Listen,
- Established.

Zwischen diesen stabilen Zuständen gibt es eine Vielzahl zeitlich befristeter Zwischenzustände. Mittels der TCP-Kontroll-Flags SYN, ACK, FIN, und eventuell auch RST (Reset) wird zwischen den Kommunikationspartnern der Wechsel bzw. der Verbleib in einem Zustand signalisiert.

ACK: Acknowledgement (Quittung),
SYN: Synchronize (Synchronisation der Sequenznummern in beiden Richtungen beim Verbindungsaufbau),
FIN: Finalize (Abbau der Verbindung),
RST: Reset (Verbindungsabbruch).

- **Wahl von Initial Sequence Number (ISN):**
  - Es kann nicht einfach null gewählt werden.
  - Eine Nummernüberlappung mit früheren Verbindungen muss vermieden werden)
- **Vorraussetzung für die Wahl von ISN:**
  - Garantiert korrekter Betrieb
    - Ohne Synchronisation der Uhrzeiten
    - Auch im Fehlerfällen

Bild: Wahl des ISN (Initial Sequence Number)

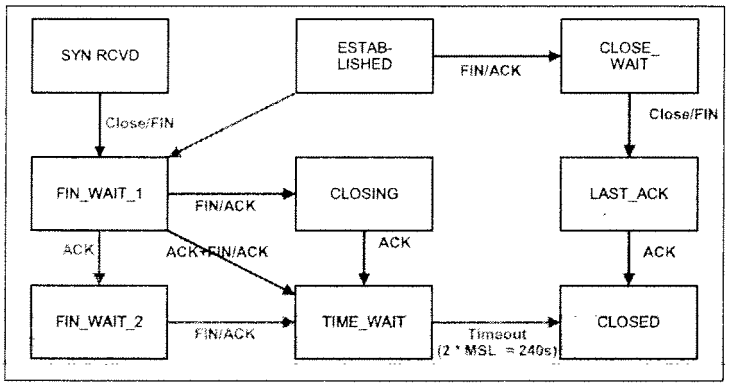


Bild: Zustandsübergangsdiagramm beim TCP-Verbindungsabbau

TCP-Protokollautomat

Das Zustandsdiagramm für TCP bestimmt die möglichen Abläufe im Einzelnen.

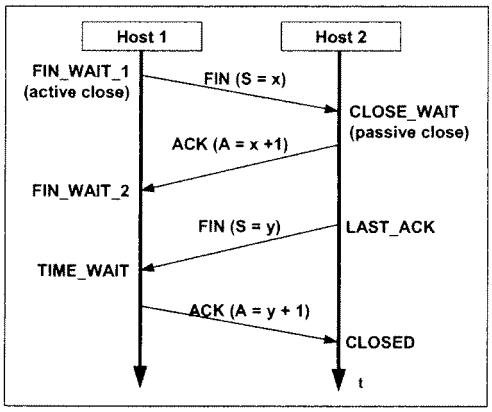


Bild: TCP-Verbindungsabbau

- **Normaler Verbindungsabbau**
  - Ein einseitiger Abbau ist vorgesehen
  - Ee muss eine Sequenznummerüberlappung vermieden werden
- **TCP muss auch nach einem Close-Befehl weiterhin Daten empfangen können.**
  - Weil der Verbindung nicht direkt geschlossen werden kann, muss dafür gesorgt werden, dass eine neue Verbindung mit einer anderen Sequenznummer anfängt.

Bild: TCP-Verbindungsabbau

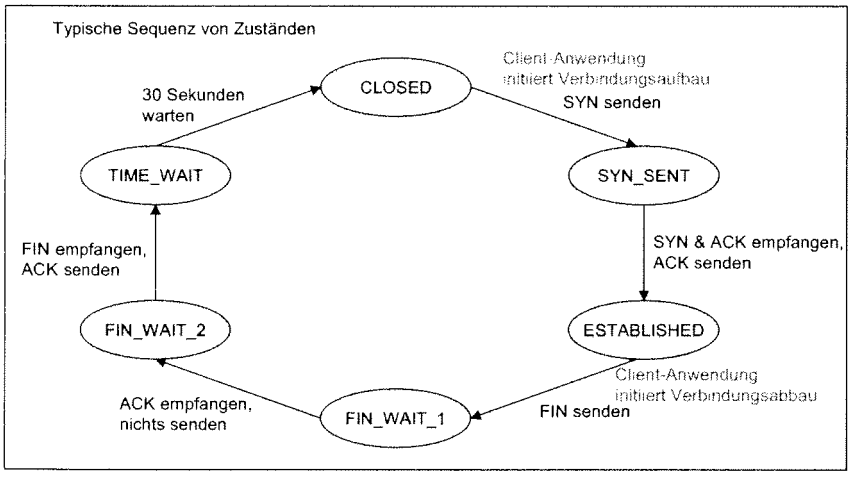


Bild: Zustandsübergangsdiagramm eines TCP-Clients

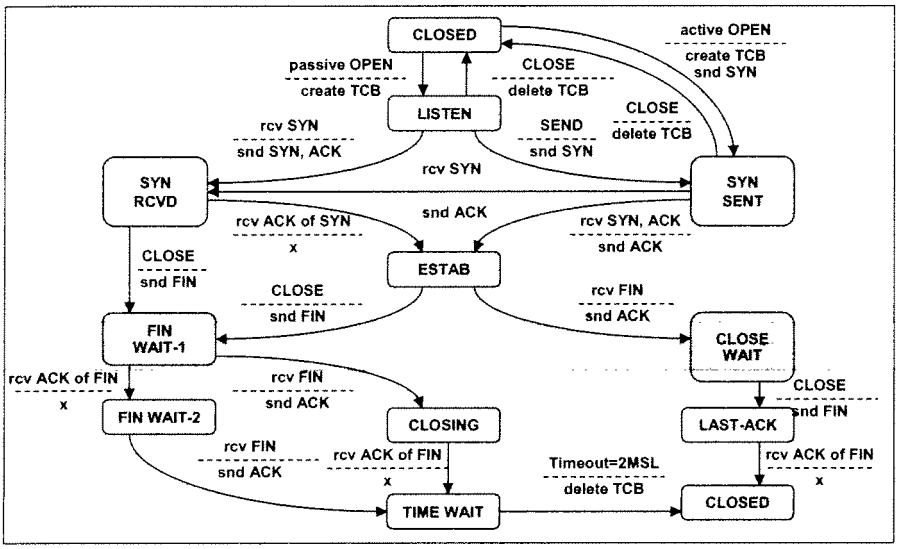


Bild: Zustandsübergangsdiagramm von TCP

Das TCP-Protokokoll verfügt über nur wenige Programmschnittstellen (RFC 793), mit denen Applikationen den Auf- und Abbau von Verbindungen sowie die Datenübertragung beeinflussen können.

Diese Programmschnittstellen werden TCP Application Program Interface (API) genannt und beinhalten die Funktionen:

- **Open:** Öffnen von Verbindungen mit den Parametern:
  - Aktives/Passives Öffnen,
  - Entfernter Socket, d.h. Portnummer und IP-Adresse des Kommunikationspartners,
  - Lokaler Port,
  - Wert des Timeouts (optional),
  - Als Rückgabewert an die Applikation dient ein lokaler Verbindungsname zur Identifikation dieser Verbindung.
- **Send**: Übertragung der Benutzerdaten an den TCP-Sendepuffer und anschließendes Versenden über die TCP-Verbindung. Optional kann das URG- bzw. PSH-Bit gesetzt werden.
- **Receive:** Daten aus dem TCP-Empfangspuffer werden an die Applikation weitergegeben.
- **Close:** Beendet die Verbindung, nachdem zuvor alle ausstehenden Daten aus dem TCP-Empfangspuffer zur Applikation übertragen und ein TCP-Segment mit dem FIN-Bit versandt wurde.
- **Status:** Gibt Statusinformationen über die Verbindung aus, wie z.B. lokaler und entfernter Socket, Größe des Sende- und Empfangsfensters, Zustand der Verbindung und evtl. lokaler Verbindungsname.
- **Abort:** Sofortiges Unterbrechen des Sende- und Empfangsprozesses und Übermittlung des RST-Bits an die Partner-TCP-Instanz.

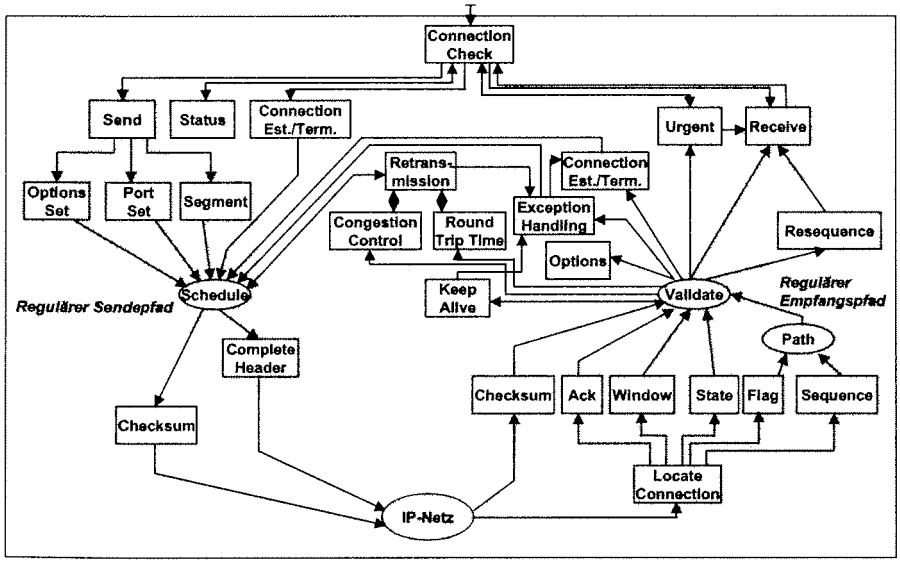
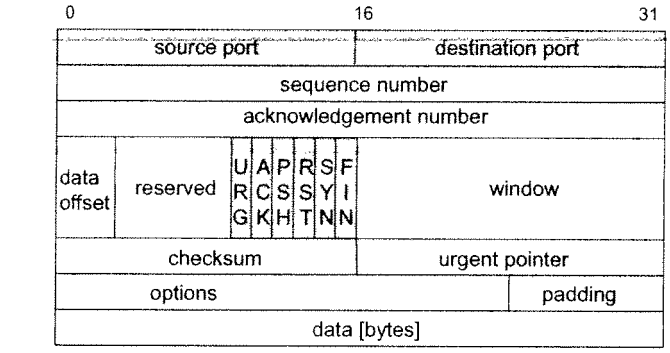


Bild: Funktionsbasierte Dekomposition von TCP

Aufbau von TCP-Segmenten

Über eine TCP-Verbindung werden die Daten in Form von festgelegten Datenblöcken oder TCP-Segmente ausgetauscht. Aufgabe von TCP ist es, die zu übertragenden Daten im Quellrechner in nummerierte Segmente zu zerlegen, was man als Segmentierung bezeichnet. Jedes Datensegment erhält einen TCP-Header. Vor Beginn einer Übertragung wird zwischen den TCP-Instanzen im Quell- und im Zielrechner die maximale Segmentgröße vereinbart. Die TCP-Segmente werden dann vom IP-Protokoll als zusammenhanglose IP-Pakete (Datagramme) übertragen. Die TCP-Instanz des Zielrechners setzt die empfangenen IP-Pakete in der richtigen Reihenfolge in die ursprünglichen Daten (Nachricht) zurück. Kommt das IP-Paket nicht beim Zielrechner an, wird die Wiederholung der Übertragung eines entsprechenden Datensegments von TCP veranlasst.



URG Urgent pointer field significant
ACK Acknowledge field significant
PSH Push function
RST Reset the connection
SYN Synchronize the sequence numbers
FIN Finalize, no more data from sender

Bild: TCP Header

Die Kontrollinformationen im TCP-Header sind wie folgt:

- **Source Port:** Der Quell-Port enthält die Portnummer des Anwenderprozesses im Quellrechner, der die Daten sendet.
- **Destination Port:** Der Ziel-Port enthält die Portnummer des Anwenderprozesses im Zielrechner, an den die Daten adressiert sind.

- **Sequence Number:** Die Sequenznummer gilt in der Senderichtung und dient zur Nummerierung der gesendeten Datensegmenten. Beim Aufbau einer virtuellen Ende-zu-Ende-Verbindung generiert jedes TCP-Modul eine Anfangs-Sequenznummer. Diese Nummern werden ausgetauscht und gegenseitig bestätigt. Im Quellrechner wird die Sequenznummer immer jeweils um die Anzahl bereits gesendeter Bytes erhöht.
- **Acknowledgement Number:** Die Quittungsnummer gilt in der Empfangsrichtung und dient der Bestätigung von empfangenen Datensegmenten. Diese Nummer wird vom Zielrechner gesetzt, um dem Quellrechner mitzuteilen, bis zu welchem Byte die Daten korrekt empfangen wurden.
- **Data Offset:** Das vier Bit große Feld mit der Bezeichnung Datenabstand gibt die Länge des TCP-Headers in 32-Bit-Worten an und damit die Stelle, ab der die Daten beginnen.
- **Control Flags:** Die Kontroll-Flags legen fest, welche Felder im Header gültig sind, und steuern somit die Verbindung. Es gibt 6 Bits, und wenn das entsprechende Bit gesetzt ist, gelten die folgenden Bedingungen:
  - **URG:** Der Urgent Pointer (Zeiger im Urgent-Feld) ist gültig.
  - **ACK:** Die Quittungsnummer ist gültig.
  - **PSH (Push)**: Die Daten sollen sofort an die nächsthöhere Schicht weitergegeben werden.
  - **RST (Reset):** Die Verbindung soll zurückgesetzt werden.
  - **SYN:** Verbindungsaufbauwunsch, der quittiert werden muss.
  - **FIN:** Einseitiger Verbindungsabbau und Ende des Datenstroms aus dieser Richtung. Auch hier muss quittiert werden.
- **Window:** Mit der Fenstergröße zur Flusskontrolle nach dem Fenster-Mechanismus steuert der Zielrechner den an ihn gerichteten Datenstrom im Quellrechner. Das Feld gibt an, wie viele Bytes (beginnend ab der Quittungsnummer) der Zielrechner in seinem Aufnahme-Puffer noch aufnehmen kann. Empfängt der Quellrechner ein TCP-Segment mit der Fenstergröße gleich 0, muss der Sendevorgang gestoppt werden. Die Fenstergröße kann die Effizienz der Übermittlung beeinflussen (Senderblockade). Die Ermittlung einer optimalen Fenstergröße gehört zu den wichtigsten Aufgaben bei der TCP/IP-Implementierung.
- **Checksum (Prüfsumme):** Diese Prüfsumme erlaubt es, den TCP-Header, die Daten und einen Auszug aus dem IP-Header (Pseudo Header), der an das TCP-Protokollmodul zusammen mit den Daten übergeben wird, auf das Vorhandensein von Fehlern (Bitfehler, Datenverlust) zu überprüfen. Bei Berechnung der Prüfsumme wird dieses Feld selbst als null angenommen.
- **Urgent Pointer (Urgent-Zeiger):** Das Protokoll TCP ermöglicht es, wichtige (dringliche) und meist kurze Nachrichten (z.B. Interrupts) den gesendeten normalen Daten hinzuzufügen und an Kommunikationspartner direkt zu übertragen. Damit können außergewöhnliche Zustände signalisiert werden. Derartige Daten werden hierbei als Urgent-Daten bezeichnet. Ist der Urgent-Zeiger gültig (URG-Flag = 1), so zeigt er auf das Ende von Urgent-Daten. Sie werden immer direkt nach dem TCP-Header übertragen. Erst danach folgen die normalen Daten.

- **Options:** Das TCP erlaubt es, Service-Optionen anzugeben. Das erste Byte im Optionsfeld legt den Optionstyp fest. In den RFCs 1323 und 2018 wurde das Optionsfeld in seiner Bedeutung wesentlich erweitert und hat folgende Bedeutung:
  - **Maximum Segment Size (MSS):** Diese Option wird beim Verbindungsaufbau genutzt, um den Kommunikationspartner mitzuteilen, welche maximale Segmentgröße verarbeitet werden kann.
  - **Window Scale (WSopt):** Mittels der Option WSopt können die Kommunikationspartner optional während der Initialisierung (also beim SYN-Segment) festlegen, ob die Größe des 16-Bit Fenster um einen konstanten Skalenfaktor multipliziert wird. Dieser Wert kann unabhängig für den Empfang (R) und das Versenden (S) von Daten ausgehandelt werden. Als Konsequenz dieses Verfahrens wird nun die Fenstergröße von der TCP-Instanz nicht mehr als 16 Bit-Wert sondern als 32 Bit-Wert aufgefasst. Der maximale Wert für den Skalenfaktor von WSopt beträgt 14, was einer neuen oberen Fenstergrenze von 1 GByte entspricht. Auf Grundlage des Skalenfaktors von WSopt wird auch der übertragene Fensterwert im TCP-Segment neu berechnet.
  - **Timestamps Option (TSopt):** Dieses Feld besteht aus den Teilen Timestamp Wert (TSval) und Timestamp Echo Reply (TSecr). Letzteres Feld ist nur bei ACK-Segment erlaubt. Mit diesen Informationen informieren sich die TCP-Instanzen über die Round Trip Time (RTT).
  - **Selective Acknowledgement (SACK):** Dieses Verfahren (RFC 2018) stützt sich wesentlich auf das Optionsfeld und es erlaubt, dieses Feld variabel zu erweitern.
  - Ergänzt werden die Optionen schließlich noch um den **Connection Count (CC)**, sowie **CC.NEW** und **CC-ECHO**, die bei der Implementierung T/TCP (RFC 1644) anzutreffen sind.
  - **Padding:** Füllzeichen ergänzen die Optionsangaben auf die Länge von 32 Bits.

TCP verhindert den gleichzeitigen Verbindungsaufbau zwischen zwei Stationen, d.h. nur eine Station kann den Aufbau initiieren. Es ist es nicht möglich, einen mehrfachen Aufbau einer Verbindung durch den Sender aufgrund eines Timeouts des ersten Verbindungsaufbauwunsches zu generieren. Der Datenaustausch zwischen zwei Stationen erfolgt nach dem Verbindungsaufbau. Gehen Daten bei der Übertragung verloren, wird nach Ablauf eines Timeouts die Wiederholung der fehlerhaften Segmente gestartet. Durch die Sequenznummer werden doppelt übertragene Segmente erkannt. Aufgrund der Sequenznummer ist es möglich, 2^{32}-1 Byte Daten (8 Gigabyte) pro bestehender Verbindung zu übertragen.

Die Flusskontrolle nach dem Fenster-Mechanismus (Window-Feld) erlaubt es einem Empfänger, dem Sender mitzuteilen, wieviel Pufferplatz zum Empfang von Daten zur Verfügung stehen. Ist der Empfänger zu einem bestimmten Zeitpunkt der Übertragung einer höheren Belastung ausgesetzt, kann er dies dem Sender über das Window-Feld bekanntgeben.

Jedes übertragene TCP-Segment unterliegt einer Zeitüberwachung (Retransmission Time); das bedeutet, dass ein Empfänger nach einer bestimmten Zeitdauer eine Quittung über die erhaltenen Segmente aussenden muss. Da diese Zeitdauer stark von der aktuellen Belastung des Netzes abhängt, muss der Retransmission Timer für jedes TCP-Segment neu berechnet und eingestellt werden.

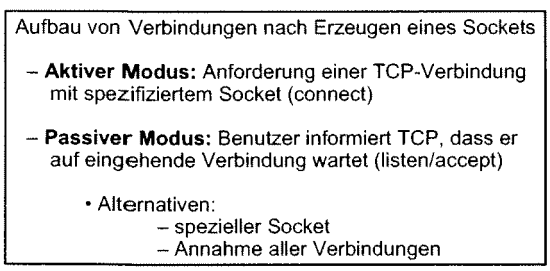
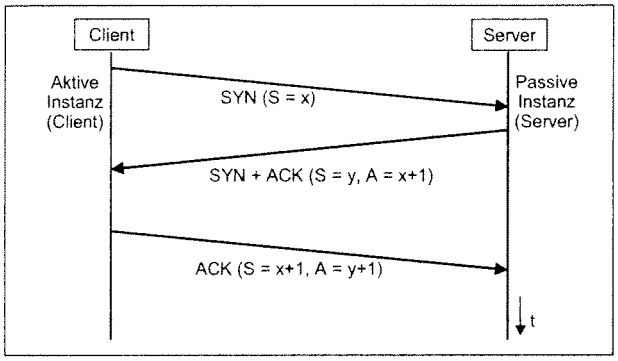


Bild: TCP-Verbindungsaufbau
(Aktiver und passiver Modus)

Auf- und Abbau von TCP-Verbindungen

Eine TCP-Verbindung ist vollduplex und als ein Paar von zwei unidirektionalen logischen Verbindungen gesehen werden kann. Die Kommunikationspartner befinden sich zum Anfang der Übertragung immer in folgenden Zuständen:

- **Passives Öffnen:** Eine Verbindung tritt in den Abhörstatus ein, wenn eine Anwendungsinstanz TCP mitteilt, dass sie Verbindungen für eine bestimmte Portnummer annehmen würde.
- **Aktives Öffnen:** Eine Anwendungsinstanz teilt TCP mit, dass es eine Verbindung mit einer bestimmten IP-Adresse und Portnummer eingehen möchte, was mit einer bereits abhörenden Anwendungsinstanz korrespondiert.



- 3-Wege-Handshake zum Verbindungsaufbau
- Austausch von Start-Sequenznummern
  (Möglicherweise wurde Verbindung mit gleichen Port-Nummern kürzlich beendet und es befinden sich noch Pakete im Netz.)

Bild: Öffnen einer TCP-Verbindung

Verbindungsaufbau

Der Verbindungsaufbau wird als Drei-Wege-Handshake bezeichnet, da drei TCP-Segmente auszutauschen sind. Der Sender wählt eine Sendefolgenummer (SEQ = x) und setzt das SYN-Flag. Der Empfänger antwortet - sofern er den Verbindungswunsch annehmen will - mit gesetzten Flags ACK und SYN, mit einer eigenen Sendefolgenummer (SEQ = y) und der Empfangsfolgenummer (ACK = x + 1). Diese gibt die nächste, vom Sender erwartete Sendefolgenummer an. Ein drittes Segment zeigt an, dass die Verbindung aufgebaut ist

Der Aufbau einer TCP-Verbindung läuft wie folgt ab. Die TCP-Protokoll-Instanz im Rechner A generiert ein TCP-Segment, in dem das Flag SYN gesetzt ist. Somit wird dieses Segment hier als SYN-Segment bezeichnet. Der Verbindungsaufbau beginnt damit, dass die beiden Kommunikationspartner einen Anfangswert für die jeweiligen Sequenznummern festlegen. Dieser Anfangswert für eine Verbindung wird als Initial Sequence Number (ISN) bezeichnet.

Detaillierte Beschreibung:

Die TCP-Instanz im Rechner A sendet dazu an den Rechner B ein SYN-Segment mit u.a. folgenden Informationen:

- SYN-Flag im TCP-Header wird gesetzt (SYN-Segment),
- frei zugeteilte Nummer des Quell-Ports,
- Zielport als Well-Known Port,
- SEQ: Sequenznummer der Quell-TCP-Instanz (hier SEQ = x).

Das gesetzte SYN-Bit bedeutet, dass die Quell-TCP-Instanz eine Verbindung aufbauen (synchronisieren) möchte. Mit der Angabe des Zielports (als Well-Known Port) wird die gewünschte Standardanwendung TCP im Rechner B gefordert.

Die Ziel-TCP-Instanz befindet sich im sogenannten Listenmodus, so dass sie auf ankommende SYN-Segmente wartet. Nach dem Empfang eines SYN-Segments leitet die Ziel-TCP-Instanz ihrerseits den Verbindungswunsch an den Ziel-Anwendungsprozess (z.B. den FTP-Prozess) gemäß der empfangenen Nummer des Zielports weiter und generiert eigene Initial Sequence Number (ISN) für die Richtung zum Rechner A.

Im zweiten Schritt des Verbindungsaufbaus sendet Rechner B ein TCP-Segment im mit folgendem Inhalt an den Rechner A:

- Die beiden Flags SYN und ACK im TCP-Header werden gesetzt,
- Die beiden Quell- und Ziel-Port-Nummern werden angegeben,
- Die Sequenznummer SEQ der Ziel-TCP-Instanz (hier SEQ = y) wird mitgeteilt.

Das ACK-Bit signalisiert, dass die Quittungsnummer in ACK-Bit diesem SYN/ACK-Segment von Bedeutung ist. Die Quittungsnummer ACK enthält die nächste von der TCP-Instanz im Rechner B erwartete Sequenznummer (SEQ = x +1).

Die TCP-Instanz im Rechner A bestätigt den Empfang des SYN/ACK-Segments mit einem ACK-Segment, in dem das ACK-Flag gesetzt wird. Mit der Quittungsnummer ACK = y + 1 wird der TCP-Instanz in Rechner B bestätigt, dass die nächste Sequenznummer SEQ = y+1 erwartet wird.

Eine TCP-Verbindung ist aus zwei unidirektionalen Verbindungen zusammensetzt. Jede dieser gerichteten Verbindungen wird im Quellrechner durch die Angabe der Ziel-IP-Adresse und von beiden Quell- und Zielports eindeutig identifiziert.

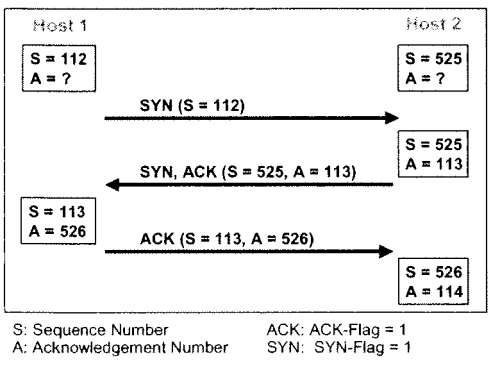


Bild: Öffnen einer TCP-Verbindung

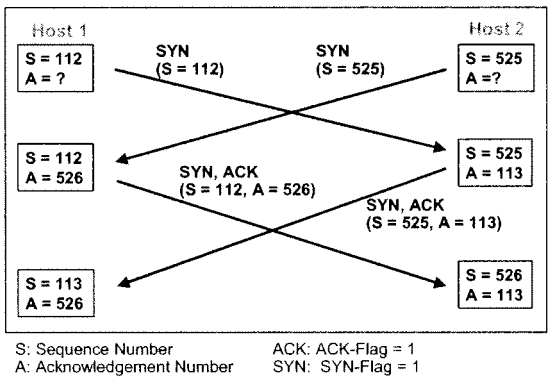


Bild: Simultanes Öffnen einer TCP-Verbindung

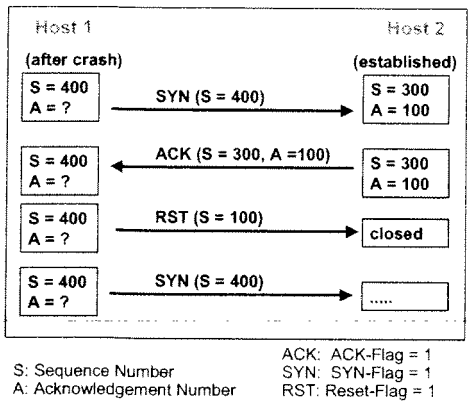


Bild: Reset während Öffnen einer TCP-Verbindung

Wurde eine TCP-Verbindung aufgebaut, so kann der Datenaustausch zwischen den kommunizierenden Anwendungsprozessen erfolgen, genauer gesagt zwischen den mit der TCP-Verbindung logisch verbundenen Ports.

Im Normalfall kann der Abbau einer TCP-Verbindung von beiden kommunizierenden Anwendungsprozessen initiiert werden. Da jede TCP-Verbindung sich aus zwei gerichteten Verbindungen zusammensetzt, werden diese gerichteten Verbindungen quasi nacheinander abgebaut. Jede TCP-Instanz koordiniert den Abbau seiner gerichteten Verbindung zu seiner Partner-TCP-Instanz und verhindert hierbei den Verlust von noch unquittierten Daten.

Die Beispiele zeigen verschiedene Möglichkeiten, die beim Verbindungsaufbau auftreten können.

Der Abbau wird von einer Seite mit einem TCP-Segment initiiert, in dem das FIN-Flag im Header gesetzt wird (FIN-Segment mit SEQ = x). Dies wird von der Gegenseite durch das ACK-Segment mit dem gesetzten ACK-Flag positiv bestätigt. Die positive Bestätigung erfolgt hier durch die Angabe der Quittungsnummer ACK = x + 1. Damit wird eine gerichtete Verbindung abgebaut. Der Verbindungsabbau in der Gegenrichtung wird entweder auf der gleichen Weise eingeleitet (wie im Bild) oder mit einem Segment, in dem die beiden FIN- und ACK-Flags gesetzt sind, begonnen (FIN/ACK-Segment). Nach der Bestätigung dieses FIN-Segments bzw. FIN/ACK-Segments durch die Gegenseite wird der Abbau-Prozess beendet.

Beim Abbau einer Verbindung tritt unter Umstände ein zusätzlicher interner Time-Out Mechanismus in Kraft. Die TCP-Instanz geht in den Zustand Aktive-Close, versendet ein abschließendes ACK und befindet sich dann im Status Time Wait. Dessen Zeitdauer beträgt zweimal die maximale Segment-Lebensdauer (Maximum Segment Lifetime, MSL), bevor die TCP-Verbindung letztlich geschlossen wird. TCP-Segmente, die länger als die MSL-Zeit im Netz unterwegs sind, werden verworfen. Der Wert von MSL beträgt bei heutigen TCP-Implementierungen in der Regel 120 Sekunden. Anschließend wird der Port freigegeben und steht über eine neue Initial Sequence Number (ISN) für spätere Verbindungen wieder zur Verfügung.

Conditions for closing a connection

- each side closes it's sending direction, receiving side may not be closed by receiver
- a sending side may only close a connection when all it's sent data have been acknowledged
- TCP informs application when it receives a close indication from the sending side

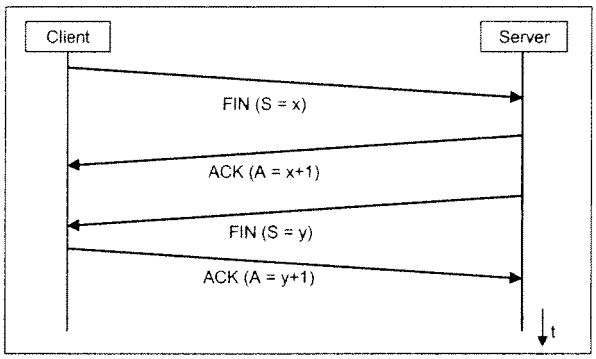
Three possible closing scenarios:

- application tells TCP to close the connection
- the remote TCP initiates close by sending a FIN control signal
- both users close simultaneously

Bild: Abbau einer TCP-Verbindung

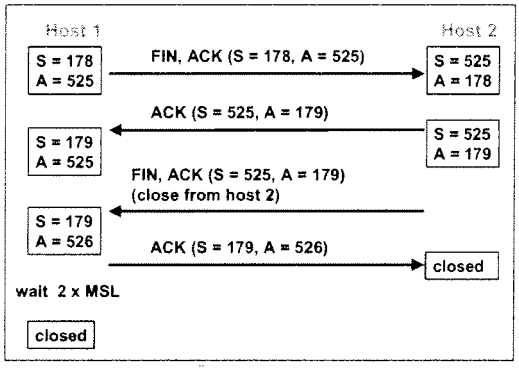
Verbindungsabbau

Der Datentransfer wird für beide Richtungen unabhängig voneinander beendet, was den Austausch von vier TCP-Segmenten erforderlich macht.



- Beide Seiten der Verbindung werden geschlossen
- Verfahren stellt sicher, dass alle gesendeten Daten vor Beenden der Verbindung ankommen

Bild: Abbau einer TCP-Verbindung



S: Sequence Number
A: Acknowledgement Number
MSL: Maximum Segment Lifetime
ACK: ACK-Flag = 1
FIN: FIN-Flag = 1

Bild: Abbau einer TCP-Verbindung

Datennaustausch

Für den Aufbau der Verbindung wird ein Handshake-Verfahren verwendet, bei dem mit drei Nachrichten beide Seiten der Kommunikation sich synchronisieren. Ziel ist der Austausch von Folgenummern (in den Feldern Sequence Number und Acknowledgement Number), mit deren Hilfe während der eigentlichen Kommunikationsphase dann verlorengegangene oder in der Reihenfolge vertauschte Segmente erkannt werden..

Die Acknowledgement Number ist der Kern des Verbindungsablaufes. Sie ist nur dann gültig, wenn das ACK-Flag gesetzt ist. Die Acknowledgement Number kennzeichnet die Bestätigung des korrekten Empfangs aller Bytes bis zu dem Byte mit der um 1 verminderten Acknowledgement Number, oder anders ausgedrückt: die Acknowledgement Number kennzeichnet das nächste Byte, auf das jetzt gewartet wird.

Üblicherweise wird auf der Sendeseite eine Zeitüberwachung aktiviert, die überprüft, ob in einem gewissen Zeitfenster die Bestätigung für das gesendete Segment ankommt. Falls nicht, dann wird einfach das Segment nochmals gesendet.

Es ist nicht notwendig, dass jedes Segment einzeln bestätigt wird. Die Window-Size gibt an, wie groß der Empfangspuffer ist, also wieviel Bytes der Empfänger zwischenspeichern kann. Der Sender darf nun so viele Bytes senden, bis dieser Puffer gefüllt ist, dann muss er auf eine Empfangsbestätigung warten. Der Empfänger kann in der Empfangsbestätigung auch gleich eine andere Fenstergröße mitteilen. Diese neue Fenstergröße kann größer oder kleiner sein, je nach Randbedingung auf der Empfangsseite. Sie kann auch auf null gesetzt werden, wenn der Empfänger jetzt eine Verarbeitungszeit braucht. Er muss dann, nachdem er wieder bereit ist, eine neue Fenstergröße mitteilen. Um eine Blockierung zu vermeiden, wenn der Empfänger vergisst, eine neue Fenstergröße mitzuteilen, prüft der Sender in regelmäßigen, exponentiell sich vergrößernden Abständen, ob der Empfänger wieder bereit ist. Übrigens hat dieser Prozess des sich verschiebenden und verändernden Fensters den Namen solcher Verfahren geprägt: Sliding Window.

Fenstergröße und Zeitüberwachung müssen in einem sinnvollen Zusammenhang stehen und an die physikalischen Verhältnisse der Übertragungsstrecke angepasst werden. So muss bei einer Satellitenverbindung aufgrund ihrer langen Laufzeit eine große Zeltüberwachung und ein großes Fenster eingestellt werden, sonst sinkt der Durchsatz erheblich. Heutzutage werden die Timer und die Window-Size dynamisch festgelegt. Dazu wird die Laufzeit bestimmt.

TCP ist also verbindungsorientiert, allerdings nur auf der TCP-Schicht, also Ende-zu-Ende. Es ist wichtig hier darauf hinzuweisen, dass mit dieser Art Verbindung keine Reservierung von Ressourcen im Netz verbunden ist. Durch Überlast im Netz wird TCP zur Verminderung der Datenrate veranlasst. TCP kann keine Aktionen im Netz veranlassen.

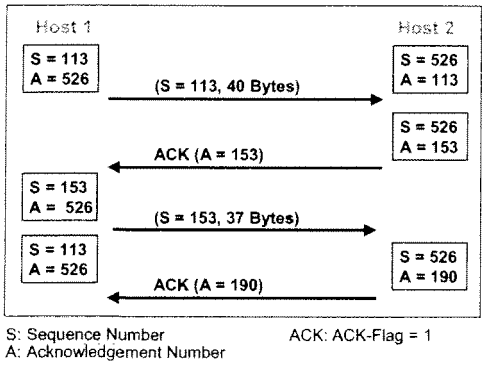


Bild: Daten Transfer in TCP

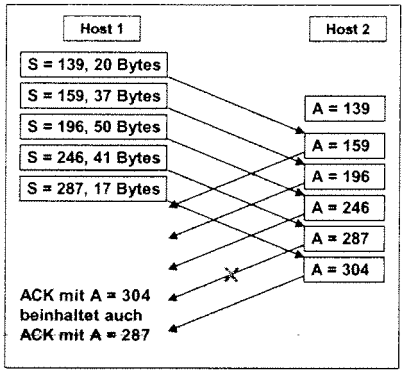


Bild: Kumulative Quittungen

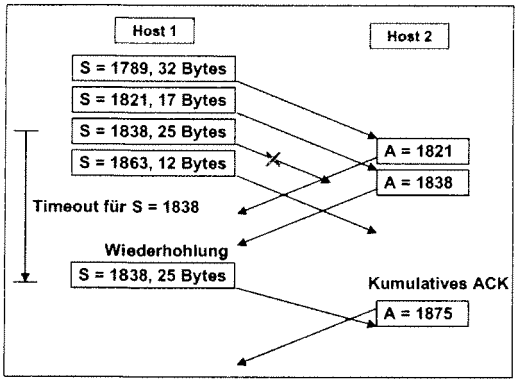


Bild: Wiederholung: Segment verloren gegangen

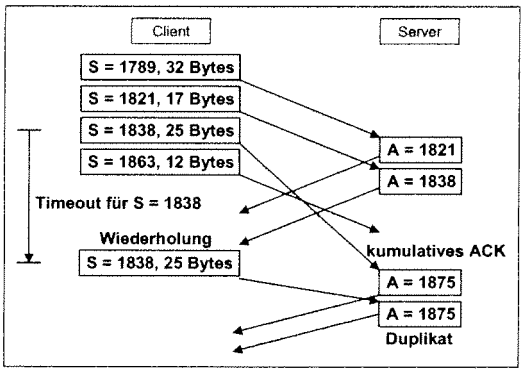


Bild: Wiederholung: Quittungsverzögerung

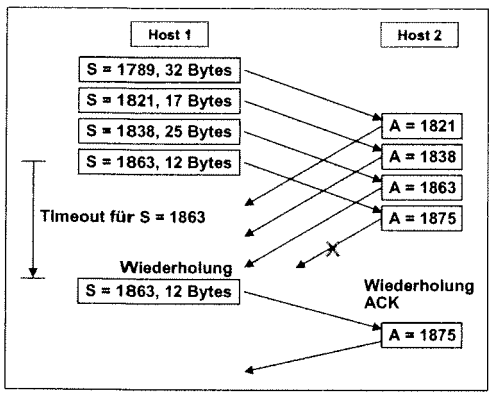


Bild: Wiederholung: Quittung verloren gegangen

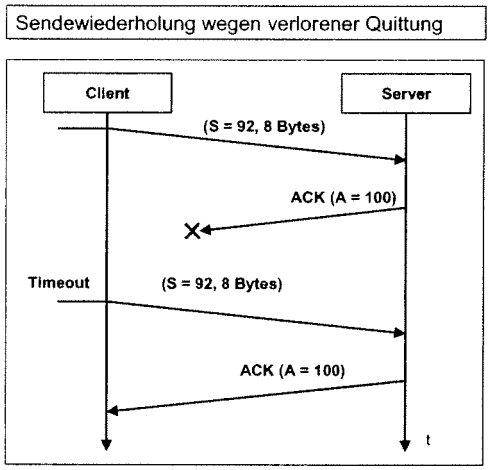


Bild: Wiederholung: Quittung verloren gegangen

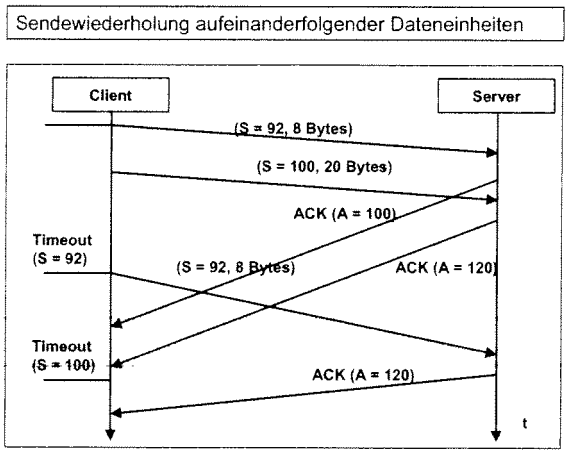


Bild: Sendewiederholung von zwei Segmenten

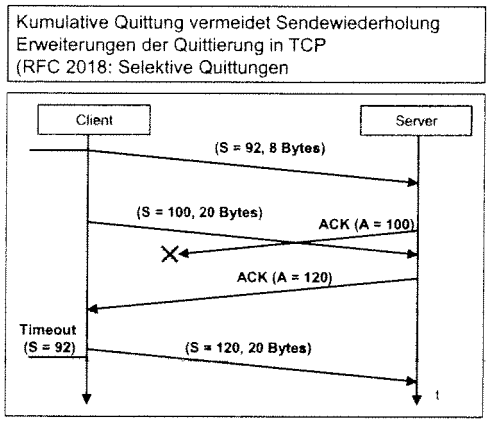


Bild: Vermeidung der Wiederholung durch Kumulative Quittung

Das Transmission Control Protocol (TCP) als zuverlässiges, verbindungsorientiertes Transportprotokoll bietet:

- Auf- und Abbau von Verbindungen auf der TCP-Schicht (nur Punkt-zu-Punkt, Ende-zu-Ende, Vollduplex),
- Flusskontrolle (Ende-zu-Ende),
- Reihenfolgesicherung,
- Zeitüberwachung,
- Prüfsummenbildung.

Die zu übertragenden Daten werden in TCP-Segmente variabler Länge aufgeteilt und mit einem mindestens 20 Byte umfassenden Protokollkopf versehen. Die Obergrenze der Segmentgröße liegt bei 65535 Byte. Allerdings ist es empfehlenswert, das TCP-Segment so klein zu machen, dass eine Fragmentierung auf der IP-Schicht nicht notwendig ist.

- Quittierungsstrategien
  - sofort
  - kumulativ
  - optional: selektiv
- Übertragungswiederholung nach ausbleibender Quittung
  - Standard: Go-Back-N
  - optional: selektive Übertragungswiederholung

Bild: TCP-Fehlerbehebung

- Flusskontrolle regelt den Datenfluss zwischen Endsystemen.
- Flusskontrolle in TCP mit Fenstermechanismus
  - Acknowledgment-Feld bestätigt Empfang aller niedrigeren Sequenznummern.
  - AdvertisedWindow-Feld gibt an, wieviele Bytes der Empfänger zusätzlich akzeptiert.
  - Empfänger erlaubt dem Sender das Senden von Daten bis Acknowledgment + AdvertisedWindow.

Bild: TCP-Flusskontrolle

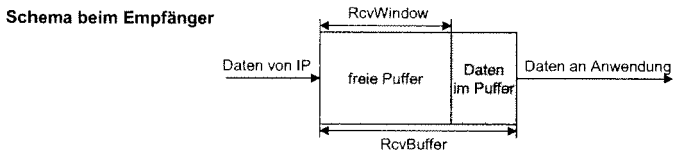


Bild: TCP-Flusskontrolle

Empfangsfenster

- gibt an, wieviel Pufferplatz der Empfänger für diese Verbindung zur Verfügung hat (explizite Kreditvergabe)
  - Feld Empfangsfenster im Kopf der TCP-Dateneinheit
  - RcvWindow
- kann dynamisch geändert werden

Szenario: Endsystem A schickt große Datei über TCP an Endsystem B

Variablen beim Empfänger

- LastByteRead: letzte Sequenznummer, die von der Anwendung aus dem Empfangspuffer gelesen wurde
- LastByteRcvd: letzte Sequenznummer, die über das Netz empfangen und in den Empfangspuffer geschrieben wurde

Es muss immer gelten: LastByteRcvd – LastByteRead ≤ RcvBuffer

Empfangsfenster: RcvWindow = RcvBuffer – (LastByteRcvd – LastByteRead)

Variablen beim Sender

- LastByteSent: letzte Sequenznummer, die gesendet wurde
- LastByteAcked: letzte Sequenznummer, die quittiert wurde

Es muss immer gelten: LastByteSent – LastByteAcked ≤ RcvWindow

Bild: TCP-Flusskontrolle

Flusskontrolle beim Protokoll TCP

Bei der Datenkommunikation entsteht das Problem, dass die Menge der Übertragenen Daten an die Aufnahmefähigkeiten des Empfängers angepasst werden muss. Die übertragene Datenmenge sollte nicht größer sein als die Datenmenge, die der Empfänger aufnehmen kann. Die Menge der übertragenen Daten muss zwischen den kommunizierenden Rechnern entsprechend abgestimmt werden. Diese Abstimmung bezeichnet man oft als Flusskontrolle (Flow Control). Die Flusskontrolle bei der Datenübermittlung über eine TCP-Verbindung erfolgt nach dem Prinzip des Sliding Windows.

Für die Zwecke der Flusskontrolle nach dem Sliding-Window-Prinzip dienen folgende Angaben (im TCP-Header):

- Sequence Number (Sequenznummer),
- Acknowledgement Number (Quittungs- bzw. Bestätigungsnummer),
- Window-Größe.

Mit der Sequenznummer werden die zu sendenden TCP-Segmente fortlaufend nummeriert. Die Sequenznummer im TCP-Header eines Datensegments stellt dessen laufende Nummer in der gesendeten Segmentreihe dar. Mit der Quittungsnummer teilt der Empfänger dem Sender mit, welche Sequenznummer als nächste bei ihm erwartet wird. Seitens des Senders stellt die Window-Größe die maximale Anzahl der Datensegmente dar, die der Sender absenden darf, ohne auf eine Quittung vom Empfänger warten zu müssen. Seitens des Empfängers kann die Window-Größe als die maximale Anzahl der Datensegmente gesehen werden, die beim Empfänger immer aufgenommen werden. Wird die maximale Länge von TCP-Datensegmenten festgelegt, so kann die übertragene Datenmenge, bzw. die Menge von Daten unterwegs, mit den erwähnten drei Parametern (Sequenz- und Quittungsnummer sowie Window-Größe) immer kontrolliert werden.

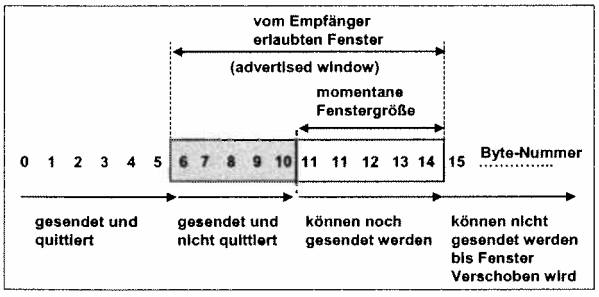


Bild: TCP-Fenstermechanismus

TCP Sliding-Window-Prinzip

Beim Protokoll TCP wird eine modifizierte Version des Sliding-Window-Prinzips verwendet. Diese Modifikation besteht darin, dass die Window-Größe in Bytes (und nicht in der Anzahl der Segmente) angegeben wird.
Damit legt die Window-Größe die maximale Anzahl von Bytes fest, die die Sendeseite absenden darf, ohne auf eine Quittung vom Ziel warten zu müssen.

Mit dem Parameter Window wird ein Bereich von Nummern markiert, die den zu sendenden Datenbytes zuzuordnen sind. Dieser Bereich kann als Sendefenster gesehen werden. Es sind vier Bereiche im Strom von Datenbytes zu unterscheiden:

- Datenbytes, die abgesendet und bereits vom Zielrechner positiv quittiert wurden.
- Datenbytes, die abgesendet und vom Zielrechner noch nicht quittiert wurden.
- Datenbytes, die noch abgesendet werden dürfen, ohne auf eine Quittung warten zu müssen.
- Datenbytes außerhalb des Sendefensters. Diese Datenbytes dürfen erst dann abgesendet werden, wenn der Empfang von einigen vorher abgeschickten Daten bestätigt wird.

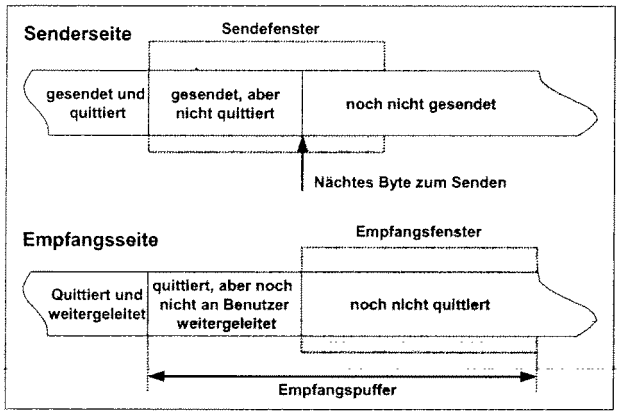


Bild: Flusskontrolle mit Fenstermechanismus

Da IP zu den ungesicherten Protokollen gehört, muss TCP über Mechanismen verfügen, die in der Lage sind, mögliche Fehler (z.B. Verlust von IP-Paketen, Verfälschung der Reihenfolge usw.) zu erkennen und zu beheben. Der Mechanismus der Fehlerkorrektur von TCP ist einfach: Wenn für ein abgesendetes Datensegment nicht innerhalb eines bestimmten Zeitraums eine Bestätigung eingeht, wird die Übertragung des Segments wiederholt.

Zu diesem Zweck gibt es beim Empfänger auch ein Fenstermechanismus.

Im Unterschied zu anderen Methoden zur Fehlerkontrolle kann der Empfänger zu keinem Zeitpunkt eine wiederholte Übertragung erzwingen. Dies liegt zum Teil daran, dass kein Verfahren vorhanden ist, um negativ zu quittieren, so dass keine wiederholte Übertragung von einzelnen Segmenten direkt veranlasst werden kann. Der Empfänger muss einfach abwarten, bis das von vornherein festgelegte Zeitlimit (Maximum Segment Lifetime MSL) auf der Sendeseite abgelaufen ist und infolgedessen bestimmte Daten nochmals übertragen werden.

Die maximale Wartezeit auf die Quittung ist ein wichtiger Parameter des Protokolls TCP. Er hängt von der zu erwartenden Verzögerung im Netz ab. Die Verzögerung im Netz kann durch die Messung der Zeit, die bei der Hin- und Rückübertragung zwischen dem Quell- und Zielrechner auftritt, festgelegt werden. Diese Zeit wird als Round Trip Time (RTT) bezeichnet. In Weitverkehrsnetzen, in denen die Satellitenverbindungen eingesetzt werden, kann es einige Sekunden dauern, bis eine Bestätigung ankommt. Es ist schwierig, die Verzögerungsrate eines Netzes im vorhinein zu wissen.

- **TCP ist ein Protokoll mit einem verschiebbaren Fenstermechanismus**
  - Bei Fenstergröße W können bis zu W Bytes ohne Quittung gesendet werden. Bei Datenquittierung verschiebt sich das Fenster weiter.
- **Jedes TCP-Segment enthält die Angabe einer Fenstergröße**
  - Sie gibt an, wieviele Bytes im Empfängerpuffer Platz haben.
  - Ursprünglich wurde immer die Gesamtfenstergröße gesendet
  - Heute wird sie durch die Congestion-Kontrolle limitiert.
- **Flusskontrolle:** Vermeidung Paketverlust am Empfangspuffer.
- **Überlastkontrolle:** Vermeidung von unnötige Übertragungswiederholung bei verzögerten Quittungen.
- **Flusskontrolle und Congestion-Kontrolle sind getrennt zu betrachten**

Bild: TCP-Flusskontrolle

Im Laufe einer Verbindung können zudem durch Netzbelastung bedingte Schwankungen von RTT auftreten. Daher ist es nicht möglich, einen festen Wert für die maximale Wartezeit auf die Quittung einzustellen. Wenn ein zu kleiner Wert gewählt wurde, läuft die Wartezeit ab, bevor eine Quittung eingehen kann. Infolgedessen wird das Segment unnötig erneut gesendet. Wird ein zu hoher Wert gewählt, hat dies lange Verzögerungspausen zur Folge, da die gesetzte Zeitspanne abgewartet werden muss, bevor eine Übertragungsüberholung stattfinden kann. Der Verlust eines Segments kann in diesem Fall den Datendurchsatz erheblich senken.

Zur effizienten Nutzung des Fenstermechanismus stehen zwei Parameter zur Verfügung, die zwischen den TCP-Instanzen im Verlauf der Kommunikation dynamisch angepasst werden:

Die **Window Size (WSIZE)** ist zu interpretieren als die Größe des TCP-Empfangspuffers in Byte. Aufgrund des maximal 2^{16}-1 großen Feldes im TCP-Header kann dieses maximal einen Wert von 65 535 Byte, d.h. rund 64 KByte (K = 1024) aufweisen. Moderne TCP-Implementierungen nutzen allerdings die in den TCP-Segmenten vorgesehene Option des WSopt, so dass nun Werte bis 2^{30}, also rund 1 GByte möglich sind. Die Window Size, ein Konfigurationsparameter der TCP-Implementierung, ist üblicherweise auf einen Wert von 4, 8, 16 oder 32 KByte initialisiert. Beim Verbindungsaufbau teilt die TCP-Empfängerinstanz ihre Windowsize dem Sender mit, was als Advertised Window Size (advWind) bezeichnet wird.

Die **Maximum Segment Size (MSS)** als der maximale Wert des TCP-Sendepuffers stellt das Gegenstück zur Window size (WSIZE) dar. Für übliche TCP-Implementierungen gilt die Ungleichung MSS < WSIZE. Die dynamische Aushandlung dieser Parameter zusammen mit der Methode der Bestimmung der sog. Round Trip Time (RTT) begründet ursächlich das gute Übertragungsverhalten von TCP auf sehr unterschiedlichen Netzen.

Ereignis	Reaktion des TCP-Empfängers
Ankunft einer Dateneinheit in Reihenfolge. Alle Daten davor bereits quittiert. Keine Lücken.	Verzögerte Quittung. Bis zu 500 ms warten auf weitere reihenfolgetreue Dateneinheit. Wird keine weitere empfangen, dann Senden der Quittung.
Ankunft einer Dateneinheit in Reihenfolge. Alle Daten davor bereits quittiert. Bereits eine weitere reihenfolgetreue auf Quittung wartende Dateneinheit. Keine Lücken.	Sofortiges Senden einer kumulativen Quittung. Beide Dateneinheiten werden quittiert.
Ankunft einer Dateneinheit mit einer nicht erwarteten höheren Sequenznummer. Erkennen einer Lücke.	Sofortiges Versenden einer duplizierten Quittung mit der Sequenznummer des nächsten erwarteten Bytes.
Ankunft einer Dateneinheit, die teilweise oder komplett eine Lücke bei den empfangenen Daten auffüllt.	Sofortiges Versenden einer Quittung, falls die Dateneinheit an der unteren Schranke der Lücke beginnt.

Falls sendebereite Daten beim Empfänger zur Verfügung stehen, werden Quittungen per Piggyback gesendet

Bild: Generierung von TCP-Quittungen

Die wichtigsten in TCP verwendeten Verfahren und Algorithmen für Zuverlässigkeit, Fluss- und Überlaststeuerung sind:

- **Slow-Start-Algorithmus:** Die von TCP verwendete Fenstergröße wird am Anfang einer Verbindung klein gewählt und sukzessive erhöht, bis die Raten der gesendeten Segmente und der empfangenen Quittungen gleich werden.
- **Congestion-Avoidance-Algorithmus:** TCP geht davon aus, dass Segmente durch **Überlast** (congestion) im Netz verloren gehen. Quittungen, die innerhalb der Zeitspanne RTO nicht eintreffen, führen - genauso wie duplizierte Quittungen - zu einer Reduktion der Senderate.
- **Nagle-Algorithmus:** Interaktive Anwendungen führen zu kurzen TCP-Segmenten, wodurch der Overhead-Anteil (Header) dieser Segmente überwiegt. Verzögerte Bestätigungen erlauben es dem Empfänger eines Segments mit einer Quittung zu warten, um diese zusammen mit eigenen Nutzdaten zu übertragen. Der Nagle-Algorithmus sammelt alle Nutzdaten bis zur nächsten fälligen Quittung, um diese dann in einem Segment zu übertragen
- **Karn-Algorithmus:** Die Schätzung der RTT kann falsche Werte ergeben, wenn dieselben Segmente wiederholt gesendet wurden. Der Karn-Algorithmus verbessert die Genauigkeit der RTT-Schätzung, indem solche Fälle nicht in die Berechnung eingehen. Hingegen wird dann der RTO-Wert erhöht (Retramission Time-Out).
- **Das Silly Window Syndrom** bezeichnet eine Situation, in der ein Empfänger wiederholt eine kleine Fenstergröße angibt und der Sender entsprechend kleine Segmente sendet. Dadurch wird die Übertragungsstrecke schlecht ausgenutzt. Durch geeignetes Verhalten von TCP-Sender und -Empfänger kann dieses Problem vermieden werden.

Die Flusssteuerung wird mit Hilfe eines Schiebefensterprotokolls (sliding window) realisiert. Quittierung und Zuweisung eines Fensters sind jedoch entkoppelt, im Gegensatz zu LLC, HDLC und X.25. Die Fenstergröße hat einen wichtigen Einfluss auf den Durchsatz. TCP arbeitet mit positiven Quittungen, die Summenquittungen darstellen. Dies bedeutet, dass ab dem ersten, nicht quittierten Segment alle Segmente wiederholt werden, d. h. es gibt keine selektive Wiederholung.

Die Überlaststeuerung (congestion control) in TCP ist ein schwieriges Problem. Gründe dafür liegen in dem verbindungs- und zustandslosen Verhalten von IP, das keine Ansatzpunkte für die Erkennung und Steuerung von Überlast bietet. TCP leistet eine Flusssteuerung nur Ende-zu-Ende, so dass hier nur indirekt auf Überlast geschlossen werden kann. Zudem haben die verschiedenen TCP-Instanzen keine Möglichkeit, zwecks Überlaststeuerung zu kooperieren. Damit muss für die Überlaststeuerung auf den Fenstermechanismus zurückgegriffen werden.

Timer

In TCP werden die Phasen Verbindungsaufbau, Verbindungsabbau und die eigentliche Datenübertragung von **Timern** (Zeitgeber, Wecker) gesteuert und überwacht. Dadurch ist TCP in der Lage, sich an Veränderungen im Netz anzupassen.

Problem
- Auf welchen Wert soll der TCP-Timeout gesetzt werden?

Beobachtungen
- Wert sollte sich an der Umlaufzeit (Round Trip Time, RTT) orientieren und etwas größer sein

Umlaufzeit variiert
- zu kurzer Wert: unnötige Sendewiederholungen treten auf
- zu langer Wert:langsame Reaktion auf Verlust der Dateneinheit

Bild: Umlaufzeit und Timeout

Eine wichtige Größe ist die Segmentumlaufzeit (RTT, Round Trip Time), die vom Empfänger als Zeitdifferenz zwischen dem Senden eines Segments und dem Empfang der zugehörigen Bestätigung gemessen werden kann. Da die Werte in Abhängigkeit der aktuellen Netzbelastung stark schwanken können, wird ein gleitender Mittelwert (SRTT, Smoothed RTT) gebildet.

Aus diesem wird RTO (Retransmission Time Out) berechnet, das Zeitintervall nach dessen Ablauf und Ausbleiben einer Quittung ein Segment erneut gesendet wird. Wenn ein wiederholtes Segment nach dem Ablauf von RTO ebenfalls nicht quittiert wird, wird der Wert von RTO vergrößert. Dieser Vorgang kann sich mehrfach wiederholen, bis der Algorithmus eine Verbindung als unterbrochen erklärt.

Messung der Umlaufzeit
Timer (Granularität variiert, bis zu 500 ms) wird benutzt. Beim Ablauf wird ein Zähler jeweils inkrementiert.

RTT_{Sample}
- Gemessene Zeit vom Versenden der Dateneinheit bis zum Empfang der dazugehörigen Quittung.
- Sendewiederholungen werden ignoriert.
- Kumulativ quittierte TCP-Dateneinheiten werden nicht betrachtet.

Glätten des gemessenen Wertes
- Gemessener Wert kann stark schwanken
- Es wird ein auf mehreren Messungen basierender Wert herangezogen: **$RTT_{Estimate}$**

Bild: Messung der Umlaufzeit

Bitrate	Bitrate x Verzögerung (RTT = 100 ms)
T1 (1.5 Mbit/s)	18 KB
Ethernet (10 Mbit/s)	122 KB
FDDI (100 Mbit/s)	549 KB
STM-1 (155 Mbit/s)	1.2 MB
STM-3 (622 Mbit/s)	1.8 MB
STM-16 (2.5 Gbit/s)	29.8 MB

• **Probleme**
– kein effizienter kontinuierlicher Datenfluss
– Sequenznummernbereich läuft schnell über
– Advertised Window (16 Bit) erlaubt Fenster von 64 KB

Bild: Rate-Verzögerungsprodukt

Nach jedem erfassten Wert für RTT_{Sample} wird der geglättete Wert der Umlaufzeit folgendermaßen bestimmt:

$RTT_{Estimate} = a * RTT_{Estimate} + (1-a) * RTT_{Sample}$

- Exponential Weighted Moving Average (EWMA)
- Neue Werte werden niedriger gewichtet als alte Werte
- Einfluss eines gemessenen Wertes sinkt exponentiell
  Typischer Wert für a = 0.875

- $RTT_{Estimate} = 0.875 * RTT_{Estimate} + 0.125 * RTT_{Sample}$

Bild: Bestimmung der Umlaufzeit

Sollte auf einen etwas höheren Wert gesetzt werden als $RTT_{Estimate}$

Bei hohen Schwankungen von $RTT_{Estimate}$ sollte ein höherer Sicherheitszuschlag gegeben werden

Timeout = $RTT_{Estimate}$ + 4 x Deviation
Deviation = a * Deviation + (1- a) * | RTT_{Sample} – $RTT_{Estimate}$ |

a = 0.875
Deviation = 0.875 * Deviation + 0.125 * | RTT_{Sample} – $RTT_{Estimate}$ |

Bild: Bestimmung des Retransmission-Timeout-Werts

Neben dem Retransmission Timer verfügt TCP noch über eine Reihe weiterer Timer:

• **Persist Timer**
- falls Fenstergröße auf Null ist und Quittungen verloren gehen, kann es zu einem Deadlock kommen
- Persit-Timer initiiert regelmäßige Nachfragen nach der Fenstergröße, auch wenn der Empfänger sein Empfangsfenster schließt
- TCP exponential backoff zur Berechnung der Timer-Werte

• **Keepalive Timer**
- Erkennt, wenn der Partner Probleme hat
  über eine TCP-Verbindung im Idle-Zustand fließen keine Daten
- ist nicht Bestandteil der TCP-Spezifikation, stellt eine Option dar
- kann dazu führen, dass bestehende Verbindungen terminieren (z.B. bei einem temporären Problem auf der Vermittlungsschicht)
- kann Servern helfen, Ressourcen nicht unnötig zu belegen, falls Client abgestürzt ist

• **2MSL-Timer**
- Misst die Zeit, die eine Verbindung im TIME_WAIT-Zustand verbringt

Bild: Timer in TCP

- **TCP ist ein Protokoll mit einem verschiebbaren Fenstermechanismus**
  - Bei Fenstergröße W können bis zu W Bytes ohne Quittung gesendet werden. Bei Datenquittierung verschiebt sich das Fenster weiter.
- **Jedes TCP-Segment enthält die Angabe einer Fenstergröße**
  - Sie gibt an, wieviele Bytes im Empfängerpuffer Platz haben.
  - Ursprünglich wurde immer die Gesamtfenstergröße gesendet
  - Heute wird sie durch die Staukontrolle limitiert.
- **Flusskontrolle:** Vermeidung Paketverlust am Empfangspuffer.
- **Staukontrolle:** Vermeidung von unnötige Übertragungswiederholung bei verzögerten Quittungen.
- **Flusskontrolle und Staukontrolle sind getrennt zu betrachten**

Bild: TCP: Fluss- und Staukontrolle

- Segment: Daten- oder Kontroll-Einheit
- Maximum Segment Size (MSS): maximal erlaubte Segmentgröße
- Full-size Segment: Datensegment mit MSS Byte
- Receiver Window (rwnd): letzter Wert des Empfangsfensters am Sender
- Congestion Window (cwnd): Maximale Datenlimite am Sender in Byte
- Initial Window (IW): Congestion Window beim Verbindungsaufbau
- Loss Window (LW): Congestion Window nach Verlustfeststellung eines Segments durch den Retransmission Timer
- Restart Window (RW): Congestion Window bei Beginn der Übertragungswiederholung

Bild: TCP Staukontrolle: Definitionen

- AdvertisedWindow = MaxReceiveBuffer - (LastByteReceived - LastByteRead)
- MaxWindow = min (CongestionWindow, AdvertisedWindow)
- EffectiveWindow = MaxWindow - (LastByteSent - LastByteAcked)

Bild: Integration von Stau- und Flusskontrolle

- Staukontrolle befasst sich mit Stausituationen in Zwischensystemen.
- Stau in Zwischensystemen führt zu Übertragungswiederholungen durch Transportprotokoll
  ⇒ Verstärkung der Stausituation

Ansätze in TCP

- Adaption des TCP-Sendefensters
  - Erhöhen des Sendefensters nach Erhalt einer Quittung
  - exponentiell durch Verdopplung des Fensters am Anfang der Verbindung oder bei Rücksetzen nach längerer Stausituation (**slow start**)
  - linear im Sättigungsbereich, d.h. Erhöhen des Fensters um 1 Paketgröße (**linear increase**)
    - Reduzieren der Fenstergröße nach Ausbleiben einer Quittung (Zeitüberwachung) auf die Hälfte (multiplicative decrease).
  - Explicit Congestion Notification (RFC 2481)

Bild: TCP-Staukontrolle

TCP bietet eine Reihe von Mechanismen zur Staukontrolle an:

- Slow start,
- Congestion avoidance,
- Fast retransmit,
- Fast recovery.

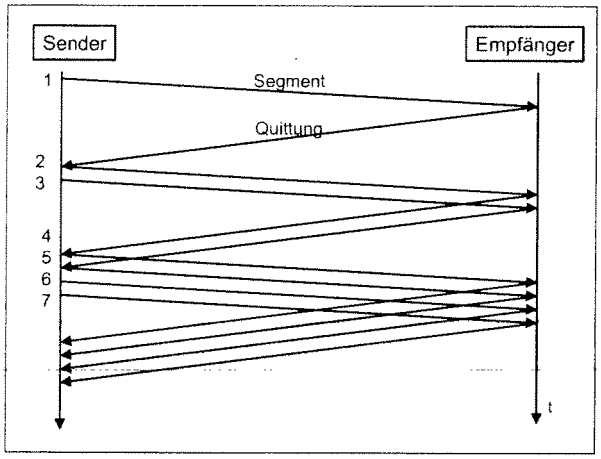


Bild: TCP Slow Start

Slow Start und Congestion Avoidance
Diese Funktionen dienen der Steuerung und Kontrolle der in ein Netz injizierten Daten. Dazu werden keine weiteren Protokoll-Elemente im TCP-Overhead, sondern die Betriebszustände, in denen sich Sender und Empfänger befinden können, werden erweitert. Die notwendigen Informationen sind:

Maximale Größe des Sendepuffers, diese kennt der Sender selbst (congestion window - cwnd).

- Es muss immer gelten
  LastByteSent – LastByteAcked ≤ min {rwnd, cwnd}
- Schwellenwert ssthresh bestimmt wie Staukontrollfenster vergrößert wird

Ablauf

- Start: cwnd = 1 MSS
- Solange cwnd ≤ ssthresh und Quittungen vor Timeout empfangen

Slow-Start:
- Exponentielles Erhöhen des Staukontrollfensters
- Verdopplung des Staukontrollfensters je Umlaufzeit
- Empfangene Quittung: cwnd + = 1

cwnd > ssthresh und Quittungen vor Timeout empfangen

Congestion Avoidance:
- Lineares Erhöhen des Staukontrollfensters
- Erhöhen des Staukontrollfensters um 1 je Umlaufzeit
- Empfangene Quittung: cwnd + = 1 / cwnd

Timeout:
- ssthresh := cwnd / 2
- cwnd = 1 MSS

Bild: TCP Staukontrolle: Slow-Start und Congestion Avoidance

Aktuelle Größe des Empfangspuffers, diesen Wert teilt der Empfänger im TCP-Protokoll-Element Window Size dem Sender mit (receiver's advertised window - rwnd).

Schwellwert, an dem die Betriebsart von Slow Start auf Congestion Avoidance umgeschaltet wird. Diesen Schwellwert kennt der Sender selbst; er wird üblicherweise auf die Hälfte des Wertes für congestion window gesetzt (slow start threshold - ssthresh).

Congestion Avoidance

- Additive increase, multiplicative decrease
  - Erhöhen des Fensters um 1 je Umlaufzeit
  - Erniedrigen des Fensters um Faktor 2 bei vermutetem Datenverlust (Quittung nicht rechtzeitig empfangen)
- Algorithmus zur Staukontrolle geht auf Van Jacobson zurück
- Modifikationen des ursprünglichen Algorithmus
- RFC 2581
- Staukontrollfenster wird initial auf 2 MSS gesetzt

Bild: Congestion Avoidance

Wenn ein Sender anfängt, Daten ins Netz zu schicken (also nach dem TCP-Verbindungsaufbau), dann kennt er die Fähigkeiten des Netzes noch nicht. Deshalb fängt er mit wenig Daten an um das Netz zu testen. Allgemein gilt, dass nicht mehr als zwei Segmente gesendet werden. Wurden diese bestätigt, schickt der Sender eine größere Datenmenge. Damit tastet er sich an den Schwellwert (slow start threshold), der willkürlich festgelegt ist. Danach wird in den congestion avoidance Modus umgeschaltet. Dabei wird jetzt die gesendete Datenmenge linear immer um ein Segment erhöht.

Die absolute Obergrenze, ab der nicht mehr erhöht wird, ist entweder durch den Sender (congestion window) oder durch den Empfänger (receiver's advertised window) vorgegeben.

Es wird immer geprüft, ob die Bestätigung eines Segmentes vor Ablauf des Timers ankommt. Tritt ein Timeout ein, dann wird mit dem Vorgang neu begonnen, wobei jetzt der Schwellwert halbiert wird. Durch geeignete Wahl der Parameter können unterschiedliche Verhaltensmuster erzielt werden. Es ist Aufgabe der Implementierung, hier das Bestmögliche aus dem Netz herauszuholen.

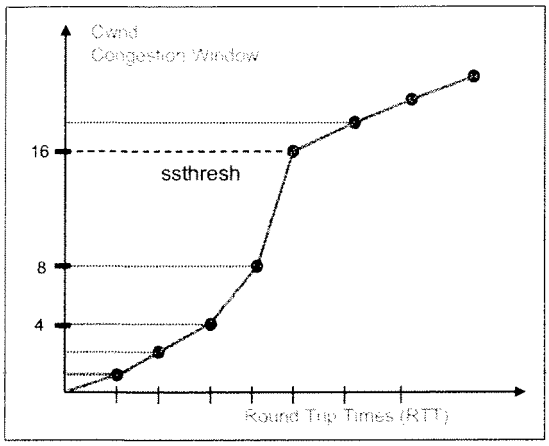
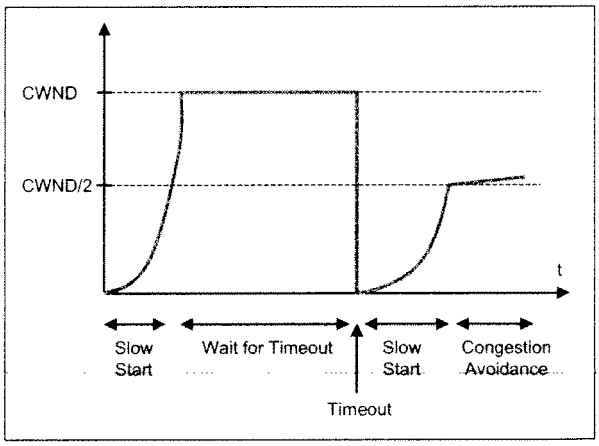


Bild: TCP Slow Start und Congestion Avoidance



cwnd Congestion Window

Bild: TCP Slow Start und Congestion Avoidance

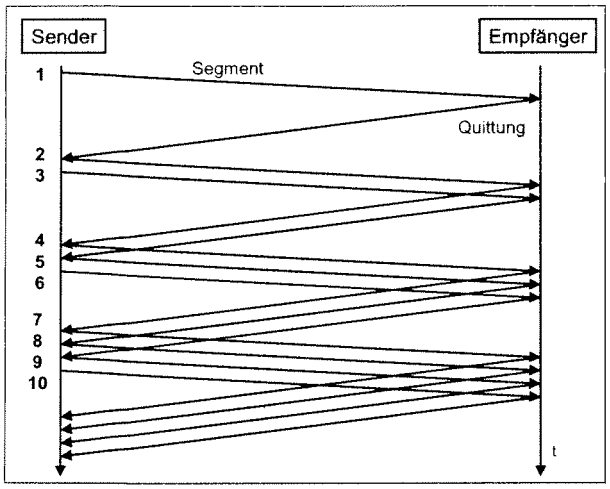


Bild: Additive Increase

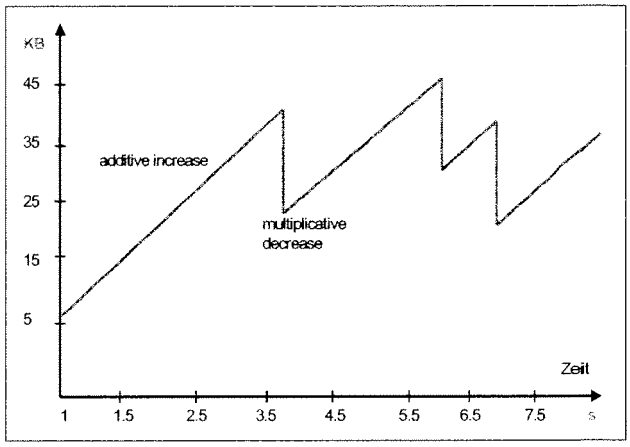


Bild: TCP-Sägezahn-Verlauf:
Additive Increase, Multiple Decrease

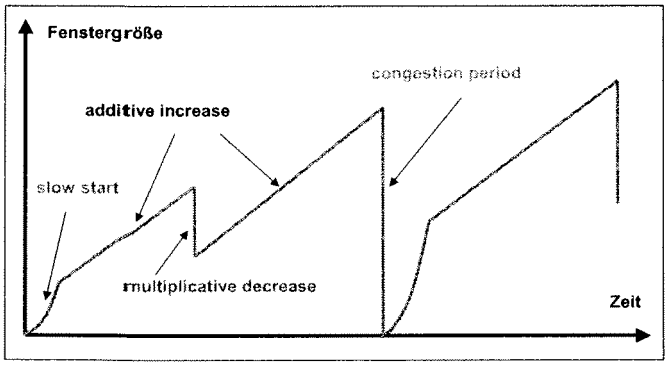


Bild: TCP-Ablaufphasen: Slow Start, Additive Increase, Multiplicative Decrease und Congestion

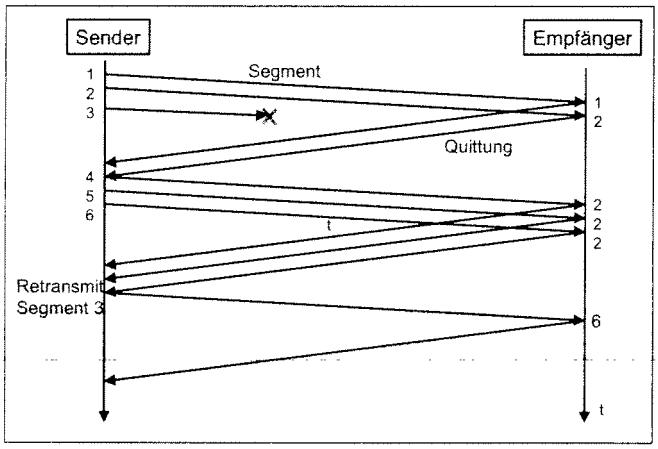


Bild: TCP Fast Retransmit

Fast Retransmit und Fast Recovery

Falls ein Segment verloren geht oder ein unerwartetes Segment empfangen wird, dann erkennt der Sender dies im Regelfall durch die Zeitüberwachung: es kommt keine oder eine falsche Bestätigung.

Dieser Vorgang kann beschleunigt werden, wenn der Empfänger sofort nach Empfang eines Segments, das nicht erwartet wurde (erkennbar durch eine falsche Sequence-Number) eine zweifache Bestätigung schickt, wobei die Acknowledge Number auf den Wert gesetzt wird, der eigentlich erwartet wurde.

Aus Sicht des Senders kann eine doppelte Bestätigung verschiedene Gründe haben:

- verlorengegangene Segmente,
- Vertauschung der Reihenfolge,
- Duplizieren des Segmentes oder der Bestätigung durch das Netz selbst.

Der Sender wartet jetzt vier identische Bestätigungen ab und sendet dann sofort das fehlende Segment ohne auf den Ablauf der Zeitüberwachung zu warten. Dieser Vorgang heißt Fast Retransmit.

Da der Empfänger ja Bestätigungen schickt, nimmt man an, dass das Netz nicht überlastet ist. Also wird nicht mit Slow Start begonnen, sondern man geht in den Zustand Fast Recovery über. In diesem Zustand wird der Schwellwert reduziert, z. B. auf zwei Segmente und dann für jede duplizierte Bestätigung um drei Segmente erhöht.

Dieser Zustand beleibt solange erhalten, bis die erste nicht-duplizierte Bestätigung ankommt. Dann geht das System wieder in den Normalzustand über.

- **Problem:**
  - möglicherweise lange Wartezeit auf Übertragungswiederholung durch lange Timeouts
- **Lösung:**
  - Jedes Segment wird quittiert
  - Bei einem verlorenen Segment erzeugt das folgende Segment eine Quittung mit unverändertem Acknowledgment-Wert
    → **Duplicate ACK**
  - Übertragungswiederholung nach 3. Duplicate ACK

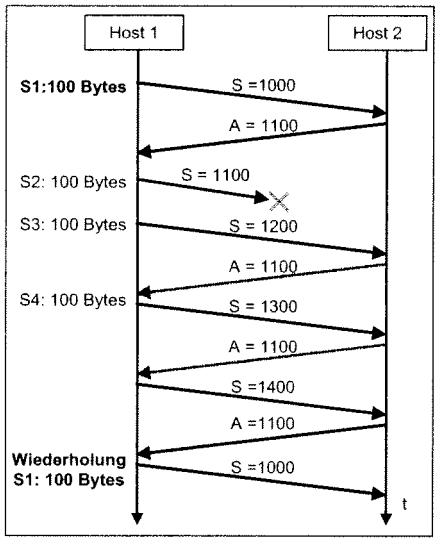
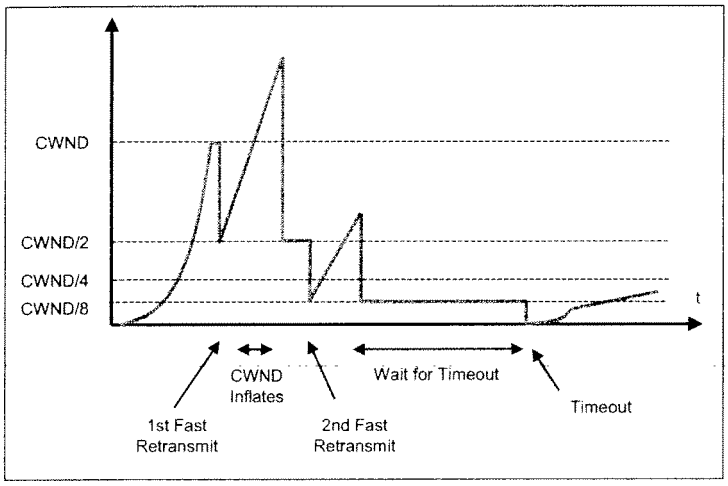


Bild: TCP Fast Retransmit



cwnd Congestion Window

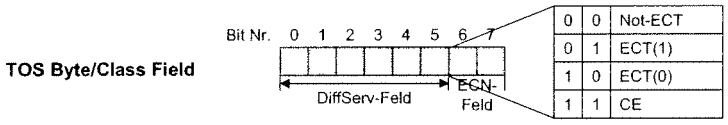
Bild: TCP Fast Retransmit und Recovery

Einige Punkte sind nicht ideal in einer TCP-basierten Umgebung:

Einer der Vorteile von TCP ist auch gleichzeitig ein Nachteil: So führt die Flusskontrolle zwar zu einem Schutz des Netzes, bewirkt aber, dass Dienste, die eine konstante Datenrate oder zumindest eine bestimmte minimale Datenrate benötigen, nicht auf TCP aufbauen können. Deshalb nutzen alle Echtzeit-Dienste im Internet heute UDP und bauen ihre Verbindungen in neuen Protokollschichten oberhalb von UDP auf.

Ein anderer Kritikpunkt ist, dass TCP die Bandbreite nicht fair unter allen Verbindungen aufteilt. So erhalten Verbindungen mit kurzen Segmentlaufzeiten automatisch auch eine höhere Bandbreite.

Der Slow Start und Congestion Avoidance Algorithmus führt bei lange anhaltenden Verbindungen zu einem Pumpen. Die Bandbreite wird solange immer erhöht, bis ein Fehler auftritt. Dann wird wieder mit einem kleinen Wert begonnen.

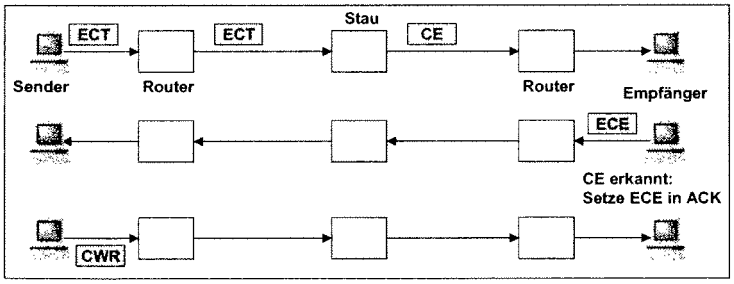


Problem: Netz ist „Black Box"
- Endsysteme schließen auf Stausituation nur indirekt über Paketverlust
- Paketverlust als Stauanzeige für verzögerungssensitive Anwendungen (z.B. Remote Login) ungünstig (Timeout + RTT)

Lösung: IP-Erweiterung ECN [RFC 3168]
- Vermeiden von Paketverlusten durch explizite Stauanzeige des Netzes (Signalisiert durch niederwertigste 2 Bits im ehemaligen IP-TOS-Feld)

Voraussetzung: Actives Queue Management im Router
- Anzeige muss erfolgen, bevor Warteschlange wirklich voll ist
- Markierung des IP-Pakets (Congestion Experienced – CE) anstatt es zu verwerfen
  - ECN-Fähigkeit muss signalisiert werden, um Unfairness zu vermeiden: ECN-Capable Transport (ECT) Bits

Bild: Explicit Congestion Notification

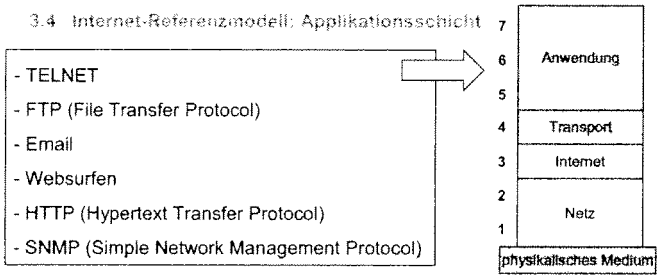


Erweiterung im TCP-Header: ECN-Echo-Flag (ECE) und Congestion-Window-Reduced-Flag (CWR)

Bei Verbindungsaufbau: ECN-Fähigkeit von TCP durch SYN+ECE+CWR signalisiert

ECE erkannt: Halbiere Staufenster, reduziere Slow-Start-Schwellenwert
Setze CWR in der nächsten Dateneinheit

Bild: Explicit Congestion Notification



- TELNET
- FTP (File Transfer Protocol)
- Email
- Websurfen
- HTTP (Hypertext Transfer Protocol)
- SNMP (Simple Network Management Protocol)

Der Begriff Internet-Dienste sind alle Dienste (Anwendungen) zusammen, die im Schichtenmodell oberhalb von TCP/ eingeordnet sind. Einige Dienste setzen auf TCP, andere auf UDP. Manche Dienste können wahlweise TCP oder UDP als Transportprotokoll nutzen.

RPC

Der RPC (Remote Procedure Call, RFC 1057) beinhaltet einen Operationsaufruf auf einem entfernten Knoten (Client-Server-Beziehung). Da die rechnerinterne Darstellung von Daten in heterogenen Netzen unterschiedlich sein kann, wird für die Codierung von Requests und Responses eine standardisierte Transfersyntax verwendet. Diese heißt XDR (External Data Representation, RFC 1014).

Verteilte Dateisysteme

In verteilten Systemen (Distributed Systems, Distributed Computing) ist die Verteilung von Dateisystemen auf mehrere Knoten von großer Bedeutung. Konzepte wie DFS (DCE File Service, DCE steht für Distributed Computing Environment) und NFS (Network File System) haben eine große Verbreitung gefunden. DFS wurde von Anfang an für heterogene Netze entwickelt. NFS (RFC 1094) kommt aus der UNIX-Welt und nutzt RPC für die Kommunikation über das Netz. NFS kann unterhalb von RPC auf TCP oder UDP aufsetzen. Für die Kopplung von DFS und NFS sind Gateways verfügbar. In der Welt der objektorientierten Programmierung werden Objekte auf verschiedene Knoten verteilt. Für die Nutzung von Objekten, die im Netz verteilt sind, existieren so genannte ORB (Object Request Broker). CORBA (Common Object Request Broker Architecture) ist das Konzept mit der größten Verbreitung.

Verzeichnisdienste

Ein Verzeichnis (Directory) enthält Angaben zu technischen Ressourcen oder zu Personen, die in Netzen verfügbar oder erreichbar sind. Der zugehörige Dienst wird allgemein als Verzeichnisdienst (Directory Service) bezeichnet. Namensdienste (Naming Service) bilden logische, leicht merkbare Namen einer Ressource oder Person auf eine (numerische) Netzadresse ab.

DNS

DNS (Domain Name System, RFC 1034, 1035) ist eine verteilte Datenbank, die die Abbildung von Endsystemnamen (Zeichenketten) zu IP Adressen bereitstellt (dabei wird in IP nur die Netz-ID berücksichtigt). DNS benutzt UDP oder TCP zur Diensterbringung. Endsysteme sind DNS-Clients, die Server werden als Domain Name Server bezeichnet. Jedes Endsystem muss einen DNS-Auflöser (resolver) lokal implementiert haben, der von den Anwendungsprogrammen aufgerufen werden kann.

DNS-Namen sind hierarchisch aufgebaut. Jede Domäne kann in Unterdomänen gegliedert sein. Die obersten Domänen der Hierarchie, TLD (Top Level Domains), umfassen festgelegte Institutionstypen und Länder. Beispiele für Institutionstypen (auch als Generic Domains bezeichnet) sind com (commercial), edu (education), org (Non-Profit-Organisationen) und weitere. Länder (Country Domains) werden durch ihre Codes nach ISO 3 166 bezeichnet. Beispiele: de (Deutschland), fr (Frankreich), uk (Großbritannien). Eine neue Domäne kann nur mit Zustimmung der nächsthöheren Domäne eingerichtet werden. Hingegen können in einer Domäne autonom Subdomänen eingerichtet werden. Die Top Level Domains werden von ICANN vergeben, während Domains über die nationalen NICs vergeben werden.

DDNS (Dynamic DNS, RFC 2136, RFC 2137) erweitert DNS, indem Name Server Aufträge zur dynamischen Änderung ihrer Datenbasis entgegennehmen können. Damit können Einträge ergänzt, gelöscht oder geändert werden. DDNS kann in einer ungesicherten oder in einer durch Authentikation gesicherten Variante genutzt werden.

X.500 Directory

X.500 ist das Konzept der ITU-T für Verzeichnisdienste. Die in Verzeichnissen nach X.500 verteilt abgelegte Information ist logisch in einem globalen Directory Information Tree organisiert. Für den Zugriff zu einem Namens- oder Verzeichnisdienst ist ein geeignetes Protokoll erforderlich, das hier als DAP (Directory Access Protocol, X.519) bezeichnet wird. Anfragen werden vom DUA (Directory User Agent) mittels DAP an den DSA (Directory System Agent) übermittelt.

LDAP

LDAP (Lightweight Directory Access Protocol, RFC 1959, RFC 2251) ist ein Zugriffsprotokoll für Directories nach X.500. Der Aufwand für den Zugriff mittels LDAP ist wesentlich geringer als für den Zugriff mittels X.500-DAP. Insbesondere benötigt LDAP keinen OSI-Protokol Stapel, es setzt direkt auf TCP und IP auf. Die aktuelle Version 3.0 enthält zusätzliche Funktionen aus X.509 (spezifiziert Formale für Zertifikate und Verfahren zu deren Überprüfung) zur Authentifizierung von Clients. LDAP wird ergänzt durch LIPS (Lightweight Internet Person Schema) zur Beschreibung von Personen durch Attribute und LDIF (Lightweight Directory Interchange Format) zum Austausch von Informationen zwischen LDAP-Servern.

TELNET

TELNET (Telecommunications Network Protocol, RFC 854, RFC 855861 und viele weitere) ist im Rahmen des Protokollstapels TCP/IP die Standardanwendung für das Einwählen in entfernte Systeme (Remote Login). TELNET setzt auf TCP/IP auf und nutzt auf der Server-Seite die Portnummern 23. Es umfasst die Codierungsregeln, um ein Terminal beim Benutzer mit einem Kommandointerpreter auf dem entfernten System zu verbinden. Das Endgerät auf der Client Seite wird in der Regel ein Arbeitsplatzrechner sein, der ein Standard Terminal nachbildet (Terminalemulation). Der Typ des Terminals kann beim Verbindungsaufbau ausgehandelt werden.

Zum Verbergen der Heterogenität zwischen Client- und Server-System verwendet TELNET das Konzept des NVT (Network Virtual Terminal). Dies ist eine standardisierte Schnittstelle zwischen den entfernten Systemen. Sowohl im Client- als auch im Server-System werden die intern verwendeten Zeichencodes und Zeichenfolgen in das NVT-Format umgesetzt.

Teilfunktionen von TELNET sind auch in anderen Anwendungen wie FTP und SMTP integriert oder werden von diesen benutzt. Während FTP nur den Dateitransfer leistet, kann (sofern die dem Client eingeräumten Rechte dies zulassen) mit Hilfe des TELNET-Client die volle Funktionalität des entfernten Server-Systems genutzt werden.

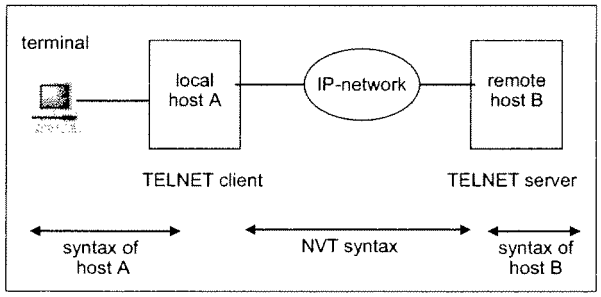


Bild: TELNET Client - Server

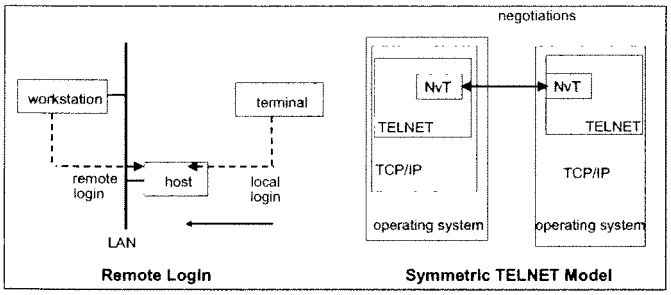


Bild: TELNET Remote Login

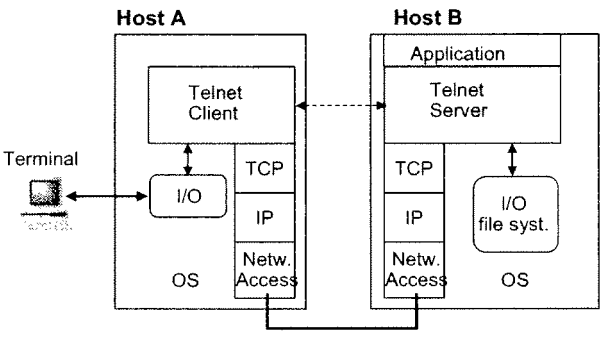


Bild: TELNET: Interne Prozesse

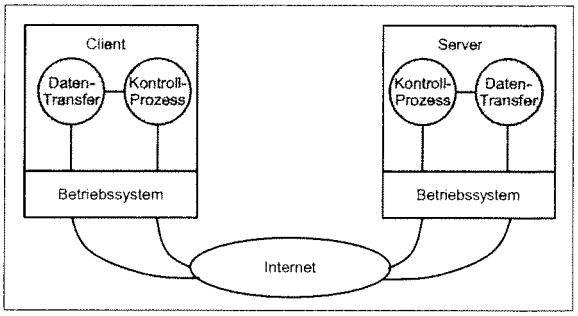
Dateitransfer (FTP)
FTP (File Transfer Protocol, RFC 959) leistet die Übertragung von Dateien zwischen Endsystemen. Es benutzt dazu Funktionen aus dem TELNET- und dem TCP-Protokoll. Durch die Berücksichtigung von Zugriffsrechten (Benutzername und Passwort), Abbildung von Dateinamen zwischen unterschiedlichen Systemen sowie der unterschiedlichen Dateiformate (Binär- oder Textdateien, Zeichencodierung) ist der zuverlässige Dateitransfer eine aufwändige Aufgabe.

FTP benötigt zwei Vollduplex-Verbindungen, von denen eine zur Steuerung und die andere zur eigentlichen Dateiübertragung genutzt wird. Auf der Serverseite werden die Portnummern 21 für die Steuerverbindung und 20 für die Dateiübertragung verwendet. Die Portnummern auf der Clientseite werden dynamisch zugewiesen. Viele FTP-Server lassen ein Login für gelegentliche Benutzer zu (Anonymes FTP). Dazu wird als Benutzernarne anonymous und als Passwort guest (bzw. die eigene E-Mail-Adresse) angegeben.

TFTP
TFTP (Trivial File Transfer Protocol, RFC 783) bietet eine zuverlässige Übertragung von Dateien auf der Basis von UDP. Es ist zur Inbetriebnahme (Bootstrapping) von Systemen innerhalb eines LAN gedacht. Da es einfach und kompakt ist, kann es im System-EPROM abgelegt werden.

- **File Transfer Protocol (FTP)**
  - Senden, Empfangen, Löschen, Umbenennen von Dateien
  - Übertragungsmodi: textuell und binär
  - separate Daten- und Kontrollverbindungen
  - Wechseln, Einrichten und Löschen von Directories
  - Kommandos: put, get, cd, ls, rm, ...
- **Trivial FTP (TFTP)**
  - Einfaches Protokoll auf UDP-Basis
  - Verbindung zu bestimmtem, voreingestellten Server-Directory
  - keine Authentifizierung
  - Anwendungsbeispiele
    - Booten von X-Terminals
    - Download von Konfigurationsdateien in Geräte

Bild: Dateitransfer



- permanente Kontrollverbindung
- Datenverbindung für jeden Datentransfer neu

Bild: File Transfer Protocol (FTP)

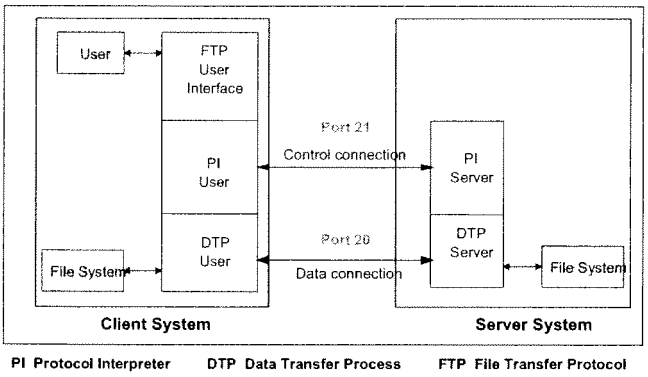


Bild: FTP Prinzip

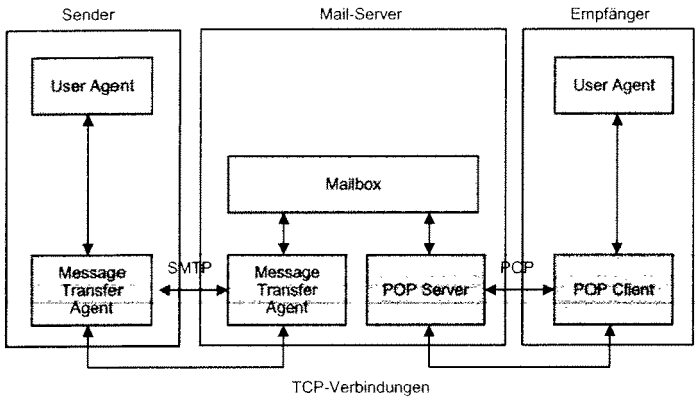


Bild: Elektronische Post

E-Mail

Die Möglichkeit, elektronische Post zu versenden, ist ein besonders attraktiver Dienst, der allerdings im Netz zusätzliche Vorkehrungen benötigt. Grund ist der, dass ein Benutzer seinen Rechner (z. B. PC) nicht ununterbrochen in Betrieb, besonders nicht ununterbrochen mit dem Netz verbunden hat. Daher müssen elektronische Nachrichten, sogenannte E-Mails, in einem Mail-Server zwischengespeichert werden, bis der Teilnehmer sich aktiv in den Mail-Server einloggt und seine Post abholt.

POP und IMAP

POP3 (Post Office Protocol, Version 3, RFC 1725) ist ein Protokoll für den Zugriff des Mail Client auf den Mail Server. POP3 setzt auf TCP auf. Seine wichtigsten Funktionen sind Login beim Mail Server, Authentifikation durch ein Passwort, Abfrage von Nachrichten auf der Mailbox des Servers sowie deren Löschung aus dem permanenten Speicher. Auf der Plattform des Mail Servers muss ein SMTP- und ein POP3-Server installiert sein. **IMAP4** (Internet Message Access Protocol, Version 4, RFC 1730) ist eine Erweiterung von POP3 und enthält zusätzliche Funktionen für den Umgang mit Mailboxes und mit Mail-Nachrichten.

- **Problem:**
  - Email-Server müssen ständig empfangsbereit sein, viele PCs sind aber nur temporär am Netz
- **Lösung:**
  - Emails werden mit einem Client-Protokoll von Server zum Client übertragen und auf Client-System abgelegt.
- **Post Office Protocol (POP)**
  - Funktionen:
    - Einloggen
    - Ausloggen
    - Nachrichten abholen
    - Nachrichten löschen
  - Flache, strukturlose Mailbox auf dem Server
- **Interactive Mail Access Protocol (IMAP)**
  - Emails können auf Server gespeichert bleiben
  - dadurch: Zugriff von mehreren unterschiedlichen Computern aus möglich

Bild: Client-Protokolle

- Direkte TCP-Verbindungen zwischen Mail-Servern
  - Senden mehrerer Nachrichten über eine TCP-Verbindung
  - TCP-Verbindung auch in der Rückrichtung nutzbar
- Nachrichten
  - enthalten nur ASCII-Text
  - maximale Nachrichtenlänge < 64 KB
  - Nachrichtenformate
    - RFC822 (Text)
    - Multipurpose Internet Mail Extensions (MIME)
- Austausch von Kommandos zwischen Client und Server
  - HELO: Vorstellung
  - MAIL: Angabe des Absenders
  - RCPT: Angabe des Empfängers
  - DATA: Senden der Nachrichten
  - QUIT: Ende
  - VRFY: Verifizieren des Benutzernamens
  - EXPN: Auskunft über Verteilerlisten
- Empfänger bestätigt jede Meldung

Bild: Simple Mail Transfer Protocol (SMTP)

SMTP

Das Simple Mail Transport Protocol SMTP ist im Gegensatz zu TELNET und FTP keine Benutzer-zu-Benutzer-Anwendung, sondern dient zum Austausch elektronischer Post - also E-Mail - auf Grundlage einer TCP-basierten, verbindungsorientierten Rechner-zu-Rechner Kommunikation.

Die SMTP-Instanz auf einem Rechner wird auch als Mail Transfer Agent MTA bezeichnet. SMTP stand früher in Konkurrenz zum OSI-Pendant X.400 E-Mail, das aber heute im Internet kaum mehr eingesetzt wird. Das X.400 E-Mail-Protokoll besitzt einen wesentlich größeren Funktionsumfang, der jedoch im Laufe der Zeit in die Internet-E-Mail-Applikationen integriert wurden.

Für SMTP ist es irrelevant, wie der Anwender an seine E-Mails kommt; es genügt, die zuzustellenden E-Mails lokal in einem sog. Message Store (MS) zu deponieren oder sie an den nächsten MTA weiterzuleiten. Das Abholen der E-Mails und ihre Aufbereitung ist Aufgabe eines User Agents (UA) als Applikation auf dem MTA-Host selbst, oder aber als Remote User Agent (RUA) auf einem abgesetzten Rechner. Die Gesamtheit - bestehend aus einem Verbund von MTA, MS, UA und RUA - wird auch als Message Transport System MTS verstanden.

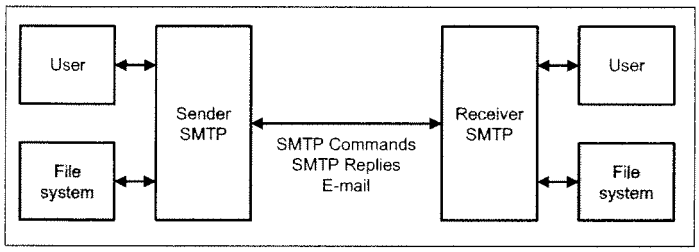


Bild: Simple Mail Transfer Protocol: Modell 1

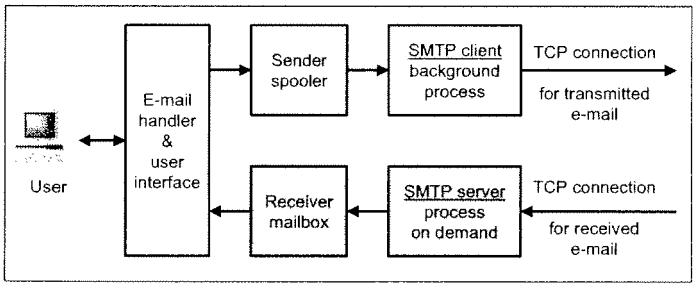


Bild: Simple Mail Transfer Protocol: Modell 2

- **Probleme des RFC 822 Standards**
  - Keine Übertragung von Binärdateien oder ausführbaren Programmen möglich
  - keine länderspezifische Zeichen
  - begrenzte Größe der Mails

⇒Multipurpose Internet Mail Extensions (MIME)
- zusätzliche Header-Zeilen (im Vergleich zu RFC 822), z.B. Content-Type, MIME-Version
- Definition von Content-Typen
- Methoden zur Codierung binärer Informationen durch ASCII-Zeichen

Bild: Multipurpose Internet Mail Extensions (MIME)

MIME

MIME (Multi-Purpose Internet Mail Extension, RFCs 1521, 1522, 2045-2049) ist ein Mechanismus zur Übertragung unterschiedlicher Datentypen in Mail-Nachrichten. Dies können Medien wie Video und Audio sein, ebenso sind nationale Zeichensätze zulässig, Die Einschränkung auf 7-Bit-ASCII-Zeichen wird also überwunden, obwohl die eigentliche Übertragung nach wie vor mit einer ASCII-Codierung nach RFC 822 erfolgt. MIME-Nachrichten werden durch für den jeweiligen Datentyp geeignete Plugin bzw. Viewer dargestellt.

- **Text**
  - Plain
  - Richtext
  - HTML
- **Multipart**
  - Mixed
    - verschiedene, unabhängige Teile
  - Parallel
    - Teile müssen gleichzeitig betrachtet werden.
  - Alternative
    - Die verschiedenen Teile enthalten dieselbe Information.
  - Digest
    - wie Mixed, mit Default-Subtyp message/rfc822

- **Message**
  - RFC822
  - Partial
    - Fragmentierung großer Mails
  - External-body
    - Pointer auf externes Objekt
- **Image**
  - GIF
  - JPEG
- **Video**
  - MPEG
- **Audio**
  - Basic (PCM μ-law)
- **Application**
  - PostScript
  - Byte-strea

Bild: MIME-Content-Typen

- **7bit**
  - ASCII Zeichen, kurze Zeilen
- **8bit**
  - ASCII- und Nicht-ASCII Zeichen, kurze Zeilen
- **Binary**
  - ASCII- und Nicht-ASCII Zeichen, lange Zeilen
- **quoted-printable**
  - ASCII-Zeichen + Hex-Codes
  - z.B. Darstellung von Form-Feed (dezimal: 12) : "=0C"
  - „weiche" Zeilenumbrüche ("3D0D0A") langer Zeilen
  - weiterhin lesbar
- **base64**
  - Abbildung von 6-Bit-Werten auf druckbare ASCII-Zeichen: 24 Bits → 4*6 Bits → 4 Bytes

Bild: MIME-Codierungen

WWW

Das WWW (World Wide Web) besteht aus Clients, Server und Objekten. Die Clients werden als WWW-Browser bezeichnet. Sie rufen von den WWW-Servern Objekte ab, um diese auszuwerten. Objekte sind elektronische Dokumente und Daten jeglicher Art, insbesondere jedoch Hypertext- und Hypermedia-Dokumente. Das WWW integriert weitere Internetdienste mit Hilfe der einheitlichen Benutzerschnittstelle des Browsers.

Hypertext- und Hypermedia-Dokumente enthalten mehrere Komponenten. Hypertext ist Text, der durch Links (Verweise) ergänzt wird. Ein Link ist ein Verweis auf eine andere Textstelle oder ein anderes Dokument (Objekt). Links können insbesondere verweisen auf:

- andere (Text-)Stellen in demselben Dokument (Objekt),
- Objekte (Dateien) innerhalb eines Dateisystems oder Computers,
- Objekte (Dateien, Dokumente) in einem Netz.

Hypermedia enthält zu Text und Links zusätzlich multimediale Anteile wie Grafik, Bilder (Bewegt- und Festbilder) sowie Sprache (bzw. allgemein Töne).

Es lassen sich drei Arten von Web-Dokumenten unterscheiden:

- **Statische Dokumente:** Diese werden vom Autor des Dokuments bei dessen Erstellung vollständig beschrieben und in einer Datei auf dem Server abgelegt.
- **Dynamische Dokumente:** Sie werden vom Server erstellt, nachdem sie vom Client angefordert wurden. Dazu benutzt der Server ein Anwendungsprogramm. Die Konsequenz ist, dass ein Dokument je nach Anfrage unterschiedliche Inhalte haben kann.
- **Aktive Dokumente:** Sie werden auf dem Server nicht vollständig beschrieben. Das Dokument enthält ein Programm, das Werte berechnen und innerhalb des Dokuments ausgeben kann. Das Programm wird dem Client zusammen mit dem Dokument übermittelt. Bei der Ausführung auf dem Browser (Client) kann das Programm mit dem Benutzer interagieren und dadurch den Inhalt des Dokuments verändern, Aktive Dokumente werden häufig mittels JavaScript bzw. Java-Applets erstellt.

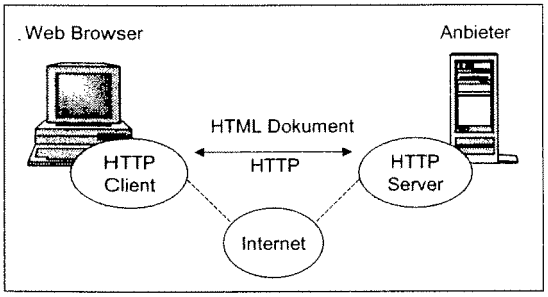
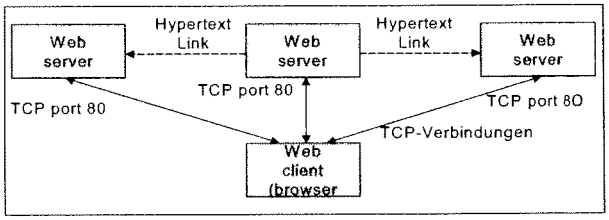


Bild: Hyper-Text Transfer Protocol (HTTP)



Client-Server-Applikation

Hypertext: über ein Netz verbundene Objekte
-Die grundlegende Idee Hypertext ist wesentlich älter als WWW
-Vorgänger: gopher (kein Hypertext)

Grafischer Browser
-Nutzung auch durch Nicht-Informatiker

Bild: World Wide Web

Informationsinhalte (Cotent) des WWWs lassen sich auf verschiedene Weise beschreiben bzw. erzeugen.

URL, URI, URN

Eine URL (Universal Resource Locator, RFC 1738) kennzeichnet den Lagerort eines Web-Dokuments durch Angabe des Servers und des Pfades im Dateisystem des Servers. Der Schema-Teil gibt die Bezeichnung des Dienstes an, z. B. f tp, http. gopher, news oder mailto. Im schemaspezifischen Teil können User, Password und URL-Pfad entfallen und die Port-Nummer kann per Default (implizit) festgelegt sein. Der Nachteil einer URL liegt darin, dass sie sich bei Veränderungen im Dateisystem eines Servers ändern kann, wodurch das Dokument nicht mehr gefunden wird.

URN (Universal Resource Name) geben einen global eindeutigen, langlebigen logischen Namen für ein Dokument an, der keine Aussage über den Lagerort macht. Der Lagerort wird in einem Directory gespeichert und dort erfragt. Mittels URN bleibt ein Dokument also langfristig unter derselben Bezeichnung verfügbar. Zudem kann es repliziert gespeichert werden, wobei der Zugriff auf einen günstigen Server erfolgt. URI (Universal Resource Identifier, RFC 1630) ist der Oberbegriff für URL und URN. Zukünftig kann URI jedoch um weitere Benennungs-Schemata ergänzt werden.

HTTP

Das Hypertext Transport Protokoll HTTP ist heute neben SMTP die zentrale Anwendung im Internet.

Der HTTP-Standard sieht das Client-/Server-Prinzip mit folgenden Funktionselementen vor:

- den HTTP-Client bzw. User-Agent, der Browser (bzw. Web-Browser) genannt wird;
- den HTTP-Server, der die HTML-Dokumente sowie die ablauffähigen Skripte beherbergt, sowie
- den HTTP-Cache-Server bzw. auch HITP-Proxy genannt, zum zeitweisen Ablegen (Cachen) der HTML-Dateien sowie einiger HTTP-Informationen.

HTTP ist durch die folgenden Eigenschaften gekennzeichnet.

- HTTP ist ein Protokoll der Anwendungsschicht, es setzt auf TCP auf.
- HTTP ist zustandslos, d. h. der Server betrachtet jede Anfrage unabhängig von vorhergehenden Anfragen.
- Eine bidirektionale Übertragung ist möglich.
- Browser und Server können bestimmte Merkmale für die folgenden Datentransfers aushandeln.
- Caches im Browser und in Proxy Servern werden unterstützt.

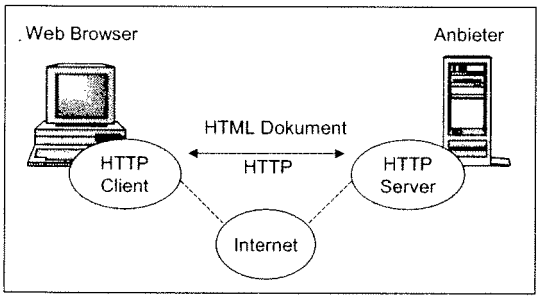
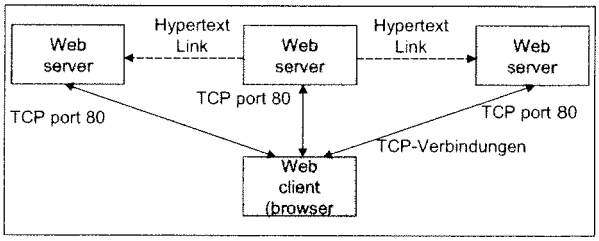


Bild: Hyper-Text Transfer Protocol (HTTP)

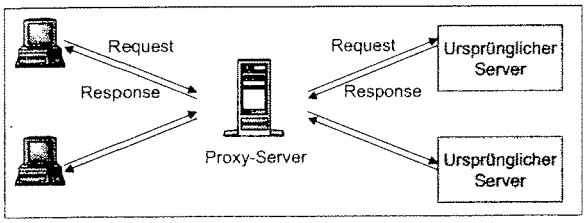


Client-Server-Applikation

Hypertext: über ein Netz verbundene Objekte
-Die grundlegende Idee Hypertext ist wesentlich älter als WWW
-Vorgänger: gopher (kein Hypertext)

Grafischer Browser
-Nutzung auch durch Nicht-Informatiker

Bild: World Wide Web



Web-Cache (auch Proxy-Server)
-Speichert Kopien angeforderter Objekte
-Fungiert sowohl als Server als auch als Client

Vorteile
- Geringere Antwortzeiten
- Reduktion des Verkehrs auf Zugangslinks

Bild: Web-Caches

- HyperText Markup Language (HTML) zur Beschreibung von WWW-Seiten
  - Tags zur Markierung von Formatierungsanweisungen
  - Beispiele
    - <B> Hallo </B>: Hallo
    - <I> Hallo </I>: Hallo
- HyperText Transfer Protocol (HTTP) zum Austausch von HTML-Seiten
  - Request / Response zwischen Client und Server über eine TCP-Verbindung
  - ausgewählte Methoden:
    - HEAD: Meta-Informationen der Web-Seite lesen
    - GET: Web-Seite lesen
    - PUT: Web-Seite speichern
    - POST: Daten an Web-Seite anhängen
    - MOVE, COPY, DELETE
    - LINK, UNLINK
  - HTTP/1.1
    - mehrere Abfragen über 1 Verbindung

Bild: World Wide Web

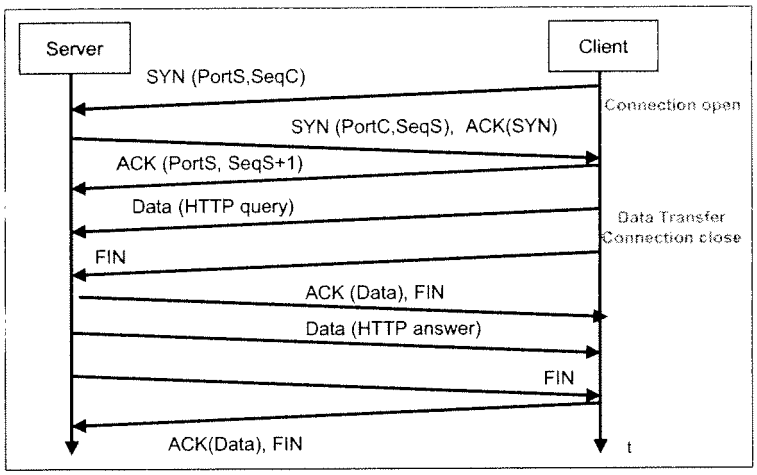


Bild: HTTP Query

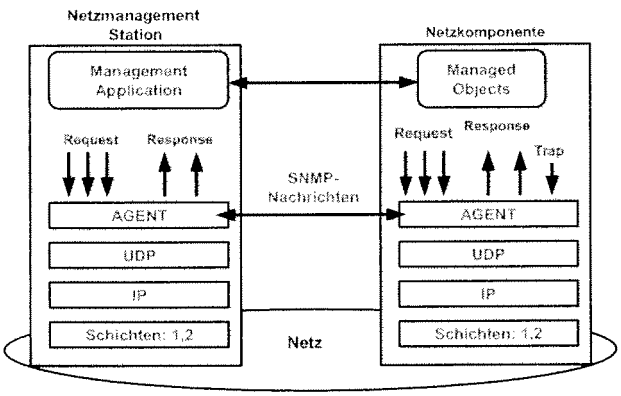


Bild: Kommunikationsmodell von SNMP

SNMP

Das SNMP (Simple Network Management Protocol) wurde im Jahr 1990 als Standard für IP-Netze erklärt.

Es existieren folgende Schlüsselelemente:

- Managementstation,
- Agent,
- Management Informationsbasis MIB,
- SNMP Protokoll.

Üblicherweise gibt es nur eine Managementstation, aber sie kann auch als Erweiterung in ein geteiltes System implementiert werden. Sie ist der Zugangspunkt für den Verwalter des Netzes zu dem Netzmanagementsystem. Das geringste, was in einen solchen System vorhanden sein müsste, ist:

- ein Satz von Managementanwendungen zur Analyse von Daten, Fehlerbehebung, usw.
- eine Schnittstelle für den Verwalter des Netzes, zu Überwachung und Kontrolle
- Übersetzen der Anforderungen des Verwalters in Befehle für Netzelemente und die Kontrolle der entfernten Netzelemente im Netz
- eine Datenbasis von Informationen aus MIBs von allen zu verwaltenden Elementen im Netz

Dabei sind nur die letzten zwei Elemente von Standardisierungsprozessen in SNMP betroffen.

Das zweite aktive Element im Netzmanagement ist der Agent. Schlüsselplattformen wie Host, Bridge, Router und Hubs müssen mit einen SNMP Agenten ausgerüstet sein, um so von einer Managementstation abfrage- und steuerbar zu sein. Auch wichtige Informationen (Alarme) sind auf diese Weise, asynchron und unaufgefordert vom Manager, vom Netzelement zu erhalten.

Es ist möglich, alle Elemente im Netz zu verwalten, indem man sie als Objekte darstellt. Jedes Objekt ist im wesentlichen eine Datenvariable, die einen Aspekt des Agenten darstellt. Eine Sammlung von Objekten ist als Management Informationsbasis (MIB) zu betrachten. Für die Managerstation funktioniert die MIB als Sammlung von Zugangspunkten auf den Agenten. Diese Objekte sind im Netz für eine Klasse von Elementen standardisiert (z.B. ein gemeinsamer Satz von Objekten wird benützt für das Managen von verschiedenen Routern). Eine Managementstation führt die Überwachungsfunktion durch Lesen der Werte von MIB Objekten durch. Sie kann eine Aktion hervorrufen oder die Konfiguration ändern, mit der Änderung des Wertes einer bestimmten Variable.

Die Managementstation und der Agent kommunizieren über ein Netzmanagementprotokoll. Das Protokoll, das für das Management von TCP/IP Netzen genützt wird, ist das Simple Network Management Protocol (SNMP). Es hat folgende Schlüsseleigenschaften:

- **get**: ermöglicht dem Manager, Werte der Objekte auf Agenten abzufragen
- **set**: ermöglicht dem Manager, Werte der Objekte auf den Agenten zu setzen
- **trap**: ermöglicht dem Agenten, den Manager über wichtige Ereignisse zu benachrichtigen

Die Standards schreiben nicht vor, wie viele Managementstationen es geben kann und wie viele Agenten eine Managementstation an sich binden kann. Es ist empfehlenswert, mindestens zwei Systeme mit Managementfähigkeiten zu haben und damit über eine Redundanz zu verfügen. Die Anzahl von Agenten kann in die Hunderte gehen.

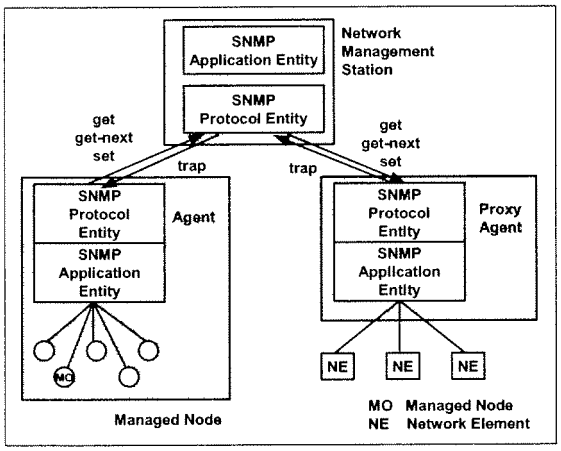


Bild: SNMP Konzept

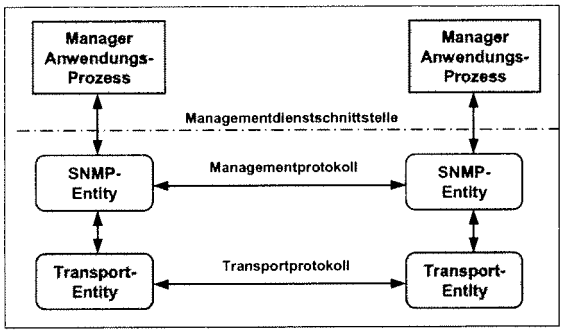
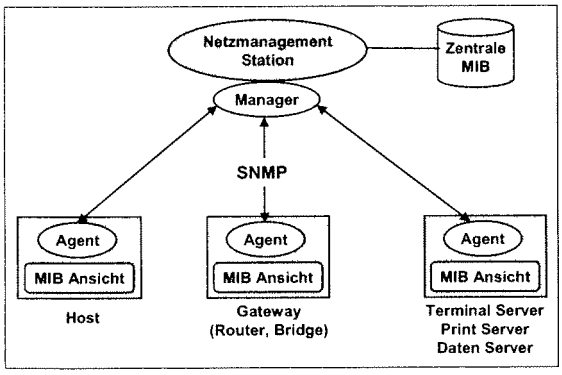


Bild: Managementdienste und -Protokolle



MIB: Management Information Base

Bild: Netzmanagement Elemente im SNMP

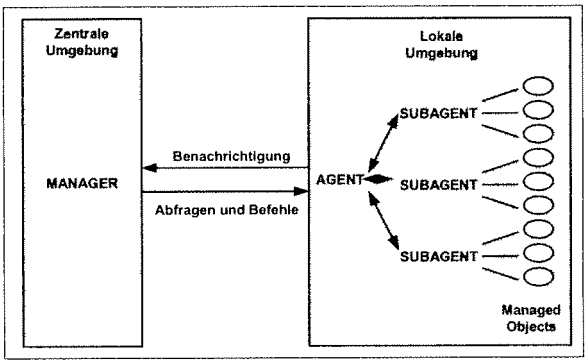


Bild: Managementmodell

Die Management Information Base ist die aktuelle Struktur oder das Schema der Datenbasis, die die Netzelemente beschreibt. Jedes Netzelement ist als ein Objekt in der MIB dargestellt. Für SNMP ist die

MIB eine hier-archische Datenbasisstruktur. Die Struktur kann man wie einen Baum darstellen, wobei jedes Blatt ein zu verwaltendes Element im Netz darstellt und jeder Ast ein System mit den dazugehörigen Elementen. Jedes System (Arbeitsstation, Server, Router, Bridge, usw.) verwaltet eine MIB, die den Stand der zu verwaltenden Elemente im System widerspiegelt.

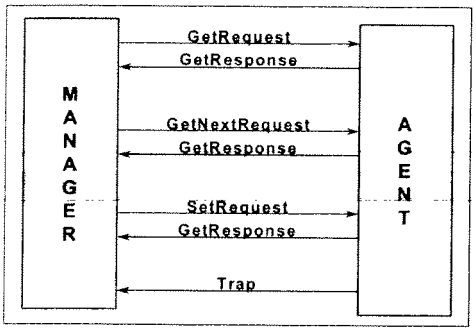


Bild: SNMPv1 Grundoperationen

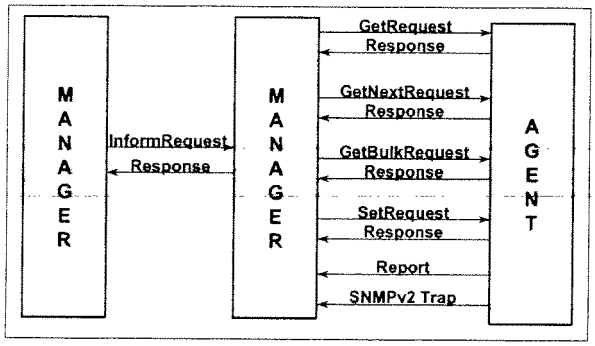


Bild: SNMPv2 Grundoperationen

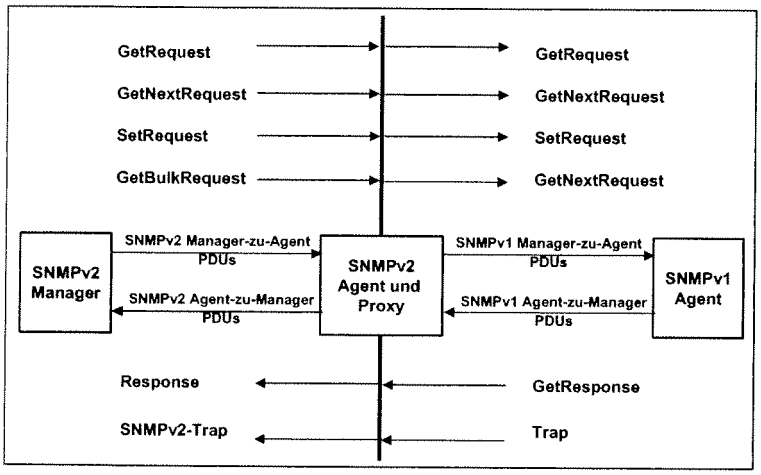


Bild: Koexistenz von v1 und v2 mit Proxy Agenten