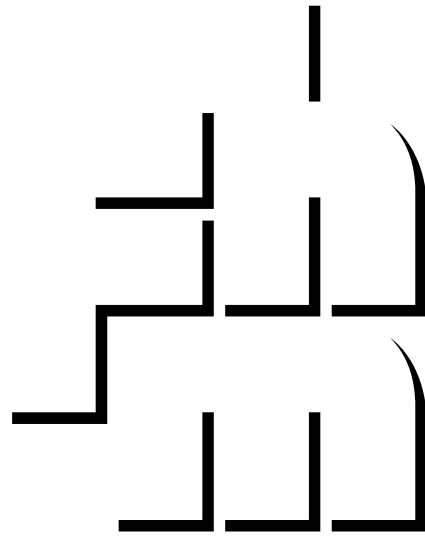




fachhochschule

munich university of applied sciences

münchen

COMPUTERNETZE

Prof. Dr. Ing Manfred Paul

Sommersemester 2004

Stand: 13.3.04

Vorwort

Die Vorlesung Computernetze wird an der Fachhochschule München im Fachbereich Elektro- und Informationstechnik seit 1993 gehalten, derzeit im Bachelor und Diplomstudiengang für Studenten des 7.Semesters. Ziel der Vorlesung ist die Vermittlung eines grundlegenden Verständnisses für die Mechanismen in Computernetzen. Der Inhalt dieses Skripts wird ständig überarbeitet und angepasst, um aktuellen Entwicklungen, soweit möglich, Rechnung zu tragen. Das vorliegende Skript der Vorlesung entspricht dem Stand des Sommeresters 2004.

Die Reihenfolge der Kapitel und die Aufteilung von ähnlicher Themen auf verschiedene Kapitel ergibt sich aus dem Vorlesungs- und Praktikumsbetrieb. So wird zu dieser Vorlesung ein Praktikum angeboten, für das weitgehend sichergestellt sein muß, dass die Studenten den Inhalt der Versuche in der Vorlesung bereits gehört haben sollten. Zudem musste der Stoff mit meiner zweiten Vorlesung Netzwerksicherheit zeitlich abgestimmt werden.

Zunächst werden Zugriffsmodelle auf Dienste in Netzen erklärt, die heutigen Dienste an sich und ihre Anforderungen an die Netzwerkinfrastruktur. Aufsetzend auf den generellen Aufgaben von Protokollen in Netzen anhand des OSI und DoD Modells und sehr grundsätzlichen Aspekten bei der Paketübertragung zwischen Computern im selben Netzwerksegment werden wesentliche Aspekte von IP und TCP/UDP beleuchtet, um Grundlagen für das Kapitel Firewalls und Netzwerksicherheit zu schaffen. Dies ist gleichzeitig der Einstiegspunkt in die gleichzeitig stattfindende Vorlesung Netzwerksicherheit.

Das Thema, wie genau die Paketübertragung innerhalb von Netzwerksegmenten aussieht, wird bewusst zurückgestellt, für das grundsätzliche Verständnis ist es zunächst zweitrangig, ob Information über Ethernet Frames oder in Paketen auf Multis über eine grüne Wiese transportiert wird.

Die folgenden Kapitel befassen sich mit den Themen Internetworking, Netzwerkadministration, Netzwerkmanagement, sowie besondere Administrations/Managementaufgaben in IP Netzen.

Der nächste Teil beschäftigt sich mit den grundlegenden Aspekten der Verbindungskontrolle in Netzen, im wesentlichen am Beispiel TCP und beinhaltet einen Ausflug zur Echtzeitübertragung, z.B. der Telephonie in Netzen, die gegenüber der reinen Datenkommunikation ganz andere Anforderungen an Netze und zu verwendende Protokolle stellt, sowie zur nächsten Version des IP Protokolls.

Schließlich wird der bis dahin zurückgestellte Teil der Kommunikation in Direktverbindungsnetzen mit den grundsätzlichen Medienzugriffsverfahren und -protokollen im leitungsgebundenen und im Wireless Bereich ergänzt.

Praktikumsversuche zu dieser Vorlesung umfassen folgende Themen:

- Administration eines Directory Services
- TCP/IP Routing und DNS/DHCP Administration
- Konfiguration einer Firewall
- Netzwerkmanagement
- Desktopmanagement
- Konfiguration eines 2-Rechner Clusters

Inhalt

1.	GRUNDLAGEN	1-8
1.1	Netzwerkmodelle	1-8
1.1.1	Zentralisierte Datenverarbeitung (Centralized Computing)	1-8
1.1.2	Verteilte Datenverarbeitung (Distributed Computing)	1-8
1.1.3	Internet Protokollentwicklung	1-9
1.2	Anforderungen an Rechnernetze	1-9
1.3	Nutzung und Verteilung von Diensten.....	1-10
1.3.1	Peer to Peer Konzept.....	1-11
1.3.2	Client - Server - Konzept	1-11
1.3.3	Client - Network	1-12
1.4	Architektur des Zugriffs.....	1-12
1.5	Multi-Tier Networking (2 ... n-tier Networking).....	1-12
1.6	Einordnung von Netzen aufgrund ihrer Größe.....	1-13
2.	NETZWERKDIENTSTE (BZW. DIENSTEPROTOKOLLE)	2-1
2.1	Terminalemulation (Telnet)	2-1
2.2	Entfernter Dateizugriff via FTP.....	2-1
2.3	File Service.....	2-1
2.3.1	NFS	2-2
2.3.2	NCP (Novell Core Protokoll).....	2-2
2.3.3	SMB (Microsoft Windows System Message Block)	2-2
2.4	Web Service (http).....	2-2
2.5	Multimedia Service.....	2-2
2.6	Application Service.....	2-3
2.6.1	Datenbankserver	2-3
2.6.2	Spezielle Datenbankservices.....	2-3
2.7	Message Services.....	2-3
2.7.1	SMTP	2-4
2.7.2	POP3	2-4
2.7.3	IMAP.....	2-4
2.7.4	Usenet News (NNTP).....	2-4
2.7.5	Kommerzielle Messaging Server	2-4
2.8	Print Services:.....	2-4
2.9	Zeitservice.....	2-5
2.10	Weitere Services	2-5
2.11	Namen und Adressen im Netzwerk	2-5
2.11.1	Dienstadressen	2-5
2.11.2	Namensauflösung.....	2-5
2.11.3	Adressvergabe.....	2-6
2.11.4	Suche nach Ressourcen	2-6
3.	PROTOKOLLE UND STANDARDS.....	3-1
3.1	Kommunikationsmodell.....	3-1
3.2	OSI Referenzmodell.....	3-2
3.3	DoD Modell	3-4
3.4	Protokolle und Standards	3-4
4.	DIREKTVERBINDUNGSNETZWERKE	4-1
4.1	Protokollaufgaben.....	4-1
4.2	Kommunikation im Direktverbindungsnetz	4-1
4.2.1	Netzwerkkarte.....	4-1

4.2.2	Ethernet Protokoll	4-1
4.2.3	Andere Zugriffsprotokolle	4-2
4.3	Frame Switching	4-2
4.4	LLC Teilschicht	4-2
5.	NETZWERKSCHICHT	5-1
5.1	Netzwerkweite Adressierung	5-1
5.2	Das Internet Protokoll (IP)	5-2
5.3	IP-Adressierung	5-3
5.4	Paketzustellung im IP-Netz	5-4
5.4.1	Das ARP Protokoll	5-4
5.4.2	Routing am Beispiel IP	5-4
5.5	Routingprotokolle	5-7
5.6	ICMP	5-8
6.	TRANSPORTSCHICHT	6-1
6.1	Übertragungsarten	6-1
6.2	Transportprotokolle im Internet	6-2
6.2.1	TCP (Transport Control Protocol)	6-2
6.2.2	UDP (User Datagram Protocol)	6-3
6.2.3	Sockets	6-3
6.3	TCP Zustandsmodell	6-3
6.3.1	Verbindungsaufbau	6-4
6.3.2	Halboffene Verbindungen	6-5
6.3.3	Verbindungsabbau	6-7
6.4	Gateways zu anderen Protokollstacks	6-8
7.	FIREWALLS UND NETZWERKSICHERHEIT	7-1
7.1	Angriffsmöglichkeiten	7-1
7.2	Funktion und Komponenten von Firewalls	7-2
7.2.1	Paketfilter	7-2
7.2.2	IP Adressumsetzung (NAT)	7-2
7.2.3	TCP/UDP Relay	7-3
7.2.4	Application Layer Gateway (Proxy)	7-3
7.3	Firewallarchitekturen	7-5
7.4	Sicherheit von Netzen	7-7
7.5	Verschlüsselung	7-7
7.5.1	Symmetrische Verschlüsselung	7-7
7.5.2	Asymmetrische Verschlüsselung	7-8
7.5.3	Digitale Signaturen	7-9
7.6	Beglaubigung (Authentication)	7-10
7.7	Schlüsseltausch	7-11
7.7.1	Diffie Hellman Methode	7-12
7.7.2	Schlüsseltausch über Dritte	7-12
7.7.3	Kerberos Protokoll	7-12
7.8	Zertifizierung	7-14
7.9	Anwendungsbeispiele	7-15
7.9.1	Secure Socket layer (SSL)	7-15
7.9.2	IP Security (IPSEC)	7-15
7.9.3	PGP (Pretty Good Privacy)	7-16
7.10	Sicherheitspolitiken	7-17

8.	VERWALTUNG VON COMPUTERNETZEN	8-1
8.1	Serverbezogene Administration	8-1
8.2	Domänenkonzept	8-1
8.3	Directory Service	8-1
8.3.1	X.500 Directory Standard	8-2
8.3.2	Navigation durch eine Directory.....	8-3
8.3.3	Aufgaben einer Directory	8-3
8.3.4	Realisierung in den verschiedenen Betriebssystemen:.....	8-4
8.3.5	Universeller Zugriff auf Directories.....	8-4
8.3.6	Metadirectories	8-5
8.4	Zugriffskontrolle in Netzen	8-5
8.4.1	Kennwortschutz	8-5
8.4.2	Zugriffsrechte	8-5
9.	NETZWERKMANAGEMENT	9-1
9.1	Netzwerkmanagement:.....	9-1
9.2	Desktopmanagement.....	9-3
9.3	Fehlertoleranz bei LAN Betriebssystemen	9-3
9.3.1	Kosten eines Systemausfalls.....	9-4
9.3.2	Realisierung von Ausfallsicherungssystemen.....	9-4
10.	BESONDERE ASPEKTE VON IP INFRASTRUKTUREN.....	10-1
10.1	Routing.....	10-1
10.1.1	Distanzvektorprotokolle	10-1
10.1.2	Verbindungsstatusprotokolle	10-2
10.2	Planung der Netzwerkinfrastruktur	10-2
10.2.1	Subnetting.....	10-2
10.2.2	Proxy ARP	10-4
10.2.3	Supernetting	10-5
10.3	Dynamische IP-Adressvergabe über DHCP.....	10-5
10.3.1	Verfahren der Adressvergabe	10-5
10.3.2	Ablauf der Lease Time	10-6
10.3.3	DHCP Frame Format	10-7
10.4	Multicast in IPNetzen	10-8
10.4.1	Multicast Adressierung	10-9
10.4.2	Aufbau der Infrastruktur	10-9
11.	AUFGABEN BEI DER PAKETÜBERTRAGUNG.....	11-1
11.1	Fehlererkennung.....	11-1
11.2	Zuverlässige Übertragung	11-1
11.2.1	Stop-And-Wait Algorithmus	11-2
11.2.2	Sliding Windows Algorithmus	11-2
11.2.3	Übertragung über mehrere logische Kanäle.....	11-2
11.3	Flußkontrolle bei Ende zu Ende Übertragung.....	11-3
11.3.1	Berücksichtigung der Round Trip Time (RTT).....	11-3
12.	ECHTZEITÜBERTRAGUNG IN NETZEN	12-1
12.1	Übertragung von Echtzeitdaten	12-1
12.1.1	RTP (Real-Time Transport Protocol).....	12-1
12.1.2	RTCP (RTP Control Protocol).....	12-1
12.2	Sitzungskontrolle und Signalisierung.....	12-1
12.2.1	H.323.....	12-2
12.2.2	IETF Protokolle	12-2

13. IPNG (IP NEXT GENERATION, IPV6)	13-1
13.1 IPv6 Paketformat.....	13-1
13.2 Adressierung.....	13-1
13.2.1 Aggregatable Global Unicast Addresses	13-3
13.2.2 Spezielle Unicast Adressen	13-3
13.3 Autokonfiguration.....	13-4
13.4 ICMPv6.....	13-4
 14. ÜBERTRAGUNGSMEDIEN FÜR NETZE	14-1
14.1 Leitungsgebundene Medien.....	14-1
14.2 Nicht Leitungsgebundene Medien.....	14-1
 15. GRUNDLAGEN UND STANDARDS FÜR DIREKTVERBINDUNGSNETZE ..	15-1
15.1 Leitungscodierung:	15-1
15.2 Frameerzeugung.....	15-2
15.2.1 Asynchrone Übertragung	15-2
15.2.2 Synchrone Übertragung.....	15-2
15.3 Physikalische Netztopologien	15-2
15.4 Physikalischer Aufbau von Netzen.....	15-4
15.5 Multiplexing.....	15-4
15.6 Medienzugriffsverfahren.....	15-4
15.6.1 Logische Topologien	15-4
15.6.2 Polling (Logische Sterntopologie)	15-4
15.6.3 Konkurrenzsystem (Contention, logische Bustopologie)	15-5
15.6.4 Token Passing (logische Ringtopologie).....	15-6
15.6.5 Vergleich Token Passing / Konkurrenzsystem.....	15-6
15.7 IEEE Projektgruppen.....	15-7
15.8 IEEE 802.3 und Ethernet	15-7
15.8.1 Entwicklung des Standards	15-7
15.8.2 CSMA/CD Zugriffsverfahren	15-9
15.8.3 Full Duplex Ethernet.....	15-10
15.8.4 Gigabit Ethernet (IEEE 802.3z, 802.3ab).....	15-10
15.8.5 10 Gigabit Ethernet (802.3ae)	15-11
15.9 IEEE 802.5 und Token Ring.....	15-12
15.10 FDDI (Fiber Distributed Data Interface).....	15-13
 16. INTERNETWORKING	16-1
16.1 Brücken.....	16-1
16.1.1 Transparente bzw. selbstlernende Brücke.....	16-1
16.1.2 Source Routing Bridge	16-1
16.1.3 SRT Bridges	16-2
IEEE 802.1d Spanning Tree Protocol.....	16-3
16.1.4 Grundlagen	16-3
16.1.5 Autokonfiguration.....	16-4
16.1.6 Topologieänderung	16-5
16.2 Frame Switching.....	16-5
16.2.1 Store-And-Forward Switching.....	16-6
16.2.2 Cut-Through Switching	16-6
16.2.3 Intelligent Switching.....	16-6
16.2.4 Token Ring Switching	16-6
16.2.5 Quality of Service.....	16-6

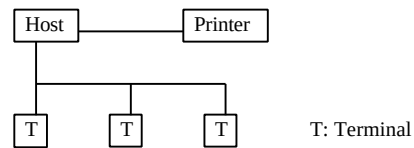
16.3	Virtuelle LANs.....	16-6
16.3.1	Funktion von VLANs.....	16-6
16.3.2	Statische und Dynamische VLANs.....	16-7
16.3.3	VLAN Tagging.....	16-7
16.4	Einsatzgebiete von Bridges, Switches und Routern.....	16-9
17.	WIRELESS LAN.....	17-1
17.1	IEEE 802.11 Wireless LAN.....	17-1
17.1.1	Spektrum Übertragung.....	17-1
17.1.2	Kopplung von Knotengruppen	17-1
17.1.3	Zugriffsverfahren	17-2
17.1.4	Frame Format	17-3
17.2	IEEE802.11 Nachfolgestandards.....	17-4
17.3	Weitere Wireless Übertragungsstandards	17-4
18.	ANHANG	18-1
18.1	Verwendete Abkürzungen.....	18-1
18.2	BOOTP AND DHCP PARAMETERS.....	18-4
18.3	Fragen zum Inhalt	18-9
18.4	Literatur	18-12

1. Grundlagen

1.1 Netzwerkmodelle

1.1.1 Zentralisierte Datenverarbeitung (Centralized Computing)

Großrechner / MDT (Mittlere Datentechnik)



Rechnerleistung zentral auf Host

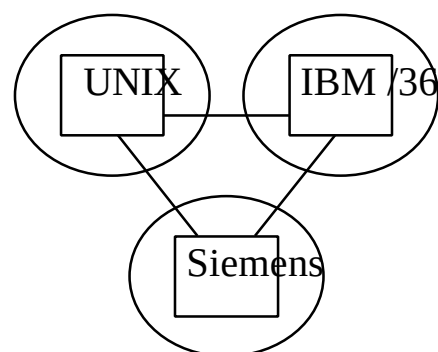
RJE (Remote Job Entry)

Starten von Funktionen auf dem Host

Interaktives Arbeiten mittels Terminal (Tastatureingaben, Bildschirmsteuerung erfolgt vom Großrechner)

Kommunikationsstandard: SNA = Systems Network Architecture (IBM)

1.1.2 Verteilte Datenverarbeitung (Distributed Computing)



Kennzeichen: Rechnerleistung ist verteilt, Gesamte Rechenleistung des Netzes wächst mit seiner Größe.
Mitte der 80er Jahre: Entwicklung von Rechnernetzen auf PC Basis.

1.1.3 Internet Protokollentwicklung

Die Entwicklung begann ca. 1970 mit einer Studie der DARPA (Department of Defense Advanced Research Projects Agency) zur Entwicklung von Protokollen zur Datenkommunikation. Der Schwerpunkt lag dabei auf obigen Anforderungen, erst sehr viel später kamen Protokolle zum transparenten Zugriff auf Filesysteme, also eine Umgebung die vergleichbar mit dem Zugriff auf Fileserver in Netzen ist, hinzu.

Die folgende Aufstellung zeigt die Entwicklung:

1972	Beginn der Internet Protokollentwicklung
1978	Internet Protokollstack weitgehend vollständig
1980	Verwendung im ARPANet (heute noch Teil des Internet)
1982	Verwendung der Internet Protokolle im BSD-Unix
1986	SUN entwickelt NFS (Network Filing System)*

Die Internet Protokollentwicklung und Standardisierung wird von mehreren Organisationen verantwortet:

- Internet Society (ISOC): Besitzt die Zielsetzung, das Internet als globale Kommunikationsplattform zu etablieren.
- Internet Architecture Board (IAB): Technische Organisation der Internetentwicklung. Das IAB ist eine Unterorganisation der ISOC.
- Internet Engineering Task Force (IETF): Standardisierungsgremium, aufgeteilt in neun Bereiche. Die IETF entwickelt Spezifikationen, die später zu Internet Standards werden. Unterorganisation der IAB.
- Internet Research Task Force (IRTF): Langfristige Forschungsprojekte, Unterorganisation der IAB.

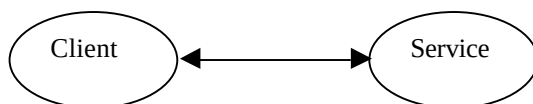
Die meisten der Protokolle und Anwendungen sind in sogenannten RFCs (Request for Comment) veröffentlicht. Diese sind im Internet verfügbar, die Pflege der RFCs wird vom SRI Network Information Center (NIC) in Menlo Park, Kalifornien, übernommen. RFCs werden bei Weiterentwicklungen von Protokollen nicht geändert, sondern es werden neue RFCs erstellt, die den geänderten Stand widerspiegeln. Insofern findet man sehr häufig mehrere RFCs zu einem bestimmten Thema. Die RFCs sind aufsteigend nummeriert, so daß höhere Nummern neuere Veröffentlichungen widerspiegeln.

1.2 Anforderungen an Rechnernetze

Die grundsätzliche Aufgabe von Rechnernetzen besteht darin, Dienste, die bestimmte Aufgaben erfüllen, zu nutzen. Beispiele sind

- Abfrage von Information
- Speichern von Information
- Senden von Informationen
- Ressourcensharing (Rechnerleistung, Speicherplatz, Drucker, andere Geräte)

Die Kommunikation zwischen zwei Rechnern im Netz erfolgt üblicherweise durch das Zusammenspiel zweier Prozesse: der Zugriff auf einen Netzwerk-“Service” erfolgt von einem Netzwerk-“Client”. Für Zugriff auf einen Mail-Service wird z.B. ein Mail-Client benötigt, für den Zugriff auf File-Service ein entsprechender Client. Letzterer wird im Allgemeinen auch als Netzwerkshell bezeichnet.



Der Service erlaubt üblicherweise die Mehrfachnutzung von verschiedenen Clients, insofern ist eine zentrale Aufgabe die gegenseitige Sperrung des Zugriffs von unterschiedlichen Clients auf gemeinsame Daten. Typische Mechanismen sind z.B.

- File - Locking (Kontrolle des Mehrfachzugriffs auf Fileebene)
- Record - Locking (Kontrolle des Mehrfachzugriffs auf Datensatzebene)
- Semaphore - Locking (Kontrolle des Mehrfachzugriffs auf Programme/Tasks)

Üblicherweise findet der Zugriff auf Dienste innerhalb einer logischen Verbindung statt. Der Verbindungsaufbau beinhaltet üblicherweise ein Login in den Dienst (Beglaubigung, Authentication s. später). Nach dem Verbindungsaufbau zum Service findet der weitere Informationsaustausch dann in Form einzelner Nachrichten statt. Bei einem Filezugriff erfolgt z.B. zunächst eine Abfragenachricht vom Client, der Service antwortet dann mit einer längeren Nachricht, die die angeforderten Daten enthält. Je nach Anforderungen der Kommunikationsinfrastruktur werden diese Nachrichten in eine Folge kleinerer Pakete zerlegt. (z.B. Ethernet Netze: max 1514 Byte Paketgröße)

Die Netzwerkinfrastruktur, die die logische Verbindung zwischen Client und Service ermöglicht, wird hier zunächst als eine Wolke dargestellt.

Diese Wolke kennzeichnet alle notwendigen Abläufe zur Steuerung der Verbindung, sie besteht zunächst aus den Übertragungsmedien, Geräten, die das Netzwerk implementieren (Switches, Router etc) und aus den Übertragungsprotokollen (sozusagen die Sprache, in der sich die Clientrechner und Server miteinander unterhalten). Wie wir später noch sehen werden, gibt es Protokolle für einzelne Aufgaben, die im Rahmen der Kommunikation erfüllt sein müssen, von Protokollen zur Sicherstellung der Netzinfrastruktur bis hin zu Protokollen zur Nutzung von Diensten. Durch den Mehrfachzugriff von verschiedenen Clients auf einen Service muß auch die Netzwerkinfrastruktur den Mehrfachzugriff unterstützen. Entsprechende Übertragungsprotokolle müssen dies regeln. Zudem stellen die verschiedenen Dienste auch ganz unterschiedliche Anforderungen an die Netzwerkinfrastruktur.

1.3 Nutzung und Verteilung von Diensten

Client und Server befinden sich üblicherweise auf unterschiedlichen Rechnern im Netz. Neuere Netzwerkbetriebssysteme (Novell NetWare seit Version 5 oder die verschiedenen Microsoft Windows Versionen) ermöglichen den Zugriff über einen Client direkt an der Konsole des Servers.

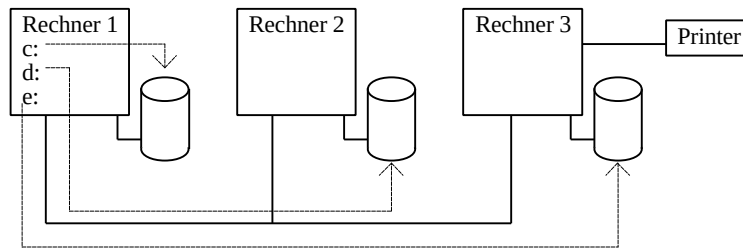
Folgende Begriffe kennzeichnen den Zugriff auf Dienste im Netz:

- Client: Dienstenutzer, Softwarekomponente eines Rechners die den Zugriff auf einen Dienst (Service) ermöglicht.
- Server: Diensteanbieter, bietet einen oder mehrere Dienste (Services) an.
- Peer: gleichwertiger Rechner, der Client und Serverfunktion parallel ermöglicht.

Damit wird in diesem Script bewusst zwischen den Begriffen Dienst (bzw. Service) und Server unterschieden.

In der Entwicklung von Netzen haben sich historisch die nachfolgenden Netzwerkkonzepte herauskristallisiert.

1.3.1 Peer to Peer Konzept



Erste Netze waren peer to peer Netze.

Aufgabe: gemeinsame Nutzung verschiedener Ressourcen (Festplatten, Drucker)

Kennzeichen: Rechner sind gleichzeitig Server und Clients

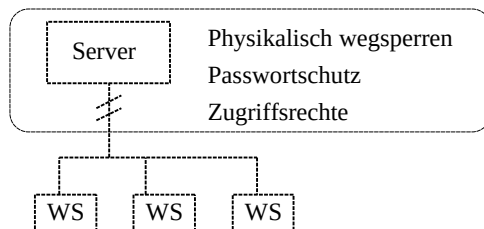
Beispiele:

Unix - Netze, Windows for Workgroups, Windows NT/2000/XP Workgroupnetze, Macintosh Local Talk

Probleme:

- Sicherheit gegen unbefugten Zugriff (Ausnahme Unix, Windows NT),
Zusätzliche Mechanismen (z.B. Bios Passwortschutz außerhalb des Betriebssystems können Sicherheit erhöhen)
- Sicherheit gegen Datenverlust

1.3.2 Client - Server - Konzept



Server:

Zentraler Speicher für Daten und Programme

Client:

Zugriff auf Services auf dem Server

Aufgaben des Server Betriebssystems

Zentraler Datenbestand, Daten möglichst effizient zur Verfügung stellen
Anbieten von Dienstprogrammen (Services)

Beispiel:

Ältere Novell Versionen (Novell 2.2, Novell 3.x)

Problematik:

Ausfallsicherheit

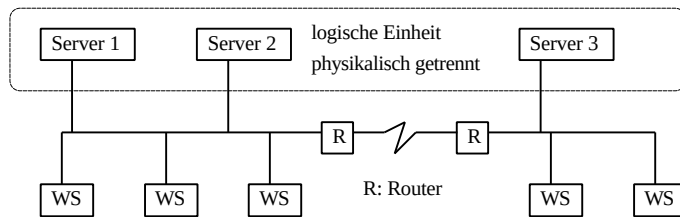
? spez. Sicherheitsfunktionen notwendig

Zugriffsgeschwindigkeit

? Fileserveroptimierung

? Netzoptimierung

1.3.3 Client - Network



In den letzten Jahren haben sich zunehmend verteilte Dienste etabliert, die von mehreren Servern ausgeführt werden. Damit loggt sich ein Client nur noch in „das Netz“ ein, es wird gleichzeitiger Zugriff auf alle Server und gleichzeitige Administration von verschiedenen Servern ermöglicht. Durch Entwicklung immer mächtigerer Verwaltungs- und Managementwerkzeugen bleiben auch komplexere Netze administrierbar.

Beispiele für verteilte Dienste:

- Dezentrale Benutzer- und Ressourcenverwaltung (Directory Service s. später)
- Verteilte Datenbanken
- Messaging Systeme

Bei Einsatz des gleichen Server-Betriebssystems auf den Servern im Netz und Verwendung gleicher Protokolle zur Kommunikation spricht man von einem homogenen Netz, ansonsten von einem heterogenen Netzwerk.

1.4 Architektur des Zugriffs

Je nachdem, wie die Komponenten einer Netzwerkanwendung auf Service bzw. Client aufgeteilt sind, gibt es grundsätzlich verschiedene Verfahren, wie die Kommunikation zwischen Client und Service abgewickelt wird.

Unterbrechung von Betriebssystemaufrufen

Dies ist der einfachste Fall, frühere Netze wurden oft auf diese Weise realisiert. Ein Netzwerkdienst stellt die komplette Funktionalität der Netzwerkanwendung dar, er behält die Kontrolle über alle gerade aufgebauten Netzwerkverbindungen (zustandsbehafteter Server). Der Client greift hier auf den Dienst zu, indem der Aufruf von Netzwerkkommandos durch den Client abgefangen und ins Netz umgeleitet wird. Das Betriebssystem, auf dem sich der Client befindet, ist nicht netzwerkfähig. Ältere Netzbetriebssysteme (z.B. solche mit DOS Clients) arbeiten auf diesem Weg, um Abfragen nach Dateien oder Druckern einfach statt zum lokalen Betriebssystem an das Netz weiterzuleiten.

Remote Operation

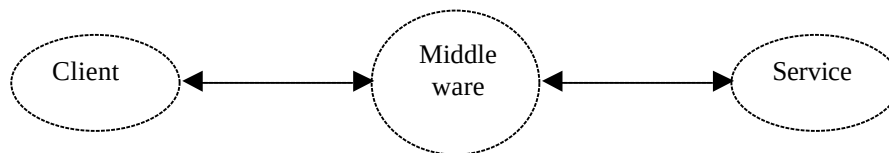
Hier ist das genaue Gegenteil der Fall. Der Client ist ein vollständiger Netzwerk-Client, der Server bzw. Service erfüllt Anfragen von Clients ohne Kontrolle, welche Clients gerade wie mit dem Server verbunden sind (zustandsloser Server). Es ist aus seiner Sicht auch nicht erkennbar, ob ein Zugriff intern oder über das Netz erfolgt. Der Remote Procedure Call (RPC) Mechanismus, über den Programme ferngesteuert werden können, ist ein Beispiel dafür.

Gemeinsamer Netzwerkzugriff

Dies ist der häufigste Fall in modernen Netzbetriebssystemen (Windows NT, Novell 4 mit Client32). Hier arbeiten Server und Client zusammen und koordinieren die Netzwerkkommunikation.

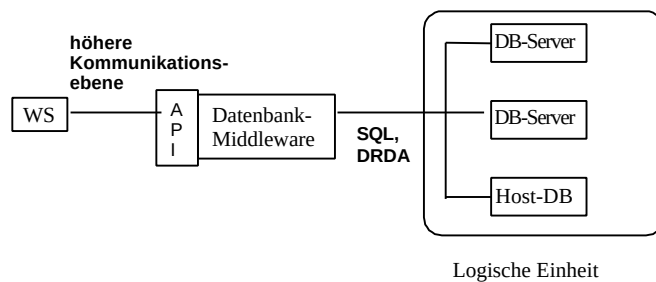
1.5 Multi-Tier Networking (2 ... n-tier Networking)

Anstelle des direkten Zugriffs von Clients auf Services (2-tier Networking) etabliert sich zunehmend eine weitere Verteilung der Komponenten innerhalb von Netzwerkanwendungen auf drei oder sogar mehr Knoten (3-tier bzw. multi-tier Networking). Im Bereich zwischen Client und Service spricht man dann von sog. Middleware.



Ziel ist hier die Verlagerung der Intelligenz vor allem weg vom Client auf weitere Komponenten. Dies erleichtert vor allem den Zugriff auf verschiedene Services, da dann ggf unterschiedliche Schnittstellen nur einmal für die zentrale Middleware, nicht aber für jeden Client realisiert werden müssen. So wie man Gateways (s. später) verwenden kann um eine Protokollumsetzung innerhalb einer Netzwerkverbindung zu einem entfernten Rechnersystem zu bewerkstelligen, kann man sich eine Middleware sozusagen als Gateway auf Anwendungsebene vorstellen. Der Client kann in diesem Fall über eine einzige Schnittstelle auf verschiedene Services zugreifen.

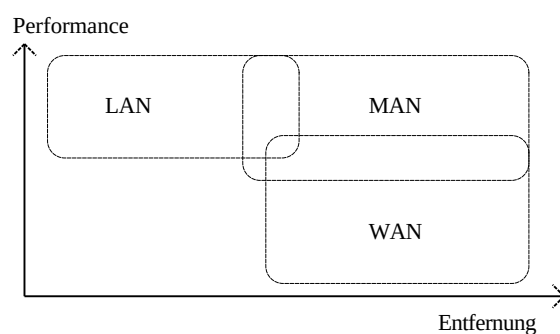
Beispiel: Steuerung der Kommunikation mit DB-Server (Datenbank Middleware)



Ein weiteres Beispiel einer solchen Lösung stellt z.B. die Verwendung einer verteilten Anwendung mit Zugriff aus einem LAN auf einen Großrechner dar, wo auf den unterschiedlichen Systemen verschiedene Methoden zur Datendarstellung und Transaktionssteuerung existieren und die Middleware dann die Aufgabe eine Anpassung der Transaktionskontrolle für beide Endsysteme übernimmt.

1.6 Einordnung von Netzen aufgrund ihrer Größe

Eine Möglichkeit Netze zu unterscheiden, ist der Vergleich von Größe (d.h. Ausdehnung) und typ. Übertragungsraten.



Man unterscheidet:

LAN: Local Area Network

- typisch: innerhalb eines Gebäudes (Firmengeländes, Campus)
- Übertragungsraten: heute bis 1Gbit/s, 10Gbit/s Standard kommt
- Datenkommunikation mehrerer *unabhängiger* Geräte
- Verwendete Übertragungsprotokolle gehen von sicheren phys. Verbindungen aus

WAN: Wide Area Network

- z.B. Datex-P, ISDN, Standleitungen, Modemverbindungen (Asynchron)
- typisch: Punkt zu Punkt, Punkt zu Multipunktverbindungen über *öffentl.* Leitungen

- Übertragungsraten 300 Bit/s ? 155 Mbit/s
- Verwendete Übertragungsprotokolle gehen häufiger von ausfallgefährdeteren phys. Verbindungen aus

MAN: Metropolitan Area Network

- Konzept (Protokolle) erlaubt Anwendungen für Daten, Audio und Video
- (Mehrdienstefähigkeit), Bandbreiten bis 155 MBit/s

Hinzu kommen neuere Netzwerkarten:

SAN: System (oder Storage) Area Network (s. später)

- Infrastruktur zur Kopplung von Rechnern, speziell Servern in sog. Clustern
- Technologien: SCSI, Fibre Channel, HIPPI (High Performance Parallel Interface)

PAN: Personal AreaNetwork

- Infrastruktur zur Vernetzung von Rechnern, Handys und Organizern über Infrarot oder Funkschnittstellen (z.B. Bluetooth)

2. Netzwerkdienste (bzw. Dienstprotokolle)

Die Anforderungen an heute verwendete Netzwerkdienste sind im Laufe der Entwicklung von Rechnernetzen aufgrund immer leistungsfähigerer Hard- und Software kontinuierlich gestiegen. Zunächst standen dabei Dienstprotokolle für sehr einfache Aufgaben wie Filetransfer, Terminalemulation und Electronic Mail in heterogenen Rechnerlandschaften im Vordergrund, so daß sich die Anforderungen zunächst grundlegend von denen in heutigen LANs unterscheiden. Erst später kamen Dienste wie der klassische File Service hinzu, der dem Benutzer den Zugriff auf entfernte Daten aus der gleichen lokalen Betriebssystemumgebung heraus genauso gestattet wie den lokalen Zugriff auf Daten.

Netzwerkdienste lassen sich in etwa in folgende Kategorien einordnen:

- Terminalemulation
- Filetransfer, Fileservice bzw. Webservice
- Multimedia Service
- Application Service (z.B. Datenbank)
- Message Service
- Print Service

LAN Betriebssysteme wurden zunächst mit eigenen Übertragungsprotokollen entwickelt (so verwendete Novell bis Ende der 90er Jahre das IPX/SPX Protokoll zur Paketüfung in Netzen, Microsoft das Netbios Protokoll). Kommerzielle Netzbetriebssysteme haben erst Mitte der 90er Jahre angefangen, die Internetprotokollwelt zu unterstützen. Der Durchbruch für die Internetprotokollwelt kam zunächst etwa 1980 mit der Integration in das Berkeley Unix und dann Ende der 90er Jahre mit der rasanten Verbreitung des Internets im kommerziellen Bereich. Heute unterstützen alle wichtigen Netzbetriebssysteme TCP/IP, Internet Technologien (typische Internet Dienstprotokolle) finden zunehmend auch innerhalb von LANs (als Intranets) Verwendung. Daher werden die nachfolgend beschriebenen Dienste auch nicht mehr nach Internetdienstprotokollen und anderen unterschieden.

2.1 Terminalemulation (Telnet)

Telnet ist ein einfaches Protokoll, um eine Terminalemulation auf einem entfernten Rechner zu realisieren. Es wird heute hauptsächlich noch im Bereich der Kommunikation zu Großrechnern verwendet. Bei der Kommunikation zwischen Terminals und Großrechnern wird im einfachsten Fall ein zeilenorientierter Zugriff durchgeführt (VT100/VT200). Übertragen werden im wesentlichen Tastatureingaben und Bildschirmausgaben. Im Laufe der Jahre wurden zunehmend Terminals durch PCs ersetzt, die für den Mainframezugriff dann eine Terminalemulation verwenden. TN3270 Protokoll (für IBM Mainframezugriff) bzw. TN5250 Protokoll (für IBM AS/400 Zugriff) sind entsprechende Internet-Protokolle, die auf auf Telnet aufsetzen.

2.2 Entfernter Dateizugriff via FTP

FTP, das File Transfer Protocol wurde entwickelt um einen Filetransfer zwischen zwei Rechnern, einem FTP Client und einem FTP Server, zu ermöglichen. Zunächst wird eine Verbindung zu einem entfernten Server aufgebaut, für den Dateitransfer ist dann ein Login nötig, häufig gestatten FTP Server den Datenzugriff über ein „anonymous“ Login. Ein FTP Zugriff ist allerdings nicht nur über einen speziellen FTP Client möglich, er kann auch aus einem Internet Browser wie z.B. Netscape erfolgen.

2.3 File Service

Im Gegensatz zum reinen Filetransfer ermöglicht ein File Service den Zugriff wie folgt:

- Bereitstellung von Daten und Programmen in einem Filesystem, optimiert auf möglichst effizienten Zugriff
- Benutzerabhängiger Dateizugriff
- Steuerung des gemeinsamen Dateizugriffs
- Zugriff auf entfernte Daten erfolgt transparent (genauso wie auf lokale Daten)
- Ermöglicht zentralen Backup im Netz
- Archivierungsfunktion durch Migration von kurzfristig nicht benötigten Daten auf 2nd-level Speicher

Ein wesentliches Kennzeichen ist, dass üblicherweise der Client im verteilten Filesystem nach Dateien sucht, um sie zu öffnen bzw. zu bearbeiten. Dabei besitzen die nachfolgend gegenübergestellten Server unterschiedliche Komplexität hinsichtlich Verwaltung, Sicherheit, etc.

2.3.1 NFS

NFS, das Network File System wurde Anfang der achtziger Jahre von SUN entwickelt und ist der Standard Filedienst auf UNIX Systemen. Es ermöglicht das Einbinden von Dateisystemen fremder Server in den Dateibaum eines Clients, um auf diese Daten genauso zugreifen zu können, wie auf eigene.

2.3.2 NCP (Novell Core Protokoll)

Novell war mit Netware in den 80er bis Mitte der 90er Jahre Marktführer im Bereich der PC basierenden Rechnernetze. Netware (heute in Version 6 erhältlich) erlaubte schon sehr früh den transparenten Benutzerzugriff auf Serverdaten wie auf lokale Daten, Novell gilt heute noch als Trendsetter für neue Technologien im Netzwerkbereich.

2.3.3 SMB (Microsoft Windows System Message Block)

Filedienst in Microsoft Windows NT/2000/.NET Umgebungen.

2.4 Web Service (http)

Ein Webservice ist im Grunde ebenfalls ein reiner Fileservice, mit dem einzigen Unterschied, dass Dateien über ein spezielles Abfrageprotokoll (http, Hypertext Transfer Protocol) übertragen werden. Http wird verwendet um im WWW (World Wide Web oder einfach Web) auf Information (Texte, Bilder, Multimediaobjekte) zuzugreifen. Das WWW integriert viele der anderen Internet Dienste und ermöglicht den Zugriff auf Information über ein einziges Benutzerinterface, des sog. Internet Browser. In diesem kann auf WWW Server genauso zugegriffen werden wie auf FTP Server, News Server, oder andere. Auf WWW Servern wird die Information in Form von HTML Seiten dargestellt. HTML (HyperText Markup Language) ist ein Datenformat, das über Schlüsselwörter (Links) auf andere Seiten im World Wide Web verweist und damit eine Verknüpfung von Information ermöglicht. Die Adressierung der entsprechenden Web Standorte erfolgt über sogenannte URLs (Uniform Resource Locator), deren Grundlage wiederum DNS ist. So wird eine typische Adresse z.B. wie folgt angegeben: <http://www.ee.fhm.edu>

Es hat sich sehr weitgehend eingebürgert, die Web Server mit www zu kennzeichnen, allerdings ist dies kein muß, der Name kann völlig beliebig gewählt werden. Die Seiten können Text in HTML format enthalten, zusätzlich können jedoch Graphiken und Programme als CGI oder Java Applets integriert sein.

Dies ist ein Beispiel klassischer Datenkommunikation. Hier spielen Latenzzeit (wie lange wartet ein Client auf eine Antwort) und Bandbreite (wie schnell wird Information übertragen) eine gewisse Rolle, andererseits sind hier gewisse Einschränkungen zumutbar. Zur Übertragung müssen jedoch Protokolle verwendet werden, die eine fehlerfreie Übertragung sicherstellen (Verlorene Pakete müssen z.B. auf jeden Fall nochmal gesendet werden). Jitter spielt hier keine Rolle.

2.5 Multimedia Service

Ein spezieller Typ von Daten auf einem Server sind Videodateien. Zunächst könnte man natürlich eine Videodatei von einem Fileserver erst vollständig zu einem Client übertragen und dann dort abspielen. Die Wartezeiten wären jedoch unter Umständen beträchtlich, zudem könnte ja bereits während des Übertragungsvorgangs das Abspielen erfolgen.

Insofern gibt es heute eigenen Video-Services in Netzen. Grundsätzlich stehen hier zwei Anwendungen im Vordergrund:

1. Video On Demand: Abspielen von Video über das Netz. Keine hohen Anforderungen bezüglich Latenzzeit
2. Videoconferencing: Videoübertragung in beiden Richtungen, daher straffe Zeiteinschränkungen, dafür ist geringere Qualität akzeptabel

Grundsätzlich erfordern Videoübertragungen über ein Netz eine andere Infrastruktur bzw. andere Übertragungsprotokolle als reine Datenkommunikation.

Auch hier gibt grundsätzlich zwei Möglichkeiten:

1. Protokolle, die die Reservierung einer bestimmten Bandbreite für die Videoübertragung reservieren.

Netzwerkdienste bzw. Dienststeprotokolle

2. Bereitstellung von genügend Bandbreite, so dass auch bei gleichzeitiger Datenkommunikation im Netzwerk noch genügend Reserve bereitsteht (Echtzeitprinzip).

Nachdem auch die Videoübertragung über langsamere Leitungen, z.B. über das Internet möglich sein sollen, ist es häufig nötig, Videos entsprechend zu komprimieren (MJPEG, MPEG1,2,4). Konfigurierbar sind bei Videoanwendungen oft die prozentuale Menge der zu übertragenden Frames und damit natürlich die Bildqualität.

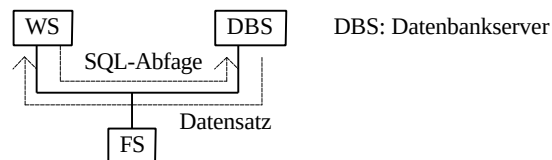
Hierfür wurde ein Protokoll entwickelt, das den Anforderungen an die Übertragung eines solchen Datenstroms besser genügt als die herkömmlichen Transportprotokolle, das **RTP** (Real Time Transport Protokoll).

2.6 Application Service

Ein Anwendungsserver stellt eine beliebige Anwendung zur Verfügung, die von entsprechenden Netzwerk Clients genutzt werden kann (z.B. Datenbank, Telefonanlage etc.). Ein häufig verwendeter Spezialfall eines Anwendungsservers ist ein Datenbankserver.

2.6.1 Datenbankserver

Ein Datenbankserver stellt ein intelligenteres Backend zur Datenhaltung zur Verfügung. Im Gegensatz zu einem Fileservice können hier über entsprechende Zugriffsprotokolle Suchanfragen nach Daten an den Server gestellt werden, dieser sucht nach den Daten und stellt diese Daten dann zur Verfügung. Damit stellt ein Application Service stellt ganz andere Anforderungen an Serverhardware und Netzwerkbetriebssystem als ein reiner Fileserver. Ein wichtiges Argument ist hier die Skalierbarkeit. Warenwirtschaftssysteme oder ähnliche datenbankbasierte Anwendungen müssen sehr oft in der Lage sein, mit der Größe von Unternehmen mitzuwachsen ohne dass bei einer bestimmten Größe eine Portierung auf ein anderes System notwendig wird. Daher finden sich sehr oft UNIX Systeme und Windows Systeme (skalierbar von PC basierende Systemen bis zur mittleren Datentechnik) als Basis für Datenbanken, bei großen Unternehmen finden sich nach wie vor oft Großrechnersysteme als Datenbankserver im Einsatz.



Typische Zugriffsprotokolle im LAN Umfeld sind:

SQL: Structured Query Language (Datenbankabfragesprache)
Client mit Datenbank - Frontend (SQL - Windows)

ODBC: Open Database Connectivity (Schnittstelle für den Zugriff auf Datenbanken unter Windows)

Häufig eingesetzte Produkte (allg. verwendbare Datenbankserver) sind Produkte von Gupta, Oracle, Informix, Sybase

2.6.2 Spezielle Datenbankservices

Die Aufgabe einer Datenbank ist es grundsätzlich, irgendwelche Daten in einem bestimmten Kontext zu speichern. Auch zur Verwaltung der Infrastruktur von Netzen ist es nötig, Informationen über das Netz zu speichern, um diese Clients bei Bedarf zur Verfügung zu stellen. Mehr dazu später.

2.7 Message Services

Ein Message Service stellt die Infrastruktur zum Versenden Nachrichten in Netzen zur Verfügung. Diese Nachrichten bestehen aus Dateien jedweden Inhalts bzw. von Programmcode zur Kopplung von Komponenten verteilter Anwendungen in Netzen. Dabei werden die entsprechenden Nachrichten bzw. Daten nicht in fest aufgebauten Kommunikationskanälen weitergeleitet, sondern wie bei einer klassischen Paketübertragung

zwischengespeichert und nach bestimmten Regeln weitergeleitet (es handelt sich um ein Store and Forward System).

Einfache Messaging Systeme stellen zentral vor allem einen Datenspeicher (Mailbox, die früher im Filesystem, heute aber innerhalb einer Datenbank realisiert wird) zur Verfügung, während ausgereifere Systeme zunehmend zusätzliche Funktionalität von der Clientseite auf die Serverseite verlagern (Webzugriff, Serverbasierende Mail-Regeln etc.)

2.7.1 SMTP

SMTP ist das Simple Mail Transfer Protocol, ein einfaches Protokoll zur Übertragung von Nachrichten zwischen Rechnern. Dabei werden sogenannte Mailserver definiert, auf denen die Mails für die Benutzer gespeichert werden und dann von einem zum nächsten Mailserver weitergeleitet werden.

2.7.2 POP3

Das POP3 (Post Office Protocol 3) ermöglicht den Zugriff von Mail Clients auf Mailboxen von Benutzern und Übertragung auf den lokalen Mailserver des Clients.

2.7.3 IMAP

IMAP (Internet Message Access Protocol) ist ein Internet Dienstprotokoll, das die Verwaltung von e-Mails auf einem zentralen Server ermöglicht, ohne das Clients ihre Mails auf den lokalen Rechner übertragen müssen.

2.7.4 Usenet News (NNTP)

NNTP (Network News Transfer Protocol) wird für spezielle News Server (Usenet) verwendet. News ist ein Kommunikationsmedium zum Austausch öffentlicher Information mit weltweit etwa 4000 Themenbereichen (News Gruppen). Die Beiträge können von allen Benutzern gelesen werden, in den meisten Newsgroups kann man auch eigene Artikel platzieren. Auch bei den Newsgroups gibt es einen hierarchischen Aufbau. Man kann bestimmte Newsgroups zu verschiedensten Themen „abonnieren“, d.h. man erhält regelmäßig neueste Informationen zu bestimmten Themen.

2.7.5 Kommerzielle Messaging Server

Die wesentlichen Beispiele kommerzieller Messaging Systeme sind:

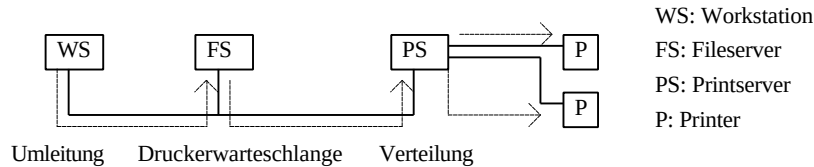
- Lotus Notes (auf allen gängigen Netzwerkbetriebssystemen)
- MS Exchange Server (auf Windows NT/2000/.NET)
- Novell Groupwise (auf Novell Netware)

Diese Server integrieren den Mailtransport, die Speicherung von Daten in Form eines Dokumentenmanagementsystems, Zugriff über betriebssystemspezifische Protokolle wie über die oben genannten Internetprotokolle etc.

2.8 Print Services:

Zur Mehrfachverwendung eines Druckers (bzw. eines Faxgeräts) muß eine logische Entkopplung der Druckaufträge erfolgen. Dies geschieht durch eine Warteschlange. Die Aufgabe eines Druckservers ist dann die

Abholung von Druckjobs aus der Warteschlange und Verteilung von Druckjobs auf verschiedene Drucker im Netz (Funktion eines Postboten).



Bedienung mehrere Fileserver und mehrerer Printer, z.B. eine Printqueue für mehrere Drucker, ein Drucker für mehrere Printqueues

Zentrale Aufgaben:

- Bereitstellung einer Warteschlange
- Weiterleiten der Druckjobs

Eine Formatierung der Druckjobs erfolgt normalerweise auf Clientseite. Dazu müssen auf dem Client Druckertreiber für alle Drucker, die verwendet werden sollen, vorhanden sein. Damit kommt der zentralisierten Verwaltung und Verteilung von Druckertreibern eine wachsende Bedeutung zu.

Als Internet Protokoll für Druckumgebungen wurde das **IPP** (Internet Printing Protocol) entwickelt.

2.9 Zeitservice

Die Verwendung einer netzwerkweit einheitlichen Zeit wird dann wichtig, wenn verteilte Dienste im Netz angeboten werden und ein Update von Information dezentral auf verschiedenen Servern im Netz erfolgen darf (z.B. in Netware zur Sicherstellung der richtigen Reihenfolge beim Update der netzwerkweiten Ressourcendatenbank (NDS)). Das entsprechende Internet Protokoll ist das **NTP** (Network Time Protocol)

2.10 Weitere Services

Die Liste von Services lässt sich erweitern, grundsätzlich können alle automatisierbaren Funktionen in einem Netz als Service realisiert werden, z.B. automatische Adressvergabe über einen Adressvergabe- Server (z.B. entsprechend DHCP Protokoll) oder ähnliches.

Üblicherweise sind mehrere Services auf einem Server realisiert, so ist häufig ein Fileserver noch mit Printservice, Faxservice, Web-Service, Routingfunktionalität usw. ausgestattet.

2.11 Namen und Adressen im Netzwerk

2.11.1 Dienstadressen

Jede Instanz, die mit einer anderen Instanz in einem Netzwerk (wie groß es auch immer sein mag) kommuniziert, benötigt eine eindeutige Adresse. Nachdem auf einem Rechner Client und Services oder auf einem Server mehrere Services koexistieren können, wird ein Adressschema benötigt, das es erlaubt, Rechneradressen zu vergeben und darüber hinaus eine Unteradressierung für Dienste auf einem Rechner zur Verfügung zu stellen. Diese Diensteadressen heißen allgemein Dienstzugriffspunkte bzw. Service Access Points (SAP), in verschiedenen Netzwerkprotokollen werden auch andere Bezeichnungen verwendet, in der TCP/IP Welt z.B. die Bezeichnung „Port“.

2.11.2 Namensauflösung

In Großrechnernetzen wird weitestgehend mit den Adressen der Kommunikationsteilnehmer gearbeitet, die Verwaltung eines größeren Netzes kann hier unter Umständen ziemlich aufwändig werden. Aufgrund der Menge von Rechnern in großen Netzen (wie z.B. dem Internet) ist es allerdings immer weniger praktikabel, mit Adressen zu arbeiten, statt dessen bevorzugen wir einfach zu merkende Namen für Rechner (So wollen wir in unserem Internet Browser nicht unbedingt <http://129.187.244.212> eingeben, um auf den Webserver der FH München zu gelangen, sondern wir bevorzugen einen Namen wie www.fhm.edu, ja richtig: www ist der beliebig zu vergebende Name für den Rechner, auf dem der Webservice läuft!).

Damit wird es nun notwendig, Namen in Adressen aufzulösen. Dies kann auf zweierlei Weise geschehen:

- jeder Client führt eine eigene Liste (das sog. Hosts File) in denen zu diversen Rechnernamen die zugehörigen Adresse vermerkt sind, oder
- es wird ein Dienst (noch eine Datenbank) im Netzwerk eingerichtet, über den die Adressen zu bestimmten Hostnamen abfragbar werden.

Letzteres wird durch **DNS** (Domain Name System) realisiert. So kann ein zentraler DNS Server von Rechnern befragt werden, unter welcher Adresse ein Rechner mit bestimmtem Namen zu finden ist. Darüberhinausgehend ist DNS ein weltweites System von Rechnern mit Adressinformation, hierbei wird eine Aufteilung des Internets in sog. Domänen vorgenommen. Innerhalb dieser Domänen werden dann die einzelnen Hosts zusammengefaßt. Die Domänenstruktur ist hierarchisch, beginnend von einer [Root] wird ein DNS Baum gebildet. Für jede Domäne sind bestimmte DNS Server verantwortlich, die entsprechenden DNS Server für darunterliegende Domänen werden dann logisch mit den DNS Servern der „höheren“ Domänen verbunden. Auf diese Weise kann die Namensauflösung eines Rechners oder einer Domäne hierarchisch durchgeführt werden.

Wir innerhalb eines Netzes nach einem bestimmten Namen gefragt, versucht zunächst der lokale DNS Server die Namensauflösung. Besitzt er keine entsprechenden Einträge in seiner Datenbank, wird die Namensauflösung zunächst an einen Root Server und dann an jeweils darunterliegende Server weitergeleitet.

2.11.3 Adressvergabe

Natürlich benötigt auch jeder Clientrechner eine eindeutige Netzwerkadresse (warum hier die sog. MAC Adresse der Netzwerkkarte im Rechner nicht ausreicht, wird später noch diskutiert). Auch hier gibt es mehrere Möglichkeiten für die Konfiguration des Rechners:

- Explizite Konfiguration des Rechners
- Automatische Adressvergabe durch einen Dienst im Netzwerk, bei dem sich der Client melden kann, um sich eine passende Adresse abzuholen (z.B. via DHCP, Dynamic Host Configuration Protocol, siehe später).

Selbstverständlich gilt dies nicht nur für Clientrechner, auch Server könnten in einem Netzwerk automatisch zugewiesene Adressen verwenden, dies hat aber einen erheblichen Nachteil. Wenn unterschiedliche Adressen zugewiesen werden, finden plötzlich Clients unter einer vorher bekannten Adresse den gewünschten Dienst nicht mehr, den sie nutzen wollen. Nachdem Serveradressen also konstant bleiben sollten, ist es gängige Praxis, diesen ihre Netzwerkadressen fest zuzuweisen.

2.11.4 Suche nach Ressourcen

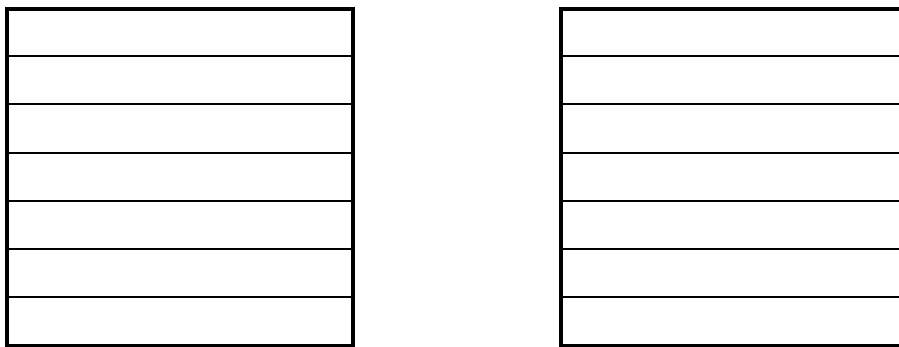
Nachdem der Clientrechner eine Adresse zugewiesen bekommen hat, kann er im nächsten Schritt den Namen des Dienstes auflösen und erhält damit die Adresse, zu der er eine Verbindung aufnehmen möchte. Fragt sich nur noch, woher der Client weiß, welche Dienste überhaupt im Netzwerk angeboten werden. Hierzu wird sozusagen ein Telefonbuch verwendet (engl. Directory), das als eigener **Directory Service** (aha: noch eine Datenbank) in einem Netzwerk eingerichtet werden kann (mehr dazu später). Weitergehend kann man natürlich nicht nur Information über Dienste speichern, sondern ggf. auch Inhalte, so wie dies z.B. Suchmaschinen im Internet tun.

3. Protokolle und Standards

Um Daten zwischen zwei Rechnern übertragen zu können, müssen zunächst einige Grundbedingungen erfüllt sein. Das beginnt damit, welcher Rechner unter welchen Bedingungen was senden darf. Es geht weiter über die Adressierung, die durchaus verschiedensten Anforderungen gerecht werden muß. Die meisten Protokolle erlauben eine Gruppierung von Adressen, um die Verteilung (das Routing) von Paketen im Netz einfacher gestalten zu können, zudem müssen auch Mechanismen zur Adressierung von Diensten vorhanden sein, da ja mehrere Dienste auf einer physikalischen Maschine koexistieren können. Eine weitere Aufgabe ist die Flußkontrolle, über die gesteuert wird, daß ein Sender einem Empfänger genau so viele Daten pro Zeiteinheit schickt, wie dieser verarbeiten kann und die Kontrolle der Gesamtübertragung durch das Senden von Bestätigungspaketen. Pakete sind Teil einer Kommunikationsbeziehung, die irgendwann einmal anfängt und wieder aufhört, parallele Kommunikation zu einem Dienst und die nötigen Codeumsetzung sind weitere Themen, schließlich endet das ganze damit, daß die Dienste, auf die zugegriffen werden soll, ja auch irgendwie im Netzwerk gefunden werden sollen.

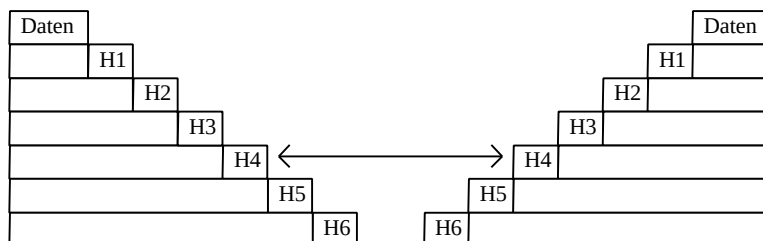
Dieser Aufgabenberg ist gewaltig, und nun könnte ein Hersteller eines Rechners natürlich auf die Idee kommen, diese Aufgabe für sich und seine Rechnerfamilie proprietär zu lösen. Diesen Ansatz verfolgten die Hersteller tatsächlich in den 70er Jahren, inzwischen ist man zu einem modularen Ansatz übergegangen. Jede diese Aufgabe wird unabhängig von anderen gelöst. Diesen Ansatz soll folgendes Kommunikationsmodell verdeutlichen.

3.1 Kommunikationsmodell



Modell zur Beschreibung von Kommunikation

- Schnittstellen klar definiert
- Jede Ebene fügt Information hinzu, die auf der Empfängerseite auf der gleichen Ebene ausgewertet wird
- Aufgabe jeder Ebene klar definiert, austauschbare Funktionsblöcke bei Kommunikation
- Ebenen werden auch als Schichten bezeichnet, jede Schicht setzt auf der nächstniedrigeren auf.



Anwendung auf Rechnerkommunikation

- Es gibt Standards für Funktion jeder Ebene
- Es gibt Standards für die Schnittstellen zwischen Ebenen

Ein übertragenes Datenpaket zwischen zwei Rechnern enthält also zunächst Header, dann nochmals Header usw. Er sieht also in etwa folgendermaßen aus:

Header 1	Header 2	Header 3	Header 4	Header 5	Header 6	Header 7	Daten
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------

Jeder Header enthält spezielle Information zur Übertragung, entsprechend der verschiedenen Aufgaben, die bei der Übertragung gelöst werden müssen. Typischerweise verweist jeder Header auf seinen Datenbereich, in dem sich tatsächlich dann der nächstfolgende Header befindet. Der Headeraufbau wird durch das verwendete Übertragungsprotokoll bestimmt, nachdem die Header aufeinander abgestimmt sind, also durch eine ganze Protokollfamilie (Protokollsuite, Protokollstack, Protokollstapel). Jeder Header, d.h. jedes Protokoll übernimmt dann ganz bestimmte Aufgaben zur Übertragung. Eine allgemeine Beschreibung dessen, was da alles festgelegt wird, liefert das sog. OSI Referenzmodell.

3.2 OSI Referenzmodell

Das OSI Referenzmodell ist ein Modell zur Beschreibung von Kommunikation. Es beschreibt die Teilaufgaben, die in einer Kommunikationsbeziehung zwischen Rechnern zu erledigen sind, ohne sich auf eine bestimmte Implementation festzulegen.

OSI: Open Systems Interconnection

Von der ISO (International Standards Organisation) entwickelt; seit 1983 Standard.

Schichten des OSI - Modells

7	Application layer
6	Presentation layer
5	Session layer
4	Transport layer
3	Network layer
2	Data Link layer
1	Physical layer

Anwendungsschicht
Darstellungsschicht
Kommunikationssteuerungsschicht
Transportschicht
Netzwerkschicht
Sicherungsschicht
Bitübertragungsschicht

Jede Schicht stellt der darüberliegenden einen Dienst mit bestimmten Funktionen zur Verfügung, der von dieser genutzt werden kann. Der Dienst ist über einen Dienstzugangspunkt (Service Access Point) erreichbar, die einzelnen Nachrichten zwischen den Schichten werden als Dienstprimitive (Service Primitives) bezeichnet.

Typische Aufgaben der Kommunikation

Die untersten beiden Schichten des OSI Modells beschreiben die Kommunikation innerhalb direkt verbundener Knoten in einem Netzwerk (Direktverbindungsnetzwerk). Typische Aufgaben sind:

1. Physical Layer

- Physikalische Schnittstellen (V24, RS232) (Spannungswerte, Impedanzen)
- Übertragungsverfahren
- Übertragungsparameter (Übertragungsgeschwindigkeit, Pulsformen, Leitungscodes)
- Prozedurale Anschlußbedingungen (Signalfolgen usw.)
- Dienste (Synchronisationsverfahren, Natzanschlußstruktur)
- Dienstqualität (Übertragungsfehlerrate, Verzögerung, Durchsatz, Verfügbarkeit)

Die Informationseinheit auf der Physical Layer wird als „Bit“ bezeichnet.

2. Data Link Layer

MAC - Teilschicht (**Media Access Control**)

Aufgaben:

Medienzugriff

physikalische Adressierung

z.B. Ethernet IEEE 802.3

z.B. Token Ring IEEE 802.5

LLC - Teilschicht (**Logical Link Control**)

Aufgaben:

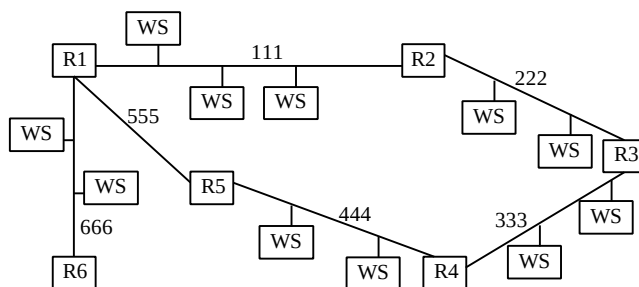
Fehlerkontrolle

Sicherung der Übertragung

Beispiele: SDLC, HDLC, IEEE 802.2 (LLC)

Die Informationseinheit ist hier ein „Frame“.

3. Network Layer



Mit der dritten Schicht verlassen wir die Direktverbindungsnetze und begeben uns auf die Ebene größerer Netzwerke (Internetworks), die aus mehreren Direktverbindungsnetzen bestehen. Geräte, die Pakete aus einem Direktverbindungsnetz in das nächste vermitteln, werden im LAN Bereich als Router bezeichnet. Im Vordergrund steht nun die „Ende zu Ende“-Kommunikation zwischen zwei entfernten Punkten eines größeren Netzes. Ziel ist, dass Protokolle der vierten Schicht des OSI Modells den darüberliegenden Dienstprotokollen einen vollständigen Transportmechanismus zur gesicherten Datenübertragung in Netzen anbieten. Damit sind folgende Aufgaben zu realisieren:

- logische Adressierung in Netzen
- Paketvermittlung ? Routing
- Flußkontrolle (teilweise)

- Kontrolle der Paketreihenfolge

Die Informationseinheit der Netzwerkschicht wird auch als „Datagramm“ bezeichnet

4. **Transport Layer**

- Zerlegung und Zusammenfügung von Paketen (Segmentierung)
- Adressierung von logischen Übertragungskanälen
- Adressierung von Diensten
- Adreß- / Namensauflösung
- Flusskontrolle
- Fehlersicherung

Die Informationseinheit der Transportschicht wird auch als „Segment“ bezeichnet.

Der Oberbegriff für die Informationseinheiten auf den Schichten 1 bis 4 ist das „Paket“. Zwischen der vierten und fünften Schicht des Referenzmodells kann man eine Grenze ziehen, die die Aufgaben von der reinen Zustellung von „Paketen“ von der Übermittlung von „Nachrichten“ (Schicht 5-7) trennt. In anderen Netzwerkmodellen werden diese Schichten, die im wesentlichen Dienstprotokolle repräsentieren, nicht aufgeteilt dargestellt, so wie sie oft auch aus einem Guß programmiert sind (s. DoD Modell später). Das Osi Referenzmodell beschreibt die Aufgaben wie folgt:

5. **Session Layer**

- Management von Sitzungen (Dialogkontrolle)

6. **Presentation Layer**

- Formatierung der Daten
- Verschlüsselung

7. **Application Layer**

- Bereitstellung einer lokalen Benutzeroberfläche
- Netzwerkservices (FTP, Directory services, Mail, MHS, SMTP)

3.3 DoD Modell

Im Gegensatz zum OSI Referenzmodell unterscheidet das Kommunikationsmodell der Internet Protokollwelt nur vier Schichten. Diese kann man den entsprechenden Schichten des OSI Referenzmodells in etwa wie folgt zuweisen:

OSI Modell

Application
Presentation
Session
Transport
Network
Data Link
Physical

DoD Modell

Process/Application
Host to Host
Internet
Media Access

Internet Protokolle

FTP, Telnet, NFS etc.
TCP, UDP
IP
Ethernet, Token Ring etc.

Diese Zuweisung lässt Rückschlüsse auf den Funktionsumfang von Protokollen der entsprechenden Schichten des DoD Modells zu.

3.4 Protokolle und Standards

Nun gibt es immer noch sehr viele Hersteller und damit eben auch verschiedene Ansätze für eine gelungene Aufteilung. Ergebnis sind die vielen Protokollfamilien, die heute im Markt nebeneinander her existieren. So gibt es z.B. die Internet Protokoll-Suite, die NetWare Protokollsuite, usw. Zudem ist es so, daß sich im Vergleich zum OSI Referenzmodell die Aufgaben des einen oder anderen Protokolls nicht unbedingt eindeutig einer bestimmten

Schicht zuordnen lassen. Manche erstrecken sich von ihrem Aufgabengebiet her über mehrere Schichten, andere füllen vielleicht nicht alle Teilaufgaben, die in einer bestimmten Schicht festgelegt sind.

Wer macht nun Protokolle?

Es gibt:

- De-Facto Standards: Standards/Protokolle, die sich am Markt entwickeln und oft aus herstellerspezifischen Entwicklungen hervorgehen
- De-jure Standards: Standards/Protokolle, die von Standardisierungsgremien verabschiedet werden

De-jure Standards werden von Standardisierungsgremien verabschiedet, z.B.

CCITT: Consultative Committee for International Telegraphy and Telephony
(X, V- Standards), heute ITU (International Telecommunication Union)

ISO: (1946 gegründet) 89 Staaten; Mitglied bei CCITT

ANSI: Amerikanischer Repräsentant der ISO

IEEE: Institute of Electrical and Electronical Engineers

Nachfolgende Tabelle zeigt eine ungefähre Zuordnung gängiger Protokollstacks zum OSI-Referenzmodell.

7	Application layer	IBM SNA	DEC DNA	NFS	Novell NCP	OSI	Apple- talk
6	Presentation layer						
5	Session layer						
4	Transport layer			TCP	SPX		
3	Network layer			IP	IPX		
2	Data Link layer			Ethernet, Token Ring, FDDI			
1	Physical layer						

4. Direktverbindungsnetzwerke

Die direkte Verbindung zweier oder mehrerer Rechner im Netzwerk wird hier als sogenanntes Direktverbindungsnetzwerk bezeichnet (teilweise auch als Netzwerksegment bezeichnet).

4.1 Protokollaufgaben

Die wesentliche Aufgabe von Protokollen der MAC Teilschicht ist (wie der Name schon aussagt) die Kontrolle des Medienzugriffs. Die Grundidee eines LANs ist es, zwischen Rechnern Datenzugriff zu erlauben und auf Ressourcen und Dienste zugreifen zu können. Damit teilen sich die entsprechenden Rechner ein gemeinsames Übertragungsmedium (shared LAN). Zugriffsprotokolle (wie z.B. Ethernet, Token Ring FDDi, etc.) steuern den Zugriff der einzelnen Rechner auf das Übertragungsmedium und die Paketzustellung an andere Rechner im gemeinsamen Direktverbindungsnetz. Neben dem reinen Zugriffsverfahren und der Adressierung werden in den Zugriffskontrollstandards auch typ. Physical Layer Eigenschaften festgelegt (z.B. Leitungscodierung, Kabellängen, Schnittstellen etc.).

Die Kommunikationsmodelle beschreiben diese Aufgaben unterschiedlich:

- Das OSI Referenzmodell beschreibt die Aufgaben solcher Protokolle auf der MAC Teilschicht (Media Access Control) der Data Link Layer und der Physical Layer,
- das DoD Modell fasst alle Aufgaben in der Media Access Layer zusammen

Obwohl heute zunehmend switched LANs im Einsatz sind, bei denen Punkt zu Punkt Verbindungen zwischen zwei Kommunikationspartnern geschaltet werden, sind die allermeisten Zugriffskontrollstandards (wie u.A. auch Ethernet) auf einen möglichen Mehrpunktbetrieb ausgelegt.

Näheres zur Zugriffskontrolle folgt später.

4.2 Kommunikation im Direktverbindungsnetz

Zunächst soll die Kommunikation zwischen zwei Rechnern im Direktverbindungsnetz nur kurz beschrieben werden, eine detaillierte Beschreibung der Vorgänge folgt zu einem späteren Zeitpunkt.

4.2.1 Netzwerkkarte

Ein Rechner wird durch Verwendung einer Netzwerkkarte (NIC: Network Interface Controller) netzwerkfähig. Auf dieser wird über einen eigenen Controller das Zugriffsprotokoll (z.B. Ethernet) komplett abgewickelt. Jede Netzwerkkarte erhält ab Werk eine eindeutige 48 Bit lange Identifikationsnummer, die aus zwei Teilen besteht:

Teil 1: 24 Bit OUI: Organisation Unique ID, Eindeutige Kennzeichnung eines Netzwerkkartenherstellers

Teil 2: 24 Bit: Vom Hersteller vergebene individuelle Nummer der Netzwerkkarte

Damit sind die IDs weltweit eindeutig und können auf der MAC Teilschicht als Rechneradresse verwendet werden (sog. MAC Adresse bzw. Hardwareadresse). Ein Datenpaket (Frame, Rahmen), das zwischen zwei Rechnern übertragen wird, enthält Quell- und Ziel-MAC-Adresse.

4.2.2 Ethernet Protokoll

Nachfolgend ist Rahmenformat eines Ethernet_II Frames gezeigt, der heute in Netzen überwiegend Verwendung findet.

Ethernet_II

Präambel	Ziel-adresse	Quell-adresse	Typ	Daten	CRC
----------	--------------	---------------	-----	-------	-----

Präambel:

Die Präambel dient durch die Bitfolge 10101010 pro Byte im wesentlichen zur Synchronisation bei der Übertragung (Bei Ethernet II handelt es sich um 7 Byte und einen SFD (Start of Frame Delimiter) Frame, der die Bitfolge 10101011 besitzt, und den Beginn des Frames markiert).

Typ:

Dieses Feld enthält einen Code, der das nächsthöhere Protokoll festlegt. Damit wird dem Empfänger mitgeteilt, entsprechend welchem Protokoll die ersten Bits und Bytes im Datenfeld des Ethernet Frames zu interpretieren sind.

CRC:

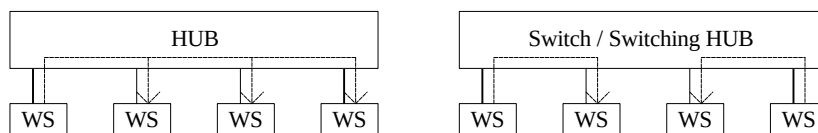
Cyclic Redundancy Check: Dient zur Erkennung von Übertragungsfehlern. Damit wird dem Empfänger einer Nachricht ermöglicht, zu erkennen, ob ein Übertragungsfehler vorliegt. Liegt ein Fehler vor, wird das Paket vernichtet, eine Fehlerkorrektur wird nicht vorgenommen. Es bleibt dann den Protokollen höherer Schichten vorbehalten für eine Wiederholung des Sendevorgangs zu sorgen.

4.2.3 Andere Zugriffsprotokolle

Weitere Zugriffsprotokolle, die in LANs heute teilweise noch im Einsatz sind, sind das Token Ring und das FDDI Protokoll. Hier werden zwar andere Mechanismen zur Zugriffskontrolle verwendet, die Adressierung zur Zustellung von Paketen läuft hier jedoch ähnlich ab wie im Ethernet.

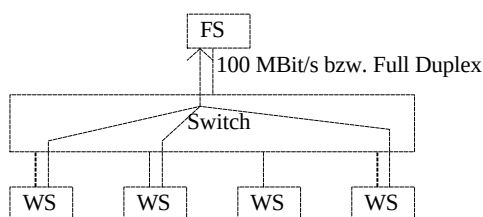
4.3 Frame Switching

In den letzten Jahren hat sich ein Wandel von Hub-basierten shared LANs hin zu switched LANs vollzogen. Hierbei wird die Technologie der Lasttrennung von Bridges verwendet, um den Datenverkehr zwischen zwei Stationen auf die entsprechenden Ports eines Switches zu beschränken und so parallel andere Kommunikationskanäle über andere Ports zu ermöglichen. Bei Hubs wird das Signal, das an einem Port ankommt, auf alle anderen Ports weitergeleitet. Switches schalten dagegen den Datenverkehr eines Ports je nach Zieladresse auf genau einen anderen Port.



Mit einem Switchport werden im Allgemeinen eine oder wenige Stationen (als Shared LAN) verbunden (bei Ethernet Switches spricht man in diesem Zusammenhang von einer sog. Collision Domain). Im Extremfall werden einzelnen Stationen mit den Ports eines Switches verbunden, dies gilt vor allem für Server um die volle Bandbreite des Kanals für den Server zur Verfügung stellen zu können.

Switches erlangten im Ethernet zum Zeitpunkt der Umstellung von Netzwerken von 10Mbit/s auf 100Mbit/s Bedeutung. Hier hat man oft die Serververbindung auf 100Mbit/s aufgerüstet, und die vorhandene Verkabelung und Netzwerkkarten auf Clientseite weiterverwendet. Durch das Multiplexen mehrerer 10Mbit/s Kanäle auf 100Mbit/s ergab sich so ein Geschwindigkeitsvorteil in Netzen, die bis dahin eine hohe Auslastung zeigten.



4.4 LLC Teilschicht

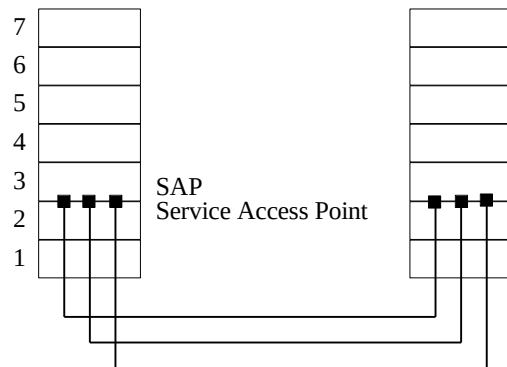
Das OSI Referenzmodell beschreibt als weitere wichtige Aufgabe die Sicherung der Übertragung zwischen zwei Knoten innerhalb eines Netzwerksegments (Direktverbindungsnetzes). Hier kommen Mechanismen zur Fehlererkennung und ggf. der Flusskontrolle zum tragen, so wie sie später im Detail noch besprochen werden. Im Gegensatz zu WANs spielt in LANs diese Teilschicht jedoch nur in seltenen Fällen eine Rolle, die genannten Aufgaben werden auf höheren Schichten (i.A. von Protokollen auf der Transportschicht) wahrgenommen. Das nachfolgend gezeigte LLC Protokoll (Logical Link Control) ist sozusagen die LAN Variante des SDLC bzw. HDLC Protokolls, die Control-Bits übernehmen dabei im wesentlichen die gleichen Aufgaben der

Übertragungssteuerung wie bei HDLC. Die Adressen reduzieren sich hier auf SAP (Service Access Point) Adressen, über die dann Dienste adressierbar sind, nachdem das Protokoll mit einem LAN Zugriffsprotokoll (Ethernet oder Token Ring) eingesetzt wird, in dem dann entsprechende Hostadressen festgelegt sind.

LLC findet man z.B. in IBM Mainframe Umgebungen zur Adressierung von Dienstprofilen auf Mainframes (über SAPs). Hierbei dient LLC zur Unteradressierung und Steuerung der Kommunikation mit Diensten.

LLC - Frame:

Header	DSAP Adresse	SSAP Adresse	Control	Information	CRC
--------	--------------	--------------	---------	-------------	-----



Ein anderer Einsatz zeigt sich aber auch in Verbindung mit verschiedenen Ethernet Frame Typen. In Novell Umgebungen wird z.B. Ethernet 802.2 oder Ethernet SNAP (beide beinhalten LLC) lediglich zur Protokolladressierung des nächsthöheren Protokolls (in diesem Fall IPX) verwendet.

5. Netzwerkschicht

Direktverbindungsnetze werden über Router zu größeren sogenannten Internetworks verbunden. Damit ergeben sich neue Aufgaben, die durch entsprechende Protokolle festgelegt werden müssen.

Typische Aufgaben von Protokollen der Netzwerkschicht sind:

- Netzwerkweite Adressierung
- Verhinderung endlos kreisender Pakete in Netzen
- Übertragungsfehlerkontrolle (i.A. bei WAN Protokollen)
- Routing (Routenfindung und Abgleich von Routinginformation in Netzen)

5.1 Netzwerkweite Adressierung

Eine der wesentlichen Aufgaben von Protokollen der Netzwerkschicht ist eine netzwerkweite Adressierung. Jedem Rechner, d.h. jeder Netzwerkkarte mit einer Hardwareadresse wird nochmal eine Netzwerkadresse zugeordnet.

Damit stellt sich zunächst die Frage, warum die bereits vorhandenen Hardwareadressen nicht ausreichen, die von den Netzwerkkartenherstellern sowieso weltweit eindeutig vergeben werden. Damit wäre ja auch in einem weltweiten Netz wie z.B. dem Internet jeder Rechner eindeutig identifizierbar. Die Antwort zu dieser Frage liegt unter anderem in der Verwaltbarkeit von Adressen auch im Hinblick auf Unterstützung von Routing. Netzwerkadressen können nach logischen Gesichtspunkten organisiert werden, Hardwareadressen sind durch die Hersteller der Netzwerkkarten vorgegeben. Wollte man eine Liste von allen Teilnehmern im Internet machen, wäre diese Liste relativ unübersichtlich. Sie wäre ungefähr genauso unübersichtlich wie ein Telefonbuch mit allen Telefonteilnehmern in Deutschland, wenn die Telefonnummern in der Reihenfolge der Anmeldung der Teilnehmer am Netz der Telekom eingetragen würden. Wie sieht die Abhilfe der Telekom aus? Nun, es werden Ortskennzahlen verwendet, unter anderem werden auch die Telefonnummern selbst nach Stadtteilen organisiert.

Eine solche Organisation findet auch bei der Verwendung von Netzwerkadressen statt. Die Adressen werden nicht Rechner für Rechner vergeben, sondern netzwerkweise, wobei ein Netzwerk unterschiedlich viele Rechner enthalten kann.

Eine Netzwerkadresse besteht grundsätzlich aus zwei Teilen:

- Netzwerkanteil (entspricht im obigen Beispiel einer Vorwahl)
- Hostanteil (entspricht der individuellen Nummer des Teilnehmers)

Damit wird eine Zuordnung des Netzwerkanteils einer Netzwerkadresse pro Netzwerksegment ermöglicht, die Unterscheidung der einzelnen Rechner erfolgt über den Host Anteil.

Das Adressformat ist je nach verwendetem Netzwerkprotokoll unterschiedlich, wie nachfolgende Beispiele zeigen.

Novell IPX Protokoll

Das IPX Protokoll verwendet 20 stellig hexadezimale Adressen, aufgeteilt in einen festen Netzwerkanteil (8-stellig bzw. 32bit) und einen Hostanteil (12stellig bzw. 48bit). Letzterer ist identisch mit der Hardwareadresse des jeweiligen Rechners und wird daher nicht individuell konfiguriert.

Damit wird eine Zuordnung einer Netzwerkadresse pro Netzwerksegment ermöglicht, die Unterscheidung der einzelnen Rechner erfolgt dann ausschließlich über die Hardwareadresse.

Internet Protokoll (IP):

Hier werden 32bit Netzwerkadressen verwendet. Die Aufteilung in einen Netzwerk und Hostanteil ist variabel, die möglichen Adressen werden in verschiedene Adressbereiche (sogenannte Klassen) aufgeteilt, die eine unterschiedliche Aufteilung in Netzwerk und Hostanteil vorgeben. Zur Unterscheidung werden die ersten Bits der IP Adresse herangezogen. Das folgende Bild zeigt eine Aufteilung der Klassen, der Netzwerkanteil ist jeweils mit x, der Hostanteil mit y markiert.

Klasse	32 bit Adresse	Dezimale Adresse	Anzahl mögl. Netzwerke	Anzahl mögl. Hosts
Class A	0xxxxxxx.yyyyyyyy.yyyyyyyy.yyyyyyyy	1-126	126	16.777.214
Class B	10xxxxxx.xxxxxxxx.yyyyyyyy.yyyyyyyy	128-191	16.384	65.534
Class C	110xxxxx.xxxxxxxx.xxxxxxxx.yyyyyyyy	192-223	2.097.152	254

Class A bedeutet nun, daß das erste Byte als Netzwerkadresse und die übrigen als Hostadresse interpretiert wird, bei Class B sind es die ersten beiden Bytes, bei Class C die ersten drei Bytes. Die restlichen möglichen Adressen sind für besondere Aufgaben reserviert. Die IP Adressen jedes einzelnen Rechners sind individuell festzulegen. Sie können entweder für jede Station einzeln konfiguriert werden, die andere Möglichkeit ist, daß die einzelnen Rechner ihre IP Adressen bei Bedarf von einem zentralen Rechner über ein spezielles Protokoll (überwiegend DHCP) abfragen. Auf dem DHCP Server wird dann ein bestimmter Adressbereich zur dynamischen Vergabe bereitgehalten.

5.2 Das Internet Protokoll (IP)

IP ist als Übertragungsprotokoll das Basisprotokoll für die Internet Protokollwelt. Es wird der dritten Schicht (Netzwerkschicht) des OSI Referenzmodells zugeordnet und setzt damit auf den Protokollen der Data-Link Layer (z.B. Ethernet) auf, die den Zugriff auf das Übertragungsmedium steuern. Im OSI Referenzmodell nach oben gesehen stellt es die Basis für die Transportprotokolle TCP oder UDP (Schicht 4 des OSI Modells) dar.

Das folgende Bild zeigt den Aufbau eines IP Frames:

4 byte				
Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				
Data				

Die einzelnen Felder haben folgende Bedeutung:

- Version: Die Version des IP Protokolls, die aktuelle Version ist 4.
- IHL: IP-Header Länge in 32bit Worte
- Type of Service: Angabe für die Upper Layer Protokolle, wie ein spezielles IP-Paket zu handhaben ist.
Festgelegt werden:
 - die Wichtigkeit Pakets in Werten von 0 bis 7 und
 - Je ein Bit für Festlegung auf
 - niedrige Verzögerungszeit
 - hohe Übertragungsgeschwindigkeit
 - hohe Zuverlässigkeit
 - bei der Übertragung
- Total Length: Gesamtlänge des IP-Pakets in Bytes, inklusive Daten und Header
- Identification: Eindeutige Identifizierung eines einzelnen Pakets
- Flags: Zwei der drei Bits legen fest, wie und ob ein Paket von der Transportschicht in mehrere IP-Fragmente unterteilt wurde.
- Fragment Offset: Dieser Wert gibt das Startbyte des zweiten oder weiteren Teils eines fragmentierten Pakets in Byte an
- Time to Live: Dieses 8 bit Feld legt eigentlich fest, wie lange ein Paket im Netz weitergereicht werden darf, bis es vernichtet wird. Ursprünglich war die Angabe eine Zeit in Sekunden, nachdem aber kein Router Pakete länger als 1s behält und damit der Wert bei jedem Übergang um 1 dekrementiert wurde, entspricht die TTL der maximalen Anzahl von Routerübergängen (Hops), bis das Paket nicht mehr weitergeleitet wird.

- ✗ Protocol: Legt fest, welches Transportprotokoll (z.B. TCP oder UDP) verwendet wird.
- ✗ Header Checksum: Fehlerkontrolle des IP Headers
- ✗ Source und Destination Address: 32bit Quell- und Zielnetzwerkadressen (siehe später)
- ✗ Options: Dieses Feld kann unterschiedliche Längen aufweisen. Wird für zusätzliche Daten verwendet wie z.B. für Source Routing, Zeitinformaton, aber auch zum Auffüllen mit Bits (Padding) um z.B. die minimale Paketlänge im Ethernet garantieren zu können.

5.3 IP-Adressierung

Als Protokoll der Netzwerkschicht erlaubt IP eine reine Netzwerkadressierung, d.h. Send- und Empfangsadressen sind hardwareunabhängig und damit in einem abgeschlossenen Netz frei wählbar. Bei Anschluß an das Internet erhält man eine oder mehrere Adressen von einem sogenannten Service Provider, also einer Firma, die einem eine physikalische Verbindung ins Internet zur Verfügung stellt.

Der Adressraum umfaßt dabei 32 Bit, wobei jeder Host (Rechner mit IP-Nummer) eine eindeutige IP Nummer zur Kommunikation benötigt. Dies schränkt die Menge der weltweit eindeutig verfügbaren Adressen deutlich ein. Je nach Klasse wird nun ein unterschiedlicher Anteil von Bytes jeweils als Netzwerkadresse und Hostadresse interpretiert. Darüberhinaus gibt es aber auch die Möglichkeit, diese feste Zuordnung zu verlassen, und eine andere individuellere Aufteilung von Hostanteil und Netzwerkanteil festzulegen. Um dies bei der Konfiguration zu können, wird neben der reinen IP Adresse eine Angabe benötigt, wie diese Adresse jeweils zu interpretieren ist. Dazu wird jedem Rechner neben der IP Adresse auch noch eine sogenannte Netzwerkmaske zugeteilt, anhand derer die Aufteilung zu erkennen ist. Die Netzwerkmaske wird dabei so definiert, daß der Netzwerkanteil einer Adresse mit 1-bits, der Hostanteil mit 0-bits dargestellt wird.

Natürlich werden hier nicht die einzelnen Bits dargestellt, sondern die Bytes werden zu Bytes zusammengefaßt und wie die Adressen selbst mit einem „.“ getrennt. Die einzelnen Bytes werden nun in dezimaler oder, bei Netzwerkmasken, in hexadezimaler Schreibweise dargestellt. Ein Beispiel für eine typische Netzwerkadresse wäre damit:

Adresse:	183. 94. 51. 45	bzw.	183. 94. 51. 45
Netzwerkmaske:	255.255. 0. 0	bzw.	FF. FF. 0 . 0

Die Netzwerkmaske definiert für die ersten beiden Bytes lauter 1-Bits, für den Hostanteil lauter 0-Bits. Diese Vorgehensweise erlaubt es einem Rechner, durch eine einfache bitweise UND-Verknüpfung aus einer beliebigen Adresse den Netzwerkanteil festzustellen. Dies würde sich wie folgt darstellen:

Adresse:	10110111 . 01011110 . 00110011 . 01001101
Maske:	<u>11111111 . 11111111 . 00000000 . 00000000</u>
Netzwerkanteil:	10110111 . 01011110 . 00000000 . 00000000 , festgestellt durch UND Verknüpfung

Eine IP-Adresse wird damit eigentlich erst interpretierbar, wenn die Netzwerkmaske bekannt ist. Bei einer klassenvollen Adressierung ist sie implizit bekannt, bei einer klassenlosen Adressierung muß die Maske explizit angegeben werden. Hier setzt sich eine Kurzschreibweise immer mehr durch, die statt der Maske an sich nur die Anzahl der 1 Bits in der Maske angibt. Damit wird aus obigem Beispiel die Adresse 183.84.51.45/16 für 16 1 Bits in der Netzwerkmaske. Voraussetzung ist hier natürlich der übliche Fall, dass die Maske eine fortlaufende Folge von 1 Bits besitzt.¹

¹ Dies muss allerdings nicht so sein, RFC 950 zeigt den sehr seltenen Fall der Verwendung sogenannter non-contiguous Subnet Masks.

Einige Konventionen, die es bei der Vergabe von IP-Adressen zu beachten gilt, sind:

Netzwerkadressen:

Adressen mit Hostanteil 0 repräsentieren nicht eine einzelne Station, sondern einen ganzen Netzwerkstrang, z.B. repräsentiert die Adresse 162.120.0.0 eine Class B Netzwerkadresse, die Hosts in diesem Netzwerk besitzen Adressen von 162.120.0.1 bis 162.120.255.254.

Broadcast:

Adressen mit einem Host- oder Netzwerkanteil, der nur aus 1-Bits besteht, sind nicht erlaubt, da diese Adressen für Broadcasts, also Pakete an alle Hosts eines Netzwerks, reserviert sind. Nach obigem Beispiel wäre also die Class B Adresse 162.120.255.255 verboten.

Loopback (bzw. Localhost):

Die Adresse 127 ist eine spezielle Adresse, die Funktionalitätstests zum Netzwerk erlaubt. Sie erlaubt die Kommunikation eines Clients und eines Hosts auf demselben Rechner via TCP/IP, ohne Daten tatsächlich auf das Netzwerk zu senden. Auch bei Broadcasts sorgt das Loopback-Interface in einem Rechner dafür, daß Daten die auf das Netz gesendet werden auch die eigene Station mit einschließen. Meistens wird 127.0.0.1 als Loopback-Adresse verwendet.

5.4 Paketzustellung im IP-Netz

5.4.1 Das ARP Protokoll

Im folgenden ist nun zu klären, wie ein Datenpaket von einer Station zur nächsten gelangt. Die Kenntnis der (logischen) Netzwerkadresse einer Zielstation reicht nicht aus, um z.B. einen Ethernet Frame zu bauen, in dem die Hardwareadresse der Zielstation enthalten sein soll. Hierzu ist eine Adressauflösung nötig, d.h. um einer IP Station ein Paket senden zu können, muß aus der IP Zieladresse erst die entsprechende Hardwareadresse der Netzwerkkarte der Zielstation ermittelt werden.. Diese Aufgabe übernimmt das ARP Protocol (Address Resolution Protocol). Die sendende Station sendet einen Broadcast mit der Nachricht "Ich habe hier ein Paket für die Netzwerkadresse soundso, wer ist das ?". Die Zielstation antwortet an die Hardwareadresse des Senders mit ihrer Netzwerk- und Hardwareadresse. Daraufhin kann das Paket an die Station mit der richtigen Hardwareadresse zugestellt werden.

5.4.2 Routing am Beispiel IP

Wir haben nun die Begriffe Netzwerk und Host eingeführt. Im folgenden wird ein Netzwerksegment (Direktverbindungsnetzwerk) als eine Verbindung zwischen mehreren Hosts betrachtet, die vom nächsten Netzwerk durch einen Router getrennt ist. Ein Router wird verwendet, um unterschiedliche Netzwerksegmente in einem Netzwerk zu verbinden.

Nachdem sich die Funktionsweise eines Routers nur mit Netzwerkadressen beschreiben läßt, wird Routing als generelle Aufgabe an sich der Schicht 3 des OSI Referenzmodells zugewiesen. Dies stellt sich ungefähr wie folgt dar:

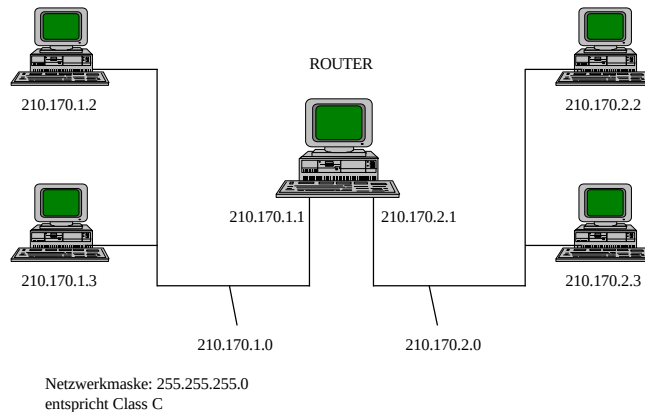
3	3	NETWORK
2	2	LLC / MAC
1	1	PHY

Routing setzt also eine logische Trennung von Netzen in Subnetze voraus. Es gibt (ältere) LAN Protokolle, die keine Netzwerkadressierung verwenden und damit auch nicht routingfähig sind (Local Area Transport LAT (Digital), NETBIOS (IBM, Microsoft)). Hier können größere Netze nur mit sog. Netzwerkbrücken (Bridges, Switches s. später) aufgebaut werden.

Protokollstacks, die routbare Protokolle auf der Netzwerkschicht beinhalten sind z.B. TCP/IP(DoD), IPX/SPX (Novell), OSI, Apple Talk, DECNET.

Datenpakete, die in ein anderes Netzwerksegment geschickt werden sollen, werden explizit einem Router zugestellt, dieser hat dann die Aufgabe, die Pakete basierend auf Informationen in seiner Routingtabelle weiterzuleiten. Insofern muß auch ein Router Netzwerkadressen nach Netzwerk und Hostanteil unterscheiden können. Die nachfolgende Abbildung zeigt ein IP-Netzwerk mit zwei Class C Netzwerksegmenten.

Jeder Netzwerkkarte in jedem Host wird eine eigene IP Nummer zugeteilt, die beiden Teilnetze erhalten z.B. die Nummern 210.170.1.0 bzw 210.170.2.0. Die 0 kennzeichnet das jeweilige Netzwerksegment, die Nummer 1 bis 254 sind für die Rechner im jeweiligen Segment vorgesehen.



Wie funktioniert nun hier der Sendevorgang von Paketen?

Je nachdem, ob sich die Zielstation im gleichen Netzwerksegment befindet oder in einem anderen, ergeben sich zwei Möglichkeiten:

1. Die Zielstation befindet sich im gleichen Teilnetzwerk:

Die Zielhardwareadresse wird mittels ARP ermittelt und das Paket im lokalen Segment zugestellt.

2. Die Zielstation befindet sich in einem anderen Netzwerk:

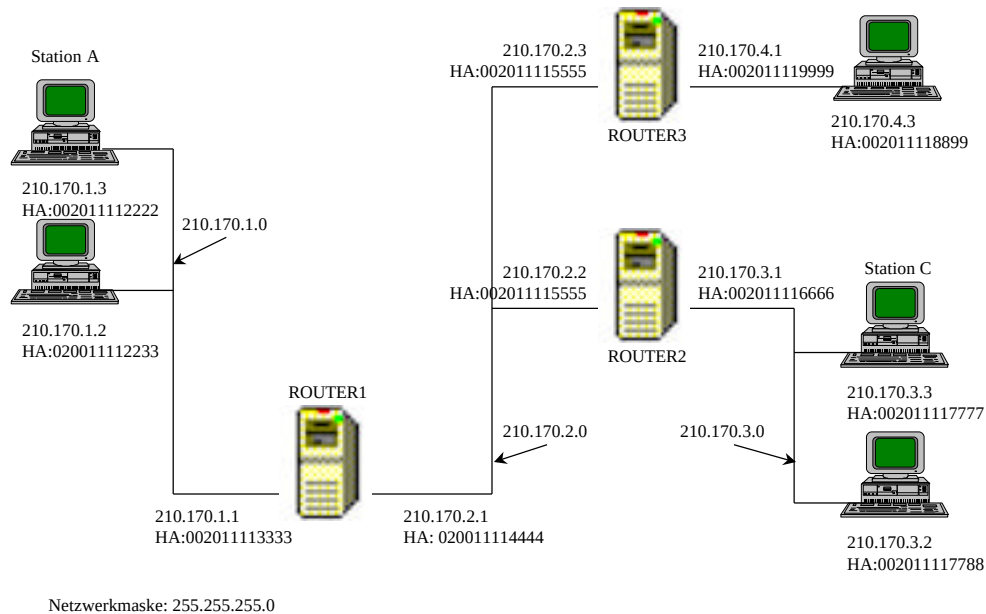
Die Station stellt dies aufgrund einer anderslautenden Netzwerkadresse der Zielstation fest, sie sendet das Paket an den Router. Dieser hat dann die Aufgabe, das Paket an die entsprechende Station im anderen Segment zuzustellen. Damit das Paket dem Router zugestellt werden kann, muß der sendenden Station natürlich auch die IP-Adresse des Routers bekannt sein, die entsprechende Hardwareadresse wird dann ebenfalls wieder über ARP ermittelt.

Damit muß die Konfiguration einer IP Station in einem Netzwerk mit Routern also folgende drei Angaben enthalten:

1. Netzwerkadresse des Hosts
2. Netzwerkmaske für diese Segment
3. Netzwerkadresse des nächsten Routers.

Nehmen wir ein Beispiel (siehe Bild), wie ein Datenpaket in einem Netzwerk mit vier Teilnetzen weitergeleitet wird.

Die Station A sendet ein Paket zur Station C. Dazu vergleicht sie zunächst die Netzwerkzieladresse mit ihrer eigenen Netzwerkadresse, um festzustellen, ob sich die Zielstation im gleichen Netzwerk befindet wie sie selbst oder in einem anderen Segment. In unserem Fall ist sie über zwei Router hinweg erreichbar.



Nachdem sich die Zielstation in einem anderen Netzwerksegment befindet, sendet die Station das Paket an den nächsten Router. Dessen Hardwareadresse wird über ARP ermittelt. Es wird also ein Frame mit folgenden Adreßfeldern erzeugt:

Ethernet Adressen:

Hardwarequelladresse: Adresse der Netzwerkkarte in der Sendestation

Hardwarezieladresse: Adresse der Netzwerkkarte im Router

IP Adressen:

Netzwerkquelladresse: 210.170.1.3

Netzwerkzieladresse: 210.170.3.3

Insgesamt wird also folgender Frame im Netzwerk 210.170.1.0 an den Router geschickt:

Ziel HW-Adresse	Quell HW-Adresse	Ziel IP-Adresse	Quell IP-Adresse	Daten
002011113333	002011112222	210.170.3.3	210.170.1.3	110110101110101

Das Paket wird zunächst an den nächsten Router gesandt. Dieser ersieht in seiner internen Routingtabelle, auf welchen Netzwerkstrang und zu welcher Zielstation das Paket weitergeleitet werden muß. Die internen Routingtabellen der drei Router sind in nachfolgender Tabelle zusammengestellt. In unserem Fall wird das Paket zum nächsten Router am Netzwerkstrang 210.170.2.0 weitergeleitet. Das Paket sieht nun wie folgt aus:

Ziel HW-Adresse	Quell HW-Adresse	Ziel IP-Adresse	Quell IP-Adresse	Daten
002011115555	002011114444	210.170.3.3	210.170.1.3	110110101110101

Dieser schaut wiederum in seine Routing Tabelle, stellt fest, daß die Empfangsstation im angrenzenden Segment 210.170.3.0 zu finden ist. Er führt einen ARP Request durch, um die Hardwareadresse zu ermitteln und stellt das Paket zu. Dies ist dann das zugestellte Paket:

Ziel HW-Adresse	Quell HW-Adresse	Ziel IP-Adresse	Quell IP-Adresse	Daten
002011117777	002011116666	210.170.3.3	210.170.1.3	110110101110101

Beim Routing werden also Pakete von einem Router zum jeweils nächsten weitergegeben, bis das Netzwerksegment mit der Zielstation erreicht ist. Dabei ändern sich jeweils die Hardwareadressen, die Netzwerkadressen bleiben konstant. Dieses Verfahren wird auch als Hop by Hop Routing bezeichnet. Jeder Router im Pfad analysiert die IP Adresse der Zielstation und legt aufgrund der Einträge in seiner Routingtabelle fest, an welche Zielstation das Paket gesandt wird. Im Netzwerksegment mit der Zielstation wird ebenfalls wieder über ARP die entsprechende Hardwareadresse ermittelt.

In unserem Beispiel sehen die Routingtabellen der drei Router etwa wie folgt aus:

Router 1:

Ziel Netzwerk	Next Hop	Interface	
210.170.1.0	210.170.1.1	210.170.1.1	lokal
210.170.2.0	210.170.2.1	210.170.2.1	lokal
210.170.3.0	210.170.2.2	210.170.2.1	remote
210.170.4.0	210.170.2.3	210.170.2.1	remote

Router 2:

Ziel Netzwerk	Next Hop	Interface	
210.170.1.0	210.170.2.1	210.170.2.2	remote
210.170.2.0	210.170.2.2	210.170.2.2	lokal
210.170.3.0	210.170.3.1	210.170.3.1	lokal
210.170.4.0	210.170.2.3	210.170.2.2	remote

Router 3:

Ziel Netzwerk	Next Hop	Interface	
210.170.1.0	210.170.2.1	210.170.2.3	remote
210.170.2.0	210.170.2.3	210.170.2.3	lokal
210.170.3.0	210.170.2.2	210.170.2.3	remote
210.170.4.0	210.170.4.1	210.170.4.1	lokal

5.5 Routingprotokolle

Router sind Geräte, die explizit beauftragt werden, ein Paket gemäß vorhandener Routingtabellen weiterzuleiten. Üblicherweise sind Router multiprotokollfähig, unterstützen also mehrere Protokollstacks und damit Routingprotokolle. Wie entstehen nun Routingtabellen? Man unterscheidet:

- Statisches Routing: Abgleich per Hand
- Dynamisches Routing: Verwendung von Routingprotokollen

Die wesentlichen Aufgaben von Routingprotokollen sind folgende:

- Kommunikation mit Nachbarroutern (gegenseitige Meldungen über Routenänderung)
- Erstellen und Verwalten von Routinginformation
- Festlegung der jeweils besten Route durchs Netz (üblicherweise durch Verwenden des Dijkstra Algorithmus unter Berücksichtigung von Pfadkosten)

Folgende Routing Protokolle finden sich u.A. heute in lokalen Netzen:

In IP-Umgebungen:

RIP I	Routing Information Protocol (Typ: Distanzvektorprotokoll, distance vector)
RIP II	Weiterentwicklung, im Gegensatz zu RIP I u.A: zusätzlich Übertragung von Subnet Information
OSPF	Open Shortest Path first (Typ: Verbindungsstatusprotokoll, Link State)

In (Novell-) IPX Umgebungen

RIP	Routing Information Protocol für IPX Umgebungen (es heißt gleich wie in der IP Welt, ist jedoch nicht kompatibel) (Typ: distance Vector)
NLSF	Netware Link Services Protokoll (Typ: Link State)

5.6 ICMP

ICMP wurde 1981 entwickelt (RFC792) und später in RFC 1256 erweitert

Grundsätzliche Aufgabe: Fehlerbehandlung und Diagnos.

Protokollheader:

IP Header	Type	Code	Checksum	
-----------	------	------	----------	--

Der ICMP Paket Typ legt die einzelne Aufgabe fest, die Codenummer dient zur weiteren Unterteilung von Nachrichten in Zusammenhang mit den einzelnen Aufgaben. Beispielsweise wird dem Sender bei einer Antwort „Destination Unreachable“ mitgeteilt, warum das Paket nicht gesendet werden konnte. Dies kann z.B. zur sog. verwendet werden MTU P

Die wesentlichen Nachrichtentypen sind in nachfolgender Tabelle zusammengefasst:

Aufgabe	Typ	Code		
Echo Request /Reply	8 /0			Ping Utility
Destination Unreachable	3	0	Netz nicht erreichbar	
		1	Rechner nicht erreichbar	
		2	Protokoll nicht erreichbar	
		3	Port nicht erreichbar	
		4	Fragmentierung erforderlich aber DF Bit gesetzt	Zur Feststellung der MTU (Max. Transfer Unit)
Source Quench	4			Führt zu Unterbrechung des Sendestroms für eine bestimmte Zeit $\approx$ IP Flußkontrolle
Redirect Message	5			Router weist einen Client daraufhin, dass eine bessere Route zu einem Ziel existiert.
Router Advertisement	9			Bekannmachung von Routern (Antwort auf Typ 10 Requests)
Router Solicitation	10			Routersuche (Default Gateway)
Time Exceeded	11			Gesendet von Routern, wenn TTL=0 $\approx$ Verwendung bei Traceroute bzw. Tracert
Parameter Problem	12			Falsche oder unbekannte Parameter im IP Header
Timestamp Request /Reply	13/14			Abfrage von Datum/Uhrzeit zur Laufzeitmessung

Tabelle: ICMP Nachrichtentypen und Codes

Weitere Nachrichten wie Information oder Address Mask Requests waren zur automatischen Adressvergabe gedacht, haben heute aber durch die Verwendung anderer Protokolle (DHCP) keine Bedeutung mehr.

Grundsätzliches Ziel von ICMP ist es Information über Netzwerke und deren Knoten an den Sender zurückzugeben. Insofern ist es auch die Basis für Tools wie Security Scanner, OS Fingerprinting Tools etc. zur Informationsbeschaffung sicherheitsrelevanter Informationen über Netzwerke und Knoten in Netzen.

6. Transportschicht

Findet der Datentransfer nur zwischen zwei Rechnern statt, die direkt im selben Segment des Netzes miteinander verbunden sind, läßt sich ein Teil dieser Aufgaben auch von Protokollen erledigen, die direkt auf ein Zugriffsprotokoll wie Token Ring oder Ethernet aufsetzen, erledigen. Ein Beispiel hierfür ist z.B. das aus dem SDLC hervorgegangene LLC Protokoll.

In einem größeren Netz mit netzwerkweiter Adressierung findet die Lösung dieser Aufgaben allerdings durch eigene Transportprotokolle statt, die man im Allgemeinen der Ebene 4 des OSI Referenzmodells bzw. der Host to Host Layer des DoD Modells zuordnet. Nachdem Quell- bzw. Zielrechner keine Informationen über die gesamte zwischen den Rechnern vorhandene Infrastruktur haben, müssen die verwendeten Transportprotokolle Mechanismen bereitstellen, dies aufzufangen.

Zum Datentransfer zwischen zwei entfernten Rechnern sind grundsätzlich folgende Aufgaben zu lösen:

- Segmentierung von Paketen
- Steuerung der Übertragungsmenge (Flußkontrolle)
- Überlastkontrolle des Netzes
- Kontrolle der Paketreihenfolge
- Senden von Bestätigungen, wenn Pakete empfangen wurden
- Fehlersicherung
- Bereitstellung verschiedener Übertragungskanäle
- Adressierung von Diensten

Die grundsätzliche Funktionsweise eines solchen Protokolls zeigt nachfolgendes Bild:

6.1 Übertragungsarten

Es gibt grundsätzlich verschiedene Ansätze, wie die korrekte Übertragung von Informationen in Netzen sichergestellt werden kann. Man unterscheidet:

Verbindungslose Übertragung:

Bei Einsatz verbindungsloser Protokolle schickt der Sender einer Nachricht entsprechende Pakete ins Netz und „hofft“ sozusagen darauf, dass die Pakete ankommen. Eine Bestätigung dafür existiert auf dieser Protokollebene nicht, dies wird dann den Protokollen der höheren Ebenen (d.h. den Dienstprotokollen) überlassen. Eine Analogie dazu ist in der realen Welt das Senden eines Briefes.

Bestätigt verbindungslose Übertragung:

Um überhaupt feststellen zu können, ob ein Paket angekommen ist, verwenden manche verbindungslose Protokolle Bestätigungsmeldungen (ACK: Acknowledgements). Eine Analogie dazu ist in der realen Welt das Senden eines Einschreibens mit Rückschein.

Verbindungsorientierte Übertragung:

Hier wird vor der eigentlichen Datenübertragung zunächst eine logische Verbindung aufgebaut innerhalb der dann die eigentliche Nutzinformation übertragen wird. Diese Verbindung wird überwacht, d.h. selbst wenn keine Nutzinformation zu senden ist, wird i.A. Kontrollinformation zwischen beteiligten Rechnern ausgetauscht. Nach dem Ende der Übertragung wird die Verbindung entsprechend wieder abgebaut. Das Existieren einer vorher vereinbarten Verbindung ist die Voraussetzung, dass erweiterte Mechanismen zur Verbindungskontrolle, wie z.B.

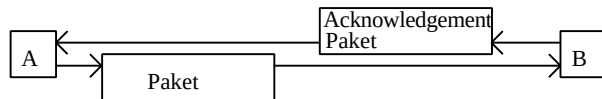
- Flusskontrolle, d.h. die Steuerung der Übertragungsmenge zwischen zwei Rechnern
- die Kontrolle der Reihenfolge von Paketen oder
- das Senden negativer Bestätigungen (NAK: Negative Acknowledgements)

bei Ausbleiben von Paketen) funktionieren können.

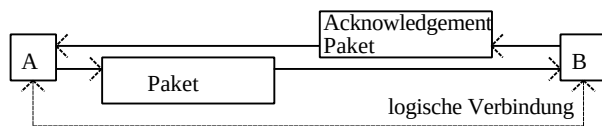
Da man davon ausgehen kann, dass die Gefahr eines Verlusts von Paketen in LANs oder heutigen WANs gering ist, verwenden die meisten Protokollstacks verbindungsorientierte Übertragung erst auf der Transportschicht oder innerhalb der Dienstprotokolle selbst.



Unacknowledged - Connectionless Service
Verbindungslose Übertragung



Acknowledged Connectionless Service



Connection Oriented Service (Verbindungsorientierte Übertragung)

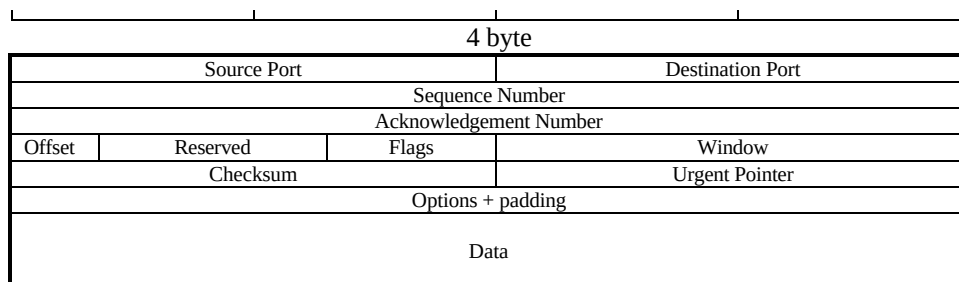
6.2 Transportprotokolle im Internet

In der Internet Protokollwelt stehen zwei alternative Transportprotokolle zur Verfügung. Eines davon, TCP (Transport Control Protocol) ist verbindungsorientiert (connection oriented), das andere ist das verbindungslose (connectionless) UDP (User Datagramm Protocol).

6.2.1 TCP (Transport Control Protocol)

Aufgaben, die durch TCP realisiert werden, sind:

- Festlegung eines Übertragungsfensters (Window-Feld), um die Übertragungsmenge festzulegen
- Segmentierung und Nummerierung von Paketen (Sequence Number)
- Bestätigungen (Acknowledgement Number)
- Adressierung von Diensten (über Source Port, Destination Port)

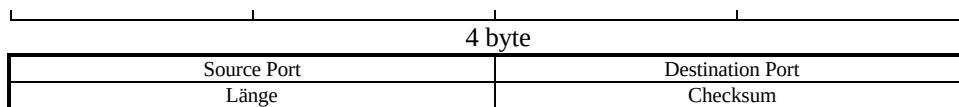


6.2.2 UDP (User Datagram Protocol)

Das User Datagram Protocol übernimmt lediglich die Aufgabe der Adressierung der Upper Layer Prozesse. Damit kann UDP als einfacher Multiplexer für verschiedene Übertragungskanäle angesehen werden.

Aufgaben der Flußkontrolle sowie Transportkontrolle müssen dann von den oberen Schichten übernommen werden. Entsprechend einfach sieht der Frame auch aus.

UDP-Frame:



Beispiele für Upper Layer Protokolle, die auf einem der beiden Transportprotokolle aufsetzen, sind:

UDP: NFS, TFTP, SNMP

TCP: FTP, Telnet, SMTP

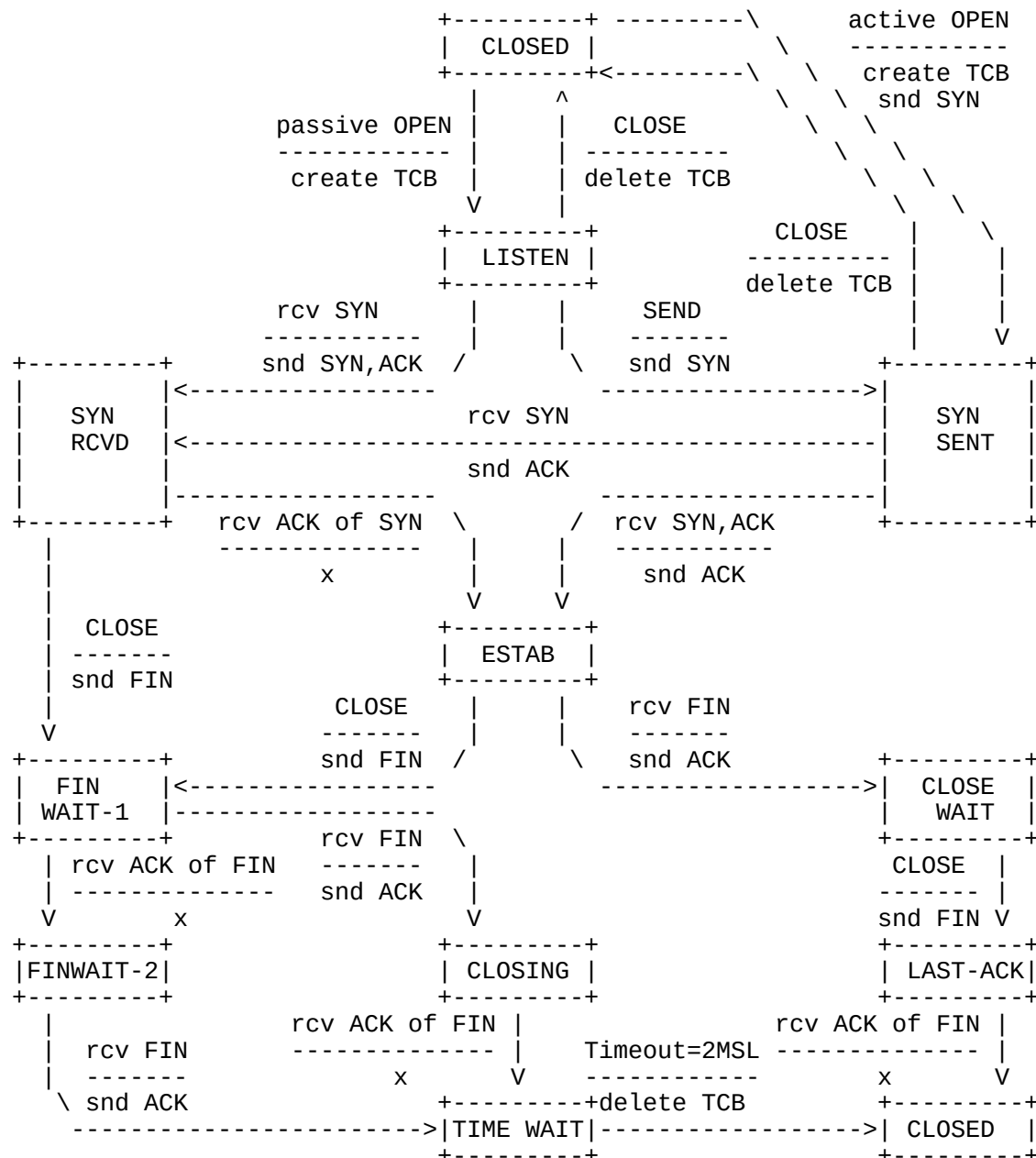
6.2.3 Sockets

Um zu verstehen, wie in der Internet Protokollwelt Verbindungen aufgebaut werden, muß zunächst der Begriff Socket erklärt werden.

Sockets stellen eine Programmierschnittstelle zu TCP bzw. UDP dar (der Begriff und die Implementation kommt aus dem Berkley UNIX). Sockets funktionieren ähnlich wie ein Dateihandling. So liefert eine Funktion zum Öffnen einer Datei als Rückgabewert einen sog. Dateihandle. Weitere Zugriffe auf diese Datei, z.B. zum Schreiben oder Lesen, finden dann über dieses Dateihandle statt. Ähnlich funktioniert der Zugriff auf TCP. Hier ist der Rückgabewert beim Öffnen einer Verbindung ein Socket, über den dann der Datenaustausch über normale Schreib/Lese Funktionen abläuft. Der Socket ergibt sich über die Kombination der IP-Adresse mit einer Port Nummer, also z.B. 192.168.5.31:1677

6.3 TCP Zustandsmodell

Um TCP zu beschreiben, wird in der entsprechenden RFC 793 das nachfolgend dargestellte Zustandsmodell verwendet. Der obere Teil zeigt den Aufbau, der untere den Abbau von Verbindungen. Die Übergangspfeile zeigen in der Syntax xxx/yyy das in einem Zustand jeweils empfangene/gesendete Paket. Um das Bild insgesamt zu verstehen, muß man wissen, dass beide Rechner beim Verbindungsaufbau separat betrachtet werden.



TCP Zustandsdiagramm (RFC 793 Figure 6)

6.3.1 Verbindungsaufbau

Eine Verbindung kommt nun wie folgt zustande:

Ein Dienst wartet auf einem Server mit einer bestimmten IP Adresse unter einer bestimmten Port Nummer auf eine Verbindung. Er geht dazu in den Status LISTEN. Dies wird als ein sogenannter „Listen Socket“ bezeichnet.

Der Client Prozeß macht nun Service Anfragen über Aufrufe zu diesem Socket. Dies geschieht wie folgt:

1. Der Client macht einen Request für eine Verbindung, d.h. er sendet ein SYN Paket mit den gewünschten Übertragungsparametern. Dadurch wechselt er in den Zustand SYN_SENT.
2. Der Server erhält den SYN Request, wechselt in den Zustand SYN_RCVD und beantwortet den Request (ACK). Mit dem Erhalt des ACKs steht nun die Verbindung aus Sicht des Clients eigentlich bereits. Gleichzeitig mit dem Senden des ACKs sendet der Server ebenfalls ein SYN um die Verbindung von seiner Seite auf mit seinerseits gewünschten Parametern aufzubauen (er sendet also insgesamt SYN,ACK).
3. Wenn der Client das SYN,ACK vom Server erhält, bestätigt den Erhalt mit einem ACK und wechselt in den Zustand ESTABLISHED. Der Server erhält das ACK und wechselt ebenfalls in den Zustand ESTABLISHED. Damit steht nun auch die Verbindung aus Sicht des Servers.

Der ganze Prozeß wird als 3-way-handshake bezeichnet, verwendet wird der kleinste gemeinsame Nenner bezüglich der Übertragungsparameter. Der wesentliche Grund für die Verwendung des Verbindungsaufbaus in 3 Schritten ist es, evtl vorhandene alte, doppelte Verbindungsaufbauversuche entkoppeln zu können. Nachfolgend ist der Aufbau von Verbindungen in verschiedenen Situationen gezeigt:

TCP A		TCP B
1. CLOSED		LISTEN
2. SYN-SENT	--> <SEQ=100><CTL=SYN>	--> SYN-RECEIVED
3. ESTABLISHED	<-- <SEQ=300><ACK=101><CTL=SYN, ACK>	<-- SYN-RECEIVED
4. ESTABLISHED	--> <SEQ=101><ACK=301><CTL=ACK>	--> ESTABLISHED
4. ESTABLISHED	--> <SEQ=101><ACK=301><CTL=ACK><DATA>	--> ESTABLISHED

Basic 3-Way Handshake for Connection Synchronization (RFC 793 Figure 7)

Die Acknowledgement Number ACK zeigt sozusagen jeweils auf die Sequenznummer des Segments, das als nächstes von der Gegenseite erwartet wird.

TCP A		TCP B
1. CLOSED		CLOSED
2. SYN-SENT	--> <SEQ=100><CTL=SYN>	...
3. SYN-RECEIVED	<-- <SEQ=300><CTL=SYN>	<-- SYN-SENT
4. ...	... <SEQ=100><CTL=SYN>	--> SYN-RECEIVED
5. SYN-RECEIVED	--> <SEQ=100><ACK=301><CTL=SYN, ACK>	...
6. ESTABLISHED	<-- <SEQ=300><ACK=101><CTL=SYN, ACK>	<-- SYN-RECEIVED
7. ...	... <SEQ=101><ACK=301><CTL=ACK>	--> ESTABLISHED

Simultaneous Connection Synchronization (RFC793 Figure 8)

TCP A		TCP B
1. CLOSED		LISTEN
2. SYN-SENT	--> <SEQ=100><CTL=SYN>	...
3. (duplicate)	... <SEQ=90><CTL=SYN>	--> SYN-RECEIVED
4. SYN-SENT	<-- <SEQ=300><ACK=91><CTL=SYN, ACK>	<-- SYN-RECEIVED
5. SYN-SENT	--> <SEQ=91><CTL=RST>	--> LISTEN
6. ...	... <SEQ=100><CTL=SYN>	--> SYN-RECEIVED
7. SYN-SENT	<-- <SEQ=400><ACK=101><CTL=SYN, ACK>	<-- SYN-RECEIVED
8. ESTABLISHED	--> <SEQ=101><ACK=401><CTL=ACK>	--> ESTABLISHED

Recovery from Old Duplicate SYN (RFC 793 Figure 9)

6.3.2 Halboffene Verbindungen

Eine bestehende TCP Verbindung ist dann halboffen,

- wenn einer der beiden TCP-Stacks geschlossen ist (CLOSED)
- die Verbindung auf seiner Seite abgebaut hat, ohne die Gegenseite zu benachrichtigen oder
- wenn die beiden Enden der Verbindung nicht mehr richtig synchronisiert sind, weil z.B. einer der beiden Rechner abgestürzt ist.

Folgendes Beispiel zeigt denn Fall eines Crashes von A's TCP. Bei einem Neustart des TCP wird A versuchen die Verbindung erneut zu öffnen, während B denkt, sie ist offen.

TCP A		TCP B
1. (CRASH)		(send 300, receive 100)
2. CLOSED		ESTABLISHED
3. SYN-SENT --> <SEQ=400><CTL=SYN>		--> (??)
4. (!!) <-- <SEQ=300><ACK=100><CTL=ACK>		<-- ESTABLISHED
5. SYN-SENT --> <SEQ=100><CTL=RST>		--> (Abort!!)
6. SYN-SENT		CLOSED
7. SYN-SENT --> <SEQ=400><CTL=SYN>		-->

Half-Open Connection Discovery (RFC 793 Figure 10)

B antwortet auf das SYN mit einem ACK um A mitzuteilen welche Sequenznummer B als nächstes erwartet (ACK=100). A stellt dabei fest, dass es keine Bestätigung auf ursprünglich gesendete Pakete bekommt und sendet einen Reset (RST). Danach wird der 3-Wege Handshake neu eingeleitet.

Das nächste Bild zeigt einen Crash von TCP A. A erhält ein Paket von B, kennt aber die dazugehörige Verbindung nicht und sendet einen RST. B bricht bei Empfang des RST Pakets die Verbindung ab.

TCP A		TCP B
1. (CRASH)		(send 300, receive 100)
2. (??) <-- <SEQ=300><ACK=100><DATA=10><CTL=ACK>		<-- ESTABLISHED
3. --> <SEQ=100><CTL=RST>		--> (ABORT!!)

Active Side Causes Half-Open Connection Discovery (RFC 793 Figure 11)

Im nachfolgenden Bild befinden sich beide Kommunikationspartner im LISTEN Zustand. Rechner B erhält ein altes, doppeltes SYN Segment und antwortet mit einem SYN,ACK. Dies führt dazu, dass Rechner A eine RST schickt.

TCP A		TCP B
1. LISTEN		LISTEN
2. ... <SEQ=Z><CTL=SYN>		--> SYN-RECEIVED
3. (??) <-- <SEQ=X><ACK=Z+1><CTL=SYN, ACK>		<-- SYN-RECEIVED
4. --> <SEQ=Z+1><CTL=RST>		--> (return to LISTEN!)
5. LISTEN		LISTEN

Old Duplicate SYN Initiates a Reset on two Passive Sockets (RFC 793 Figure 12)

Generell gilt: Ein RST muß immer dann gesendet werden, wenn ein Segment ankommt, das offensichtlich nicht zur aktuellen Verbindung passt. Im Zweifelsfall darf kein RST gesendet werden.

Man kann drei Fälle unterscheiden. RST wird dann gesendet wenn:

1. keine Verbindung besteht (Socket CLOSED). Dies gilt speziell für SYN Pakete.
2. die Verbindung nicht synchronisiert ist (d.h. der dazugehörige TCP sich nicht im Zustand LISTEN, SYN-SENT oder SYN-RECEIVED befindet) und das ankommende Paket beinhaltet ein falsches ACK.

Es darf kein RST geschickt werden, wenn

3. die Verbindung synchronisiert ist (ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME-WAIT). Hier darf der Empfänger eines Segments nur ein leeres ACK Segment mit der aktuellen SEQ Nummer und ein ACK mit der nächsten zu erwartenden SEQ Nummer der Gegenseite senden. Die Verbindung bleibt bestehen.

Der Empfang eines RST wird akzeptiert, wenn

- sich der Empfänger im SYN-SENT Status befindet (d.h. der RST als Ergebnis eines SYN Requests geschickt wurde), wenn das ACK Feld den SYN bestätigt
- ansonsten die Sequence Number SEQ im erwarteten Bereich liegt (Sendefenster: siehe später)

Nach Überprüfung des RST wechselt der Empfänger in den Zustand CLOSED. Befindet sich der Empfänger im Zustand LISTEN oder SYN RECEIVED, nachdem er vorher in LISTEN war, geht er in den Zustand LISTEN.

6.3.3 Verbindungsabbau

Wenn eine Seite keine Daten mehr zu senden hat, schließt sie die Verbindung über eine CLOSE Sequenz. Genaugenommen müssen dazu beide Seiten die Verbindung abbauen. Dazu müssen drei Fälle unterschieden werden:

1. Der Benutzer weist seinen TCP an, die Verbindung zu schließen.

Der TCP geht in den Status FIN-WAIT-1. In diesem Zustand kann TCP noch Segmente empfangen, aber nicht mehr senden. Alle vorherigen Segmente wie auch das FIN werden solange gesendet, bis sie von der Gegenseite bestätigt werden. Die Gegenseite bestätigt das FIN (mit ACK). Die Gegenseite schließt die Verbindung nun ebenfalls und sendet dazu seinerseits ein FIN, das von der lokalen Seite mit ACK bestätigt werden muß.

2. Die Gegenseite schliesst die Verbindung durch Senden eines FIN Segments

Beim Erhalt eines FIN teilt TCP der lokalen Anwendung mit, dass die Verbindung geschlossen werden soll. Die lokale Anwendung weist TCP an die Verbindung seinerseits zu schließen (CLOSE), es werden alle restlichen Daten und danach ein FIN gesendet, das von der Gegenseite mit ACK beantwortet wird. Kommt kein ACK innerhalb eines Timeouts zurück, wird die Verbindung trotzdem geschlossen (CLOSED).

3. Beide Seiten schließen die Verbindung gleichzeitig

Hier senden beide Seiten ein FIN. Nachdem beide alle vorhergehenden Segmente abgearbeitet haben, bestätigen sie sich gegenseitig den Verbindungsabbau mit ACKs und schließen die Verbindung.

Eine normaler und ein gleichzeitiger Verbindungsabbau sind nachfolgend gezeigt:

TCP A			TCP B		
1.	ESTABLISHED			ESTABLISHED	
2.	(Close)				
	FIN-WAIT-1	--> <SEQ=100><ACK=300><CTL=FIN, ACK>	-->	CLOSE-WAIT	
3.	FIN-WAIT-2	<-- <SEQ=300><ACK=101><CTL=ACK>	<--	CLOSE-WAIT	
4.				(Close)	
	TIME-WAIT	<-- <SEQ=300><ACK=101><CTL=FIN, ACK>	<--	LAST-ACK	
5.	TIME-WAIT	--> <SEQ=101><ACK=301><CTL=ACK>	-->	CLOSED	
6.	(2 MSL)				
	CLOSED				

Normal Close Sequence (RFC 793 Figure 13)

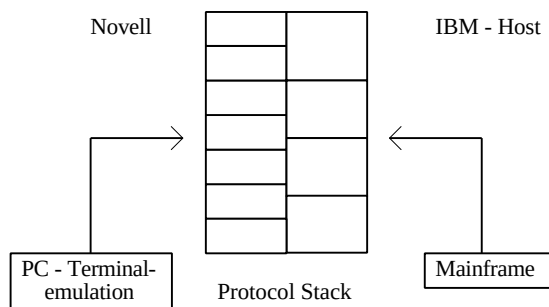
TCP A			TCP B		
1.	ESTABLISHED			ESTABLISHED	
2.	(Close)			(Close)	
	FIN-WAIT-1	--> <SEQ=100><ACK=300><CTL=FIN, ACK>	...	FIN-WAIT-1	
		<-- <SEQ=300><ACK=100><CTL=FIN, ACK>	<--		
		... <SEQ=100><ACK=300><CTL=FIN, ACK>	-->		
3.	CLOSING	--> <SEQ=101><ACK=301><CTL=ACK>	...	CLOSING	
		<-- <SEQ=301><ACK=101><CTL=ACK>	<--		
		... <SEQ=101><ACK=301><CTL=ACK>	-->		
4.	TIME-WAIT			TIME-WAIT	
	(2 MSL)			(2 MSL)	
	CLOSED			CLOSED	

Simultaneous Close Sequence (RFC 793 Figure 14)

6.4 Gateways zu anderen Protokollstacks

Gateways dienen zur Protokollumsetzung zwischen zwei Netzen. Dies kann eine einfache Protokollumsetzung zwischen zwei gleichen Protokollstacks zum Zweck der Zugriffssicherung sein (z.B. eine IP/IP Gateway in einem Firewall), eine Protokollumsetzung bestimmter einzelner Protokolle (z.B. IPX/IP-Gateway) bis hin zu einer Umsetzung eines Protokollstacks (z.B. Umsetzung der Protokolle der OSI-Ebenen 1-4 bei einem Gateway in die IBM SNA-Welt).

Nachfolgendes Bild zeigt ein SNA Gateway. Auf der LAN Seite wird z.B. TCP/IP eingesetzt, auf der Hostseite SNA. Auf dem Client läuft eine Programm zur Emulation eines SNA Terminals, das auf TCP/IP aufsetzt. Damit findet die Protokollumsetzung auf den Ebenen 1-5 statt, darüber werden die Tastatureingaben bzw. Bildschirmausgaben vom Großrechner zur Terminalemulation geschickt, d.h. die Umsetzung dieser Daten findet dann für jeden Client individuell auf dem Clientrechner statt.



Weitere Beispiele sind Protokollgateways, wie z.B. IPX/IP Gateways oder IP/IP Gateways, wie sie z.B. in Firewalls verwendet werden.

7. Firewalls und Netzwerksicherheit

Gerade beim Anschluß ans Internet müssen Schutzmechanismen gegen unbefugtes Eindringen in ein firmeninternes Netz von außen vorgesehen werden. Grundsätzlich gibt es hier zwei Möglichkeiten:

- Sichern jedes firmeninternen Rechners (der Aufwand steigt natürlich mit der Zahl der Rechner, so daß dies kaum mehr sinnvoll realisierbar ist.
- Zentralisierung der Sicherheitsproblematik eines Netzzugriffs von außen auf einen einzigen Punkt. Dieser Punkt wird dann speziell abgesichert (sog. Firewall)

7.1 Angriffsmöglichkeiten

Die folgende Liste zeigt eine (nicht vollständige) Auswahl an Angriffsmöglichkeiten für Netze:

- Ausspionieren von Daten auf Systemen durch Einbruch in diese Systeme
- Ausspionieren bei der Übertragung zwischen Systemen (Mitlesen)
- Manipulation von Daten auf Systemen (z.B. durch Hacken, Viren, Java Applets, Active X Controls), im Prinzip werden hier Programme durch irgendeinen Mechanismus auf einen Client übertragen und dort ausgeführt.
- Manipulation von Verbindungen (Übernehmen einer Verbindung mit Hilfe gefälschter Adressen)
- Verhindern der normalen Funktion eines Rechners (Denial of Service Attack). Beispiele hierfür sind: Ping of Death oder Syn Flooding

Was sind Möglichkeiten, die ein Hacker nutzen kann, um in ein Netzwerk einzubrechen:

- Paßwortattacken (Hacken)
- Social Engineering
- Fehler und Hintertüren in Programmen oder Protokollen
- Informationslecks in Firmen
- Versagen der Authentisierungsmechanismen

7.2 Funktion und Komponenten von Firewalls

7.2.1 Paketfilter

Eine sehr einfache Möglichkeit ist, Kommunikationsbeziehungen zu filtern. Hier können ICMP, IP, UDP bzw. TCP Filter implementiert werden. Damit ist diese Firewall lediglich ein Router, der abhängig von bestimmten Adressen routen kann.

IP Filter: Hier werden im wesentlichen Absender und Ziel-IP-Adressen gefiltert. Die Firewall routet dann nur Pakete von und zu bestimmten Ziel bzw. Quelladressen.

Hierzu wird im Router/Firewall eine Adressliste mit diesen Adressen geführt. Es gibt bei den meisten Produkten die Möglichkeit, entweder aufzuführen, zu welchen Adressen Kommunikationsbeziehungen erlaubt sind (Pass List) oder verboten sind (No-Pass List). Normalerweise ist die gängige Vorgehensweise, daß zunächst jeder Paketverkehr über die Firewall verboten wird, und dann notwendige Ausnahmen von dieser Grundregel gemacht werden.

Die meisten Paketfilter sind nicht nur in der Lage, nach IP-Adressen zu filtern, sondern dies auch noch abhängig von Quell und Zielpports durchzuführen. Problematisch ist der Einsatz speziell von Port-Filtern dann, wenn Protokolle mit mehreren logischen Verbindungen wie z.B. FTP, bestimmte Protokolle im Multimediabereich gefiltert werden sollen. So müssen z.B. für FTP Übertragungen ganze Bereiche möglicher Portnummern geöffnet werden, um Kommunikation zu ermöglichen. Diese Ports sind dann natürlich wieder als Angriffspunkte offen.

Man unterscheidet statische und dynamische Paketfilterung. Bei dynamischen Paketfiltern wird der Status jeder Verbindung mit in einer Tabelle gespeichert, die dann Grundlage zur Filterung ist. Antwortpakete, die zu bestimmten ausgehenden Verbindungen gehören, werden durchgelassen, auch wenn entsprechende Filterregeln das für andere eingehende Pakete verhindern würden. Die Filterregeln werden dabei dynamisch angepaßt, was die Konfiguration solcher Problemstellungen erheblich erleichtert. Diese Art der Paketfilterung wird häufig als Stateful Inspection oder Stateful Filtering bezeichnet. Sie wird vor allem für Protokolle eingesetzt, die mehrere Verbindungen benötigen (z.B. FTP).

7.2.2 IP Adressumsetzung (NAT)

Hier werden den Rechnern innerhalb eines Intranets inoffizielle Adressen gegeben, diese werden dann bei Kommunikation ins Internet dynamisch auf die gültige Adresse des Routers/Firewalls gesetzt. Aus dem Internet betrachtet beginnt und endet damit jede Kommunikation mit dem Router, der Router selbst sorgt dann für die „Verlängerung“ der Verbindung zum jeweiligen Client im Netz.

Als inoffizielle Adressen können natürlich beliebige Adressen gewählt werden, es empfiehlt sich jedoch aus Sicherheitsgründen einen der reservierten Adressbereiche zu verwenden.

Als reservierte Adressbereiche stehen folgende Adressen zur Verfügung:

192.168.0.0	...	192.168.255.255
172.16.0.0	...	172.31.255.255
10.0.0.0	...	10.255.255.255

Man unterscheidet statisches und dynamisches NAT:

Dynamisches NAT

Beim dynamischen NAT werden private Adressen eines ganzen Subnetzes in eine offizielle Adresse umgesetzt (dies ist die öffentliche Seite des NAT Routers, d.h., die Seite, von der aus die Verbindung ins Internet besteht). Der Vorteil ist, daß für Kommunikationsbeziehungen, die aus dem Intranet heraus gestartet werden, die ursprüngliche Quelladresse einer Station im Internet nicht sichtbar wird. Aus der Sicht des Clients ist die Verbindung transparent ins Internet, d.h. dem Client ist die Adressumsetzung nicht bewußt, er denkt, er hängt direkt am Internet. Ob die internen Rechner von außen erreichbar sind, wenn die internen Adressen außen bekannt sind, ist abhängig von der Konfiguration der jeweiligen Firewall.

Statisches NAT

Statisches NAT kann dazu verwendet werden, einzelne interne Rechner mit bestimmten internen IP-Adressen, auf entsprechende offizielle IP Adressen der Firewall umzusetzen. Hier erhält jeder interne Rechner eine entsprechende

offizielle Adresse auf der öffentlichen Seite des NAT Routers. Damit ist der interne Rechner aus dem Internet über diese offizielle IP-Adresse erreichbar.

Wird statisches und dynamisches NAT parallel eingesetzt, muß die Firewall auf der öffentlichen Seite mehrere IP Adressen parallel erhalten (Multihoming), eine für die dynamische Komponente und je eine für jede statische Verbindung ins interne Netz.

Einen absoluten Schutz vor dem Eindringen von außen bietet diese Lösung nicht. Sind die im internen Netz verwendeten Adressen im außen bekannt, könnten Sie auch erreicht werden (zumindest wenn es sich nicht um die oben angegebenen Adressen aus den privaten Adressbereichen handelt, die im Internet nicht geroutet werden). Aus diesem Grunde sollte durch entsprechende Filterung der Routingprotokolle zusätzlich dafür gesorgt werden, daß der NAT Router die Routing Information über die internen Netze nicht nach außen weitergibt.

7.2.3 TCP/UDP Relay

Hier findet auf der Ebene 4 eine Portumsetzung innerhalb von Kommunikationsbeziehungen statt. Das Gateway wartet auf einem vordefinierten Port auf eine Verbindung, diese wird dann über einen anderen Port weitergeleitet.

Logisch wird dadurch eine Verbindung zwischen Client und Firewall und davon getrennt eine zweite Verbindung von der Firewall zum entsprechenden Host im Internet aufgebaut.

Da hier kein Routing verwendet wird, das IP-Paket also nicht weitergeleitet wird sondern tatsächlich neu aufgebaut wird, spricht man in diesem Zusammenhang auch von einem sogenannten Dual Homed Gateway, also einem Gateway mit zwei physikalisch getrennten Netzwerkverbindungen.

Die Anwendung dieser Gateways ist durch die feste Zuordnung von Zielsystemen mit bestimmten Zielpports begrenzt. Dies kann z.B. bei Nutzung verschiedener Server problematisch werden, die denselben Dienst über unterschiedliche Ports anbieten.

Ähnlich funktionieren auch Protokollumsetzer wie IP/IP Gateways oder auch IPX/IP Gateways. Letztere wickeln dann eben die Verbindung innerhalb des Intranets bis zum Gateway über Novell's IPX Protokoll ab.

7.2.4 Application Layer Gateway (Proxy)

Hier findet eine vollständige Trennung der Kommunikation innerhalb eines Intranets mit dem Internet statt. Die Firewall wird ist damit ein Dual Homed Gateway mit einer Überwachung der Verbindungen auf der Applikationsschicht. Dies wird im allgemeinen mit einer IP Adreßumsetzung (NAT) kombiniert (s.o.).

Möchte ein Benutzer z.B. eine bestimmte Seite auf einem WWW Server im Internet lesen, wird zum Proxy Server eine Verbindung aufgebaut. Der Proxyserver baut nun seinerseits anstelle des Clients die Verbindung zur gewünschten Web Site auf und liest sie. Von dort wird sie dann dem Client zur Verfügung gestellt. Der Proxy Server besitzt einen Cache der für manche Dienstprotokolle die am häufigsten gelesenen Seiten zwischenspeichert. Bei einem erneuten Zugriff kann dann die Nachfrage direkt aus dem Cache befriedigt werden. (z.B. HTTP Proxy Cache).

Protokolle, für die häufig Proxy Server eingesetzt werden, sind:

- HTTP
- SMTP
- FTP
- NNTP

Die Firewall muß natürlich die entsprechenden Protokolle verstehen, um protokollspezifisch eingreifen zu können. Dies erhöht zunächst den Aufwand solcher Gateways ganz gewaltig. Zudem besteht hier das Risiko, daß Angriffe, die das Gateway nicht selbst betreffen, auch nicht mitprotokolliert werden können.

Proxy Server können auf verschiedene Weisen eingesetzt werden:

Proxy Cache

Dies ist die Standardkonfiguration

Hierarchischer Proxy Cache

Hier werden mehrere Proxy Server zu einer Hierarchie zusammengestellt, daß heißt, wenn ein Proxy Server eine Seite benötigt, holt er Sie wiederum aus einem übergeordneten Cache. Zur Kommunikation der Caches untereinander wird hierbei das Internet Cache Protokoll (ICP) eingesetzt.

Transparent Proxy

Auf Client Seite (im Netscape oder Internet Explorer muß normalerweise eingestellt werden, daß ein Proxy verwendet wird). Ein Transparenter Proxy steuert dies automatisch (transparent für den Benutzer)

Revers Proxy (Web Accelerator)

Der wesentliche Aspekt ist bei Proxy und Proxy Cache Servern natürlich die Performance des entsprechenden Servers, des verwendeten Betriebssystems sowie des Proxy Servers selbst. So gibt es inzwischen allerdings so schnelle Proxy Server (z.B. Novell Bordermanager), die als Web Accelerator dienen können. Hier wird einem Webserver, auf dem aus dem Internet viele Anfragen eintreffen, einfach ein Web Accelerator vorgeschaltet, der die am häufigsten gewünschten Seiten aus dem Cache ohne der Notwendigkeit des Plattenzugriffs liefern kann.

7.3 Firewallarchitekturen

Grundsätzlich ist es möglich Firewalls als fertige Produkte zu kaufen, oder aber sie selbst mit Hilfe von Standardhardware und Firewall-Softwarekomponenten selbst aufzubauen. Hierbei gibt es verschiedene Möglichkeiten, wie eine Anordnung von Komponenten zu einer Firewall erfolgen kann.

Grundsätzlich muß beim Aufbau einer Firewall bedacht werden daß sich folgende drei Komponenten in etwa die Waage halten müssen:

- Aufwand
- Kontrolle
- Performance

Grundsatz ist: *Die sicherste Firewall ist die, die keinen Verkehr mit der Außenwelt zuläßt!*

Im folgenden werden die gängigsten Konzepte kurz vorgestellt:

Dual Homed Host

Screened Host Architektur

Screened Subnet Architektur

Aufgaben:

Ext: Verhindern von Adressfälschungen

DMZ: Hier würde eine Paketüberwachung (Snooping, Herumschnüffeln) nur Informationen von und zum Bastion Host liefern, aber keine Information über das interne Netz

Int. Dieser Router hat die Aufgabe , das Intranet vor der DMZ und vorm Internet zu schützen

Problematik: Mehrere Router liefern natürlich auch mehrere Angriffspunkte

Im folgenden werden einige Variationen gegenübergestellt:

Firewalls und Netzwerksicherheit

Wie können jetzt die gängigsten Protokolle am besten gefiltert werden:

DNS:

DNS benötigt Port 53 für die Kommunikation mit anderen DNS Servern, hier ist ein Portfilter einsetzbar. Allerdings muß hier unterschieden werden, ob ein DNS Request eines Clients oder eines Servers erfolgt.

Telnet:

Telnet ist ein sehr einfaches Protokoll, innerhalb einer Telnetverbindung werden z.B. Paßwörter unverschlüsselt übertragen. Eingehende TelnetSessions sollte man über IP-und Portfilterung soweit wie möglich beschränken, zusätzlich sollten nur einmal benutzbare Paßwörter verwendet werden.

Ausgehende Telnet Sessions sollten über Paketfilter und/oder Proxy Server abgewickelt werden

FTP:

Bei klassischem FTP kann eigentlich fast nur ein Proxy Server verwendet werden, bei passive Mode FTP ist ein einfacher Paketfilter ausreichend, wobei hier auch wieder der gesamte eingehende Datenverkehr über einen gesicherten Host zu übertragen sind.

HTTP:

Für ausgehendes HTTP empfiehlt sich der Einsatz von Proxy Servern. Problematisch ist dann aber die Verwendung dynamischer HTML Seiten und CGI Scripts. Zudem werden Informationen wie auch Paßworte unverschlüsselt übertragen. Abhilfe bietet hier das SSL (Secure Socket Layer) Protokoll.

Für eingehendes HTTP empfiehlt sich eine genaue Zugriffsüberwachung sowie der Einsatz spezieller Programme zum Upload auf einen internen Server, wenn dies überhaupt notwendig ist.

SMTP/NNTP:

SMTP ist als Store and Forward Protokoll sehr gut für den Einsatz eines Proxy Servers geeignet. Zudem ist es sinnvoll alle Verbindungen per Paketfilter auf einen bestimmten Bastion Host zu lenken. Ähnliches gilt für NNTP.

7.4 Sicherheit von Netzen

Grundsätzlich sind Firewalls nur eine Möglichkeit, Netze zu sichern. Diese ist auch nur für Angriffe aus dem Internet erfolgreich. Aus diesem Grund gibt es Firmen, die auch zwischen Abteilungen Firewalls einsetzen.

Zudem müssen neben Firewalls aber auch andere Mechanismen zum Schutz vorhanden sein. Folgende Mechanismen geben eine Übersicht:

- Physikalischer Zugriffsschutz auf bestimmte Rechner (insbesondere Server)
- Geheimhaltung, Integrität und Verfügbarkeit
- Identifikation und Beglaubigung
- Überwachung des Netzes auf Eindringlinge
- Maßnahmen zur Geheimhaltung von Paßworten (physikalisch und elektronisch)
- Verwenden von Verschlüsselung (Beim Netzzugriff und bei der Informationsübertragung durch ungesicherte Netze)
- Überwachung der Netzwerkaktivität (Auditing)
- Zugriffskontrolle auf Ressourcen

Die gesicherte Übertragung von Information wird mit der zunehmenden Nutzung des Internets immer wichtiger. Anwendungen sind z.B. ein Aufbau von gesicherten verschlüsselten Kanälen durch das Internet zwischen verschiedenen Standorten einer Firma (Virtual Private Networks VPN) oder auch die Identifikation von Benutzern im Internet für Anwendungen wie Electronic Banking oder E-Commerce. Man unterscheidet im Allgemeinen folgende Anwendungsgebiete bei der Herstellung einer gewissen Netzwerksicherheit:

- Verschlüsselung von Nachrichten
- Identification (elektronische Signatur): Hier wird durch Übertragung von bestimmten Prüfsummen bewiesen, daß eine Nachricht von einem bestimmten Benutzer kommt
- Authentication (Beglaubigung): Hier sendet ein Benutzer private Information, die beweist, daß er der ist, für den er sich ausgibt.

7.5 Verschlüsselung

Es gibt bei der Verschlüsselung zwei grundsätzlich unterschiedliche Ansätze:

- symmetrische Verschlüsselung (Secret Key Encryption) und
- asymmetrische Verschlüsselung (Public Key Encryption).

7.5.1 Symmetrische Verschlüsselung

Das wesentliche Kennzeichen ist, daß zum Verschlüsseln wie zum Entschlüsseln derselbe Schlüssel verwendet wird. Dieser muß vor einer Kommunikationsbeziehung zwischen beiden Kommunikationspartnern über einen getrennten Übertragungskanal ausgetauscht werden.

Zur Verschlüsselung gibt es u.A. folgende Verfahren:

- Verschiebung von Alphabeten
- Ersatz von Zeichen (Substitution)
- Vertauschen von Bits (Permutation)
- Algebraische Methoden

Eines der bekanntesten Verfahren ist der DES Algorithmus (Data Encryption Standard), der bei der ISO als DEA-1 bezeichnet wird. Hier wird ein Text in Blöcke von 64bit-Binärworten eingeteilt und blockweise mit einem 56

byte Schlüssel verschlüsselt. Dazu wird auf den Klartext 16mal eine Kombination von Substitution und Permutation angewandt.

Der Algorithmus wurde später weiterentwickelt zu 3-fach DES, hier wird mit zwei Schlüsseln gearbeitet. Der Text wird mit Schlüssel 1 verschlüsselt, mit Schlüssel 2 entschlüsselt und wieder mit Schlüssel 1 verschlüsselt.

RC2 und RC4 sind Verschlüsselungsalgorithmen mit variabler Schlüssellänge. Die Algorithmen sind in einer Reihe kommerzieller Produkte verfügbar (Netscape, Novell Netware, Lotus Oracle ...). Die Verschlüsselungsgeschwindigkeit hängt nicht von der Schlüssellänge ab. Üblicherweise wird eine Schlüssellänge von 128bit verwendet, für den Export wird die Schlüssellänge auf 40bit beschränkt. Damit sind nur noch 2^{40} (=ca $1 \cdot 10^{12}$) verschiedene Schlüssel realisierbar.

Ein weiteres bekanntes Verfahren ist der 1990 entwickelte IDEA Algorithmus. Mit einer Schlüssellänge von 128bit werden Blocks von 64bit Länge verschlüsselt. Hierbei wird durch bitweises Exklusiv oder, durch Addition mod 2^{16} (65536) sowie Multiplikation modulo $2^{16}+1$ (65537) ein verschlüsselter Text erzeugt. IDEA gilt als sicherer als DES oder sogar 3-fach DES.

7.5.2 Asymmetrische Verschlüsselung

Hier werden jeweils zwei Schlüssel berechnet, ein sogenannter Public Key und ein sogenannter Private Key. Beide Schlüssel stehen in einem komplexen mathematischen Zusammenhang. Zur Verschlüsselung von Nachrichten wird der Public Key verwendet, zum Entschlüsseln ein Private Key, der nur dem Empfänger bekannt ist. Dabei kann der Public Key jedermann bekannt sein. Solange der Private Key geheimgehalten wird, ist die Verschlüsselung sicher.

Im wesentlichen basieren diese Berechnungen auf Verwendung von Methoden der Restklassenalgebra. Dadurch sind die Verschlüsselungsalgorithmen wesentlich langsamer als symmetrische Verschlüsselungsverfahren, dies wird heute durch die ständig steigende Rechenleistung von Computersystemen aufgefangen. Eines der bekanntesten asymmetrischen Verfahren ist die sog. RSA (Ron Rivest, Adi Shamir, Leonard Adleman) Public Key Encryption. Ein anderes Verfahren ist DSA (Digital Signature Algorithm). Während RSA universell einsetzbar ist, ist DSA nur für digitale Signaturen einsetzbar.

Es gibt zwei wesentliche Anwendungen:

- Verschlüsselung: Hier wird eine Nachricht mit einem public key verschlüsselt, sie kann dann mit Hilfe eines nur dem Empfänger bekannten private keys nur von diesem Empfänger gelesen werden.
- Elektronische Signatur (z.B. in Mailsystemen): Hier kann eine Nachricht mit Hilfe eines nur dem Sender bekannten private keys signieren. Jeder mögliche Empfänger kann dann mit Hilfe des Public Keys überprüfen, ob die Nachricht auch wirklich authentisch ist, d.h. ob sie wirklich vom angegebenen Sender stammt.

In einer bestehenden Verbindung können auch beide Verfahren gleichzeitig zur Anwendung kommen, dann benötigen Sender und Empfänger zwei entsprechende Schlüsselpaare.

Basis für die Berechnung von Schlüsseln sind möglichst große Primzahlen (100 bis über 200 bits), wobei das Produkt dieser Primzahlen Teil der Schlüssel ist. Es wird vorausgesetzt, daß es genauso schwierig ist, einen mit einem Public Key verschlüsselten Text mit demselben Schlüssel wieder zu entschlüsseln, wie die Zerlegung des Produkts zweier Primzahlen zu finden.

Im folgenden wird gezeigt, wie man prinzipiell die Schlüssel berechnen kann:

Beispiel: Primzahlen $p=13$ und $q=19$,

Damit ergibt sich das Produkt $n=p*q$ zu 247.

Zur Findung des public keys wird noch die Funktion $s=(p-1)*(q-1)$ benötigt, hier ergibt sich s zu 216.

Nun wählt man einen public key (e,n) so, daß e und s keinen anderen gemeinsamen Teiler besitzen als die Zahl 1. Hier könnte z.B. $e=5$ zufällig gewählt werden (üblich sind 3, 17 oder $65537=2^{16}+1$).

Der public key ist in diesem Beispiel damit $(5,247)$.

Ein private key (d,n) wird nun so gewählt, daß der Rest der ganzzahligen Division von $d*e/s$ 1 ergibt. Hier kann z.B. $d=173$ gewählt werden. Damit ist der private key $(173,247)$.

Sind die Schlüssel generiert, werden p und q vernichtet

Verschlüsselt wird nun die Nachricht im Klartext: „12“

Verschlüsselung:

$12^5/247=248832/247=1007$ Rest 103, damit ist „103“ nun der verschlüsselte Text

Entschlüsselung:

$103^{173}/247=1,663*10^{349}/247=xxxx$ Rest 12, damit ist der entschlüsselte Text „12“

7.5.3 Digitale Signaturen

Zur Verwendung für Signaturen besteht das Problem, daß ein großer Aufwand getrieben wird, wenn die gesamte Nachricht verschlüsselt werden muß. Abhilfe schafft die Verwendung sogenannter Einweg-Prüfsummen (oder Message Digests bzw. One Way Hash-Funktionen) $h=H(M)$, wobei M die Nachricht ist (M für Message).

Ein solcher Message Digest besitzt (unabhängig vom Umfang der zu sichernden Nachricht) eine feste Länge m (wie z.B. 64 oder 128 bit, das National Institute of Standards and Technology NIST empfiehlt 160 bit für Hashfunktionen). Damit werden die Verschlüsselungs- und Entschlüsselungszeiten gegenüber der Zeit zur Verschlüsselung der ganzen Nachricht erheblich verkürzt.

Um fälschungssichere Nachrichten zu senden, muß ein Message Digest so generiert werden, daß folgende Eigenschaften erfüllt sind:

- Mit vorgegebenem M kann man h leicht berechnen
- Mit vorgegebenem h kann man M kaum berechnen
- Mit gegebenem M kann man kaum eine andere Nachricht N mit Message Digest $H(N) = H(M)$ finden.

Sehr oft ist auch noch eine weitere Eigenschaft, die sogenannte Kollisionsresistenz erforderlich:

- Es ist schwierig, zwei beliebige Nachrichten M und N zu finden mit $H(M)=H(N)$

Der entsprechende Message Digest identifiziert damit eindeutig eine bestimmte Nachricht, er liefert sozusagen einen eindeutigen Fingerabdruck.

Das folgende Beispiel zeigt, was geschieht, wenn die Kollisionsresistenz nicht erfüllt ist:

- Cäsar bereitet zwei Versionen eines Vertrags vor. Eine begünstigt Vertingetorix, die andere benachteiligt ihn.
- Cäsar bringt an jedem Dokument einige kleine Änderungen an und berechnet für jede Version den Hashwert. Nehmen wir an, er erzeugt so 2^{32} Dokumente.
- Er vergleicht danach die Hashwerte aller Versionen der beiden Dokumente und sucht übereinstimmende Paare. Er rekonstruiert die beiden Dokumente, die denselben Hashwert liefern.
- Cäsar läßt Vertingetorix die Vertragsversion unterzeichnen, die ihn begünstigt. Er benutzt dabei aber ein Protokoll, bei dem nur der Hashwert unterzeichnet wird.
- Später ersetzt Cäsar den von Vertingetorix unterschriebenen Vertrag durch die Version mit dem gleichen Hashwert, die Vertingetorix nicht unterschrieben hat und das Schlamassel ist perfekt.

Es gibt verschiedene Einweg Hashfunktionen. Die bekanntesten sind:

- Snefru (128 bzw. 256 bit)
- N-Hash (128 bit)

- MD2, MD4 und MD5 (MD steht für Message Digest) (128bit)
- SHA (Secure Hash Algorithm) (160bit)

Digitale Signatur und Public Key Verfahren werden häufig kombiniert und gewährleisten damit Sicherheit und Authentizität.

7.6 Beglaubigung (Authentication)

Verschlüsselung kann eine sichere Übertragung von Information gewährleisten. Was allerdings vorausgesetzt werden muß, ist, daß die beiden Kommunikationspartner auch die richtigen sind. Dies sicherzustellen, kann natürlich auf einem separaten Weg erfolgen, schwierig wird das Unterfangen allerdings, wenn zunächst eine Verbindung zu einem Kommunikationspartner aufgebaut werden soll, und in derselben Session eine gesicherte Verbindung hergestellt werden soll. So ist z.B. beim Einloggen in EDV-Systeme aber auch im Bereich des E-Commerce ist eine einwandfreie Identifikation eines Nutzers des Systems bzw. eines Kunden erforderlich. Hierzu werden häufig symmetrische und asymmetrische Verschlüsselung kombiniert.

Um diese Identifikation sicherzustellen, kann zunächst eine Beglaubigung erfolgen. Hier gibt es grundsätzlich mehrere Möglichkeiten:

- Eingabe eines geheimen Passworts (Hier kann ein System überprüfen, ob ein Nutzer das richtige Passwort hat)
- Überprüfung ob ein eingegebenes Passwort gültig ist (durch Anwendung einer Einweg-Hash-Funktionen auf das Passwort und Vergleich dieser Funktion mit dem früher gespeichertem Wert)

Bei der zweiten Möglichkeit sind die Passwörter selbst nicht mehr im System gespeichert, dies erhöht die Sicherheit. Allerdings ist auch eine Datei mit Hash-Funktionen von Passwörtern durch Vergleiche mit einer zweiten Datei mit Hashfunktionen gängiger Passwörter zu knacken. (z.B. über Angriff mit einem Wörterbuch).

Dies läßt sich durch Anwendung von Zufallssequenzen (Salt), die auf die Passwörter angewandt werden, bevor die Hashfunktion gebildet wird, erschweren, wenn auch nicht verhindern. Zusätzlich zu den Hash-Funktionen werden dann die Salt-Werte in der Datenbank gespeichert.

Beispiel: User Login bei Novell Netware

Zunächst hat der Server folgende Information über den Benutzer gespeichert:

- User Name
- Eine Userspezifische Zufallszahl (Salt)
- Eine Hash Funktion (Message Digest) $HSP = H1(\text{Salt}, \text{Passwort})$
- Public RSA Key des Users
- Private Key in verschlüsselter Form (verschlüsselt mit Hash Funktion HSP)

Die Authentisierung läuft nun in folgenden Schritten ab:

Erstellen der Verbindung:

Schritt 1: Der User verbindet sich mit dem Server

Schritt 2: Server überprüft, ob der User in der NDS existiert

Beginn des Logins:

Schritt 3: Server sendet den userspezifischen Salt Wert und eine allg. Zufallszahl (R1) (Nachricht A, Rätsel)

Berechnungen am Client:

Schritt 4: Client berechnet eine Hashfunktion $HSP = H1(\text{Salt}, \text{Passwort})$ selbst da diese nicht vom Server gesendet wird

Schritt 5: Client berechnet eine Zahl Y durch Verschlüsselung der Zufallszahl R1 mit HSP als Schlüssel (RC2 Verschlüsselung)

Schritt 6: Client erzeugt eine Zufallszahl R2 und einen Block von Zufallsdaten S

Schritt 7: Client erfragt den Public Key des Servers

Beenden des Logins:

Schritt 8: Verschlüsselung von R2, S, Y mit dem Public Key des Servers und Übertragung zum Server (Nachricht B, des Rätsels Lösung)

Schritt 9: Server entschlüsselt Y aus der ihm vorliegenden Hashfunktion HSP und Zufallszahl R1

Firewalls und Netzwerksicherheit

Schritt 10: Server berechnet ebenfalls $H_2(HSP, R_1)$ und vergleicht den Wert mit Y $\neq$ entspricht Passwortcheck

Schritt 11: Server verschlüsselt R_2 , den Private Key des Users und S mit Y als Schlüssel und sendet diese Nachricht zum Client (Nachricht C , Bestätigung des Zugriffs)

Client erhält Private Key vom Server:

Schritt 12: Client entschlüsselt die Nachricht und erhält den Private Key des Benutzers, dazu muß HSP von vorher verwendet werden

Schritt 13: Client checkt über R_2 , das die Nachricht vom richtigen Server kam, nur der konnte R_2 aus der Nachricht B entschlüsseln

Durch diese Kombination aus symmetrischer und asymmetrischer Verschlüsselung wird nachgeprüft, ob der Benutzer das richtige Passwort besitzt ohne daß das Passwort wirklich über die Leitung geschickt wird.

An weiteren Verfahren der Beglaubigung wie Smart Cards oder biometrische Verfahren (Fingerabdruck, Augenscan, usw.) wird gearbeitet.

7.7 Schlüsseltausch

Es wurde bisher gezeigt, daß asymmetrische Verschlüsselung auf einen sicheren Übertragungskanal zum Austausch von Schlüsseln verzichten kann. Auf der anderen Seite aber sind asymmetrische Verschlüsselungsverfahren wesentlich aufwendiger als symmetrische. Daher werden gerade zur Verschlüsselung langer Nachrichten nach wie vor gerne symmetrische Verschlüsselungsverfahren bevorzugt. Hier stellt sich aber, wie bereits gesagt, das Problem des Schlüsseltausches vor der eigentlichen Nachrichtenübertragung.

Der häufigste Weg, eine gesicherte Verbindung aufzubauen, ist, daß einer der Kommunikationspartner (z.B. Asterix) einen secret key an den anderen Kommunikationspartner (Obelix) überträgt, in dem er ihn mit dem public key von Obelix verschlüsselt. Für die eigentliche Dauer der Kommunikationsbeziehung wird nun als Sitzungsschlüssel der secret key verwendet (Dieses Verfahren wurde in obigem Beispiel eines Logins in ein Novell System bereits demonstriert).

Hier ist allerdings noch zu klären:

1. Wie der Sitzungsschlüssel generiert wird
2. Wie Asterix den public key von Obelix erhält
3. Wie Asterix sicher sein kann, daß er überhaupt mit Obelix kommuniziert

Zu 1. Generieren des Sitzungsschlüssels

Der Sitzungsschlüssel kann entweder durch Asterix selbst erfolgen und dann auf geeignetem Weg zu Obelix transportiert werden (z.B. verschlüsselt mit dem public key von Obelix) oder aber durch das Einschalten einer neutralen Stelle, mit der beide Kommunikationspartner vorab einen Schlüssel vereinbart haben.

Zu 2. Erhalt des public keys von Obelix

Asterix kann den public key wie folgt erhalten haben:

- von Obelix selbst oder
- aus einer Datenbank für public keys eines Dritten (eines sog. Trust Centers)

Die letztere Variante hat den Vorteil, daß eine Kommunikation auch mit einem anderen (auch bisher unbekanntem) Gesprächspartner erfolgen kann, wenn derjenige einen public key besitzt, der dem Trust Center bekannt ist.

Zu 3: Gegenseitige Authentizität der Kommunikationspartner:

Hier müssen entsprechende Verfahren zur Verteilung öffentlicher Schlüssel zum Einsatz kommen oder einen Mechanismus geben, mit dem nachgewiesen werden kann, dass die Verteilung richtig erfolgt ist.

7.7.1 Diffie Hellman Methode

Eine mögliche Variante des Schlüsseltauschs für symmetrische Schlüssel ist das Diffie Hellmann Verfahren. Es funktioniert wie folgt:

Asterix und Obelix müssen sich auf zwei große Primzahlen n (so daß $(n-1)/2$ muß auch eine Primzahl ist) und g einigen, diese können auch öffentlich bekannt sein. Asterix wählt nun eine große Zahl (512bit) x , Obelix entsprechend y . Diese Zahlen werden geheimgehalten.

Beispiel: $n=47$ und $g=3$

Asterix wählt $x=8$, Obelix $y=10$

Asterix sendet: $(n, g, g^x \bmod n)$ entspricht: $(47, 3, 28)$ mit $3^8 \bmod 47 = 28$

Obelix sendet: $g^y \bmod n = 3^{10} \bmod 47 = 17$

Asterix berechnet: $(g^y \bmod n)^x = 17^8 \bmod 47 = 4$

Obelix berechnet: $(g^x \bmod n)^y = 28^{10} \bmod 47 = 4$

Nach den Gesetzen der modularen Arithmetik ergeben beide Berechnungen $(g^{x \cdot y} \bmod n) = 4$ als Schlüssel.

Trotz der Eleganz dieses Verfahrens ist es angreifbar, und zwar durch eine sogenannte Man-In-The-Middle Attack. Der Man-In-The-Middle nimmt einfach am Verfahren teil in dem er sich eine Zahl z überlegt und damit sowohl Asterix als auch Obelix täuscht.

7.7.2 Schlüsseltausch über Dritte

Statt einer direkten Kommunikation gibt es die Möglichkeit für den Generation oder aber auch nur den Austausch von Schlüsseln eine dritte Stelle einzuschalten, der beide vertrauen. Diese Stelle wird als Schlüsselverteilungszentrum (Key Distribution Center KDC), Trusted Authority (TA) oder Trust Center bezeichnet. Beispielsweise kann sich nun einer der Kommunikationspartner (Asterix) an das Trust Center wenden, um einen Sitzungsschlüssel für die Kommunikation mit einem anderen Teilnehmer (Obelix) zu erhalten. Dieser wird vom Trust Center generiert und in zwei Versionen (jeweils verschlüsselt mit secret keys von Asterix und Obelix) an Asterix gesandt. Asterix decodiert seinen Sitzungsschlüssel und sendet die zweite Version an Obelix, der seine Version des Sitzungsschlüssels ebenfalls dekodiert. Damit kann der Sitzungsschlüssel für die nachfolgende Kommunikationsbeziehung verwendet werden.

7.7.3 Kerberos Protokoll

Es gibt verschiedene Standardprotokolle für Beglaubigung und Schlüsseltausch, wie Wide-Mouth Frog, Yahalom, Needham-Schroeder, Otway-Rees, Dass oder Woo-Lam. Eines der bekannten ist Kerberos, das am MIT (Massachusetts Institute of Technology) entwickelt wurde. Es wird im wesentlichen als verteilter Dienst verwendet,

der es einem Client ermöglicht, sich bei einem oder mehreren Servern zu beglaubigen. Optional kann eine Verschlüsselung der Kommunikation erfolgen.

Ein Kerberos Key Distribution Center besteht aus zwei Services, dem Authentication Service und dem sogenannten Ticket Granting Service (TGS). Der Authentication Service kennt alle secret keys aller Teilnehmer

Kerberos verwendet drei Sicherheitsobjekte:

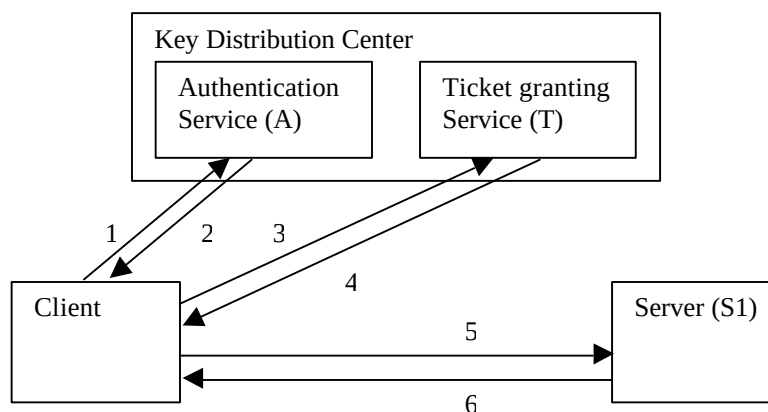
Kerberos Ticket: Für den Zugriff auf den Server werden sogenannte Kerberos Tickets erzeugt. Ein Ticket enthält folgende Informationen:

- den Namen des Clients
- die Identification des Servers
- ein Zeitintervall für die Gültigkeit des Tickets
- einen Sitzungsschlüssel für den Zugriff auf den Server

Authenticator: sichert die Identität des Clients gegenüber dem Server. Er wird vom Client erzeugt und mit einem Sitzungsschlüssel verschlüsselt, er enthält die aktuelle Uhrzeit und eine Prüfsumme

Sitzungsschlüssel: Dient der verschlüsselten Kommunikation zwischen Client und Server

Ein Beglaubigungsvorgang bei einem Server läuft wie folgt ab:



1. Der Client fordert ein Ticket für den Zugriff auf den Ticket Granting Service vom Kerberos Authentication Service an und sendet dazu
 - Seine eigene Identität
 - Die Identität des Servers T
 - Eine Zufallszahl (engl. Nonce)
2. Der Authentication Service sendet
 - einen Sitzungsschlüssel K_{CT} für den Zugriff auf den Service T sowie die Nonce, verschlüsselt mit dem Clientschlüssel (der aus dem Passwort des Nutzers gebildet wurde)
 - ein Ticket, um sich beim Server T auszuweisen, verschlüsselt mit dem Schlüssel des Servers T

3. Der Client möchte mit dem Server S1 kommunizieren und sendet
 - das von A erhaltene Ticket zum Server T
 - seinen Authenticator
 - die Identification für S1
 - eine Nonce (Zufallszahl)

Nachdem Authenticator und Ticket beide mit dem Sitzungsschlüssel K_{CT} verschlüsselt sind, dieser aber nur dem Client und dem Authentication Service bekannt sind, ist hierdurch die Beglaubigung erfolgt.
4. Der Ticket Granting Server prüft das Ticket und sendet
 - einen Sitzungsschlüssel K_{CS1} für den Zugriff auf Server S1 und die Nonce des Clients, verschlüsselt mit dem aktuellen Sitzungsschlüssel K_{CT} sowie
 - ein Ticket für S1
5. Für jede Nutzung des Servers S1 (d.h. für jeden Request) sendet der Client
 - seinen Authenticator und
 - das Ticket für S1.
6. Optional sendet der Server selbst die Nonce zurück an den Client, um sich selbst zu authentisieren.

Auf diese Weise werden Sitzungsschlüssel generiert und eine gegenseitige Beglaubigung durchgeführt.

7.8 Zertifizierung

Bei Anforderung eines Schlüssels aus einer zentralen Datenbank könnte es evtl. geschehen, daß die Datenbankabfrage nach dem public key von Obelix abgehört wird und Asterix einen anderen Schlüssel als den von Obelix erhält, ohne es zu bemerken. Auf der anderen Seite könnte auch Obelix seinen public key an Asterix schicken, jemand könnte ihn abfangen und statt dessen seinen eigenen Public key senden. Wie können nun Asterix und Obelix verifizieren, daß sie tatsächlich miteinander und nicht mit einem dritten kommunizieren.

Dieses Problems läßt sich durch die Verwendung von Zertifikaten, die von einem Trust Center (in diesem Fall auch als Certificate Authority (CA) bezeichnet) lösen. Diese Organisation überprüft z.B. die wirklichen Identitäten dieser beiden Kommunikationspartner und stellt hierüber ein elektronisches Zertifikat aus, das dann für die Kommunikation verwendet werden kann. Damit entspricht die Funktion einer solchen Organisation in etwa der eines Notars. Erhält Vertingetorix von Cäsar ein beglaubigtes Dokument, so kann Vertingetorix davon ausgehen, daß Cäsar das Dokument wirklich eigenhändig unterschrieben hat, der Notar könnte dies gegebenenfalls auch bezeugen.

Der wichtigste Standard für digitale Zertifikate ist der X.509 Standard. Ein X.509 Zertifikat, das Obelix nun an Asterix schicken kann, muß folgende Informationen enthalten:

- Den Namen des Eigentümers des Zertifikats (Asterix)
- Einen seiner öffentlichen Schlüssel (public key)
- Den Namen der CA
- Eine digitale Signatur der CA

Nachdem verschiedene Algorithmen für digitale Signaturen verwendet werden können, muß der benutzte Algorithmus ebenfalls angegeben werden. Zudem haben Zertifikate oft eine begrenzte Gültigkeit. Zertifikate können bei Mißbrauch von der ausstellenden CA widerrufen werden. Sie veröffentlichen dazu eine sogenannte Certificate Revocation List (CRL), die periodisch aktualisiert wird. Jeder Teilnehmer einer Kommunikation, der ein Zertifikat erhält kann anhand dieser Liste überprüfen, ob es noch gültig ist.

Für jedes Schlüsselpaar (private und public key) wird ein eigenes Zertifikat ausgestellt. Wenn genügend Personen einer bestimmten Certificate Authority vertrauen, läßt sich so eine ganze Hierarchie von gesicherten Verbindungen realisieren.

Wichtig zu wissen ist, dass nicht der Besitz eines Zertifikats ausschlaggebend für den Nachweis einer Identität ist (Zertifikate sollen ja frei verteilt werden können), sondern der Nachweis, dass man im Besitz des private keys ist, der zum im Zertifikat genannten public key gehört.

7.9 Anwendungsbeispiele

In der Internet Protokollwelt gibt es heute folgende wesentliche Verfahren zum Aufbau verschlüsselter Verbindungen

- Verschlüsselung der Dienstprotokolle: Die wird über eine eigene Schicht oberhalb des Transportprotokolls, die sogenannte Secure Socket Layer (SSL) erreicht, auf die dann beispielsweise http aufsetzen kann (wird dann als secure http bzw. https bezeichnet)
- Verschlüsselung auf der IP-Schicht über IPSEC

Auch im Bereich eMail spielen die vorgestellten Mechanismen heute eine große Rolle zur Bereitstellung von Verschlüsselungs- und Beglaubigungsmechanismen. Die wesentlichen sind S/MIME, PEM und PGP.

7.9.1 Secure Socket layer (SSL)

SSL wurde 1995 von Netscape als Sicherungsprotokoll für die Transportschicht in die IETF eingebracht und ist die Grundlage für das weiterentwickelte TLS (ransport Layer Security Protocol). Ursprünglich wurde es zur Absicherung einer http Kommunikation entwickelt, ist inzwischen auch für andere Dienstprotokolle verfügbar (siehe nachfolgende Portliste). Im Internet Modell sitzt SSL sitzt sozusagen zwischen der Transportschicht und der darüberliegenden Dienstprotokollschicht (bzw. Session Layer im OSI Modell).

Folgende well known ports werden für SSL verwendet:

Port	Protokoll	Bedeutung
443	https	http over SSL
465	ssmtp	smtp over SSL
563	snntp	nntp over SSL
992	telnets	telnet over SSL
995	spop3	pop3 over SSL

Beim Aufbau einer SSL Verbindung wird zuächst ein Handshake durchgeführt:
Eim Verbindungsaufbau eines Clients zu einem Server erfolgt wie folgt:

1. Client Hello: Client teilt dem Server mit:
 - welche kryptographischen Verfahren er unterstützt.
2. Server Hello: Übermittelt werden:
 - Das kryptographische Verfahren
 - Ein Serverzertifikat (optional)
 - Parameter für einen Schlüsseltausch (optional)
 - Eine Zertifikatsanforderung zur Beglaubigung des Clients (optional)
3. vom Client zum Server: übermittelt werden:
 - das Clientzertifikat (wenn angefordert)
 - Parameter für Sitzungsschlüssel, verschlüsselt mit dem public key des Serves
 - Digitale Signatur zur Überprüfung des Client Zertifikats (optional)

7.9.2 IP Security (IPSEC)

IPSEC stellt nicht ein einziges Protokoll dar, sondern eine Architektur zum Aufbau verschlüsselter Kommunikationsbeziehungen IPSEC ermöglicht den Einsatz

- verschiedener Verschlüsselungsalgorithmen
- verschiedener Sicherheitsdienste (Zugangskontrolle, Beglaubigung, Integrität, Vertraulichkeit)
- Kontrolle, wie detailliert die Sicherheitsdienste angewandt werden.

IPSEC ermöglicht den Aufbau von verschlüsselten Kanälen zwischen Routern oder von Clients zu Routern und ist damit eine Schlüsseltechnologie für den Aufbau von VPNs (Virtual Private Networks).

IPSEC besteht im wesentlichen aus zwei Teilen:

Teil 1: Protokolle zur Implementation der Sicherheitsdienste:

 AH (Authentication Header) und ESP (Encapsulating Security Payload)

Teil 2: Schlüsselmanagement:

 ISAKMP Protokoll (Internet Security Association and Key Management Protocol)

7.9.3 PGP (Pretty Good Privacy)

Eines der grundsätzlichen Probleme bei der Verteilung von public keys ist das Problem, eine Vertrauenskette zu bilden. PGP geht davon aus, dass jeder Benutzer seine eigenen Kriterien hat, nach denen er anderen zertifizierten Schlüsseln vertraut (z.B. vertraut er evtl. Schlüsseln, die er direkt von ihm bekannten Personen erhalten hat) und wie weit er diesen Zertifikaten vertraut. PGP erlaubt daher eine beliebige Verflechtung von Zertifikaten. Ein Benutzer signiert nun zunächst seinen public key selbst und kann nun seinen Schlüssel mit beliebigen anderen e-Mail Teilnehmern austauschen und deren Schlüssel ebenfalls signieren. Im Lauf der Zeit ergibt sich eine Sammlung unterschiedlichster Schlüssel und Zertifikate die als Key Ring bzw. Schlüsselbund bezeichnet wird. Zu jedem Schlüssel wird gespeichert ein Zertifikat gespeichert, das festhält, mit welchem Algorithmus der Schlüssel erzeugt wurde (RSA, DSS etc.) und welcher Vertrauensumfang diesem Zertifikat entgegengebracht wird. Über den Vertrauensumfang wird sozusagen festgelegt, ob man einer bestimmten Person zutraut, das Verfahren zur Zertifizierung von Schlüsseln zu beherrschen oder nicht.

Je nach Version von pgp werden unterschiedliche Verschlüsselungsverfahren und Algorithmen verwendet.

Das Beglaubigen von Nachrichten wird wie folgt durchgeführt:

Der Sender erzeugt aus dem Rumpf der Nachricht eine Einwegprüfsumme und verschlüsselt diese mit seinem privaten Schlüssel. Welches Verfahren zur Bildung der Prüfsumme herangezogen wurde (z.B. MD5) wird in der Nachricht spezifiziert.

Der Empfänger durchsucht bei Erhalt der Nachricht seinen key-Ring nach dem öffentlichen Schlüssel von A, berechnet ebenfalls die Prüfsumme aus der Nachricht und vergleicht diese mit der erhaltenen Prüfsumme. Zudem teilt PGP dem Empfänger mit, welches Vertrauensmaß der Empfänger dem öffentlichen Schlüssel des Senders entgegenbringt, und zwar auf der Grundlage der vorhandenen Zertifikate, die im Key Ring für den Sender enthalten sind.

Nachrichtenverschlüsselung wird wie folgt durchgeführt:

Der Sender verschlüsselt die Nachricht mit einem Einmalschlüssel auf Basis eines symmetrischen Algorithmus (z.B. DES). Dieser Einmalschlüssel wird mit dem public key des Empfängers verschlüsselt. Die Nachricht wird in ASCII kodiert und zusammen mit dem Schlüssel übertragen. Der Empfänger dekodiert zunächst den Schlüssel und damit dann die eigentliche Nachricht.

7.10 Sicherheitspolicen

Um ein Netzwerk sicher zu machen, muß für jedes Netz ein eigener Sicherheitsstandard festgelegt werden, der folgende Punkte berücksichtigt:

- Welche Netzwerkverbindungen werden genutzt und sind insofern angreifbar
- Risikoanalyse
- Untersuchung der möglichen Angriffsarten auf das Netz
- Check und Einführung von Schutzmechanismen
- Unabhängige Evaluierung

Es gibt unabhängige Organisationen, die für Betriebssystem und Netze Sicherheitszertifikate vergeben. Das vielleicht bekannteste ist in USA das NCSC (National Computer Security Center). Hier werden Systeme nach Klassen eingeteilt, das zugrundeliegende Konzept heißt TCSEC (Trusted Computer System Evaluation Criteria)

Class D (Minimaler Schutz)

Class C1 (vorhandene Zugriffssicherheit)

Class C2 (Kontrollierte Zugriffssicherheit)

Class B1 (Bestätigte Zugriffssicherheit)

Class B2 (Strukturierter Schutz)

Class B3 (Sicherheitsdomänen)

Class A1 (Verifiziertes Design)

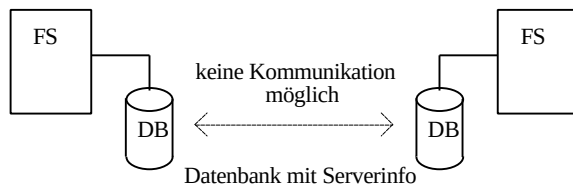
In Europa werden Produkte gegen die ITSEC (Information Technology Security Evaluation Criteria) Kriterien getestet. Hier werden dann von europäischen Organisationen Zertifikate für die Integrität und die Effektivität von Systemen vergeben. Typische Zertifikate sind E2-Level für Effektivität und F-C2 für Funktionalität.

E2 kombiniert mit F-C2 entspricht dann etwa wieder der C2 Zertifizierung der NCSC.

Betriebssystem wie Windows NT und NetWare sind z.B. C2 zertifiziert, allerdings mit einem sehr großen Unterschied: Bei Novell Netware gilt C2 für ein komplettes Netzwerk, bei Windows NT nur für den Rechner mit dem lokalen Betriebssystem an sich.

8. Verwaltung von Computernetzen

8.1 Serverbezogene Administration



Datenbank mit Serverinfo (Novell 3: Bindery)

- Objekte
- User
 - Gruppen
 - Zugriffsprofile für User
 - Volumes (Filesystem)
 - Dienste (Gateway, ...)

Information ist serverbezogen

Problematic: Abgleich der Information bei mehreren Servern muß manuell erfolgen, d.h z.B. muß ein und derselbe Benutzer auf drei verschiedenen Servern definiert werden, weil er dort auf Daten zugreifen will. NetWare 3 ist aufgrund der hohen Stabilität heute noch relativ verbreitet (vor allem bei kleineren Firmen, wo die Nachteile einer serverbezogenen Administration noch keine so große Rolle spielen).

8.2 Domänenkonzept

Grundsätzlich löst das Domänenkonzept das Problem, das Benutzer im Netzwerk mehr als einmal definiert werden müssen. Alle Server, für die eine zentrale Benutzerverwaltung erfolgen soll, werden in einer Domäne zusammengefaßt. An einer zentralen Stelle, dem sogenannten Domänenkontroller, wird die gesamte Benutzerverwaltung des Netzwerks zentralisiert. Als Ausfallsicherung steht noch ein Sicherungs-Domänenkontroller zur Verfügung. Microsoft Windows NT verwendet die Bezeichnungen PDC (Primary Domain Controller) und BDC (Backup Domain Controller), Windows 2000 unterscheidet hier nicht mehr, sondern verwendet nur DCs (Domänencontroller)

Werden mehrere Domänen zusammengefaßt, können bei Windows NT/2000 zwischen den Domänen Vertrauensstellungen definiert werden. Das führt dazu, daß eine Domäne der anderen vertraut, die Aufgabe der Authentisierung (Beglaubigung) von Benutzern im Netzwerk richtig durchzuführen. Ein Benutzer loggt sich in eine Domäne ein und kann dann auch auf Ressourcen einer weiteren Domäne zugreifen, wenn diese der ersten vertraut. Vertrauensstellungen sind einseitig oder zweiseitig, ab Windows 2000 auch transitiv. Bei Nicht-transitiven Vertrauensbeziehungen würde das bedeuten, dass wenn z.B. Domäne 1 der Domäne 2 vertraut und Domäne 2 der Domäne 3, dies nicht bedeutet, dass auch Domäne 1 automatisch Domäne 3 vertrauen würde.

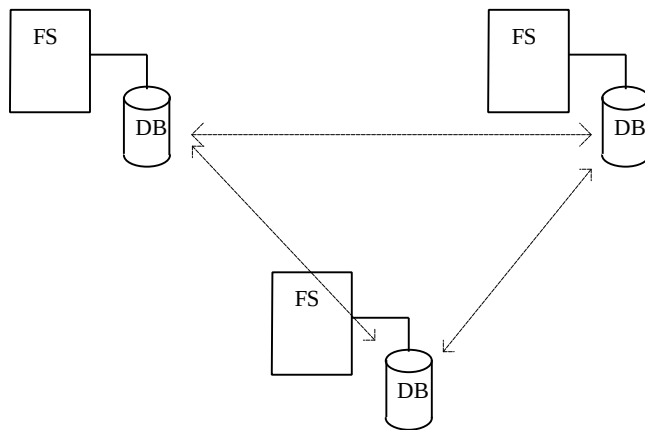
Die Problematik besteht bei Windows NT in folgenden Aspekten:

- ✗ Trustbeziehungen sind nicht transitiv, was bei mehr Domänen unübersichtlich werden kann.
- ✗ Innerhalb der Domäne gibt es keine Hierarchien von Objekten, also z.B. keine hierarchische Anordnung von Benutzern, sondern nur Listen, die entsprechend lang werden können.

Beide Nachteile werden mit Active Directory Service von Microsoft (Windows 2000) behoben.

8.3 Directory Service

Die Aufgabe einer Directory (deutsch: Telefonbuch) ist es, Netzwerkinformation netzwerkweit, aber zentral administrierbar zur Verfügung zu stellen. Über einen Directory Service kann man dadurch z.B. die Zugriffsrechte auf alle Ressourcen netzwerkweit steuern.



Kennzeichen einer solchen Datenbank mit Information über Netzwerk Ressourcen sind:

- ☞ Verteilte Struktur
- ☞ Hierarchischer Aufbau

Partitionen (Teile) der Datenbank und Kopien dieser Partitionen werden auf mehrere Server verteilt. Dies bietet zunächst einige Vorteile:

- ☞ Lastverteilung bei der Authentisierung
- ☞ Backup von Netzwerkinformation verteilt

Um dies zu erreichen, sind allerdings erhöhte Anforderungen an die Bandbreite zwischen Servern zur Datenübertragung notwendig, aber heute zunehmend auch gegeben.

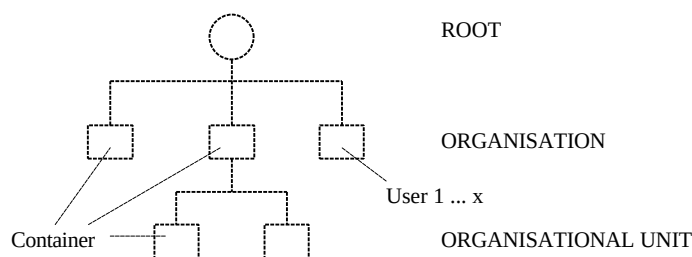
Objekte in einem Directory Service sind netzwerkweit verwendbar, nicht mehr nur serverbezogen. Der Server, der bisher Zentrum der Administration war, wird nun ein Objekt in der Ressourcendatenbank.

8.3.1 X.500 Directory Standard

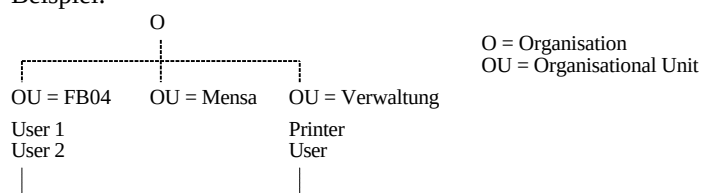
Die meisten Directory Dienste folgen dem ITU Standard X.500. Dieser beschreibt das Modell einer allgemein verwendbaren Directory zur hierarchischen Speicherung beliebiger Netzwerkinformation.

Die hierarchische Speicherung von Information wird in einer Baumstruktur realisiert. Man unterscheidet:

- Blattobjekte (Leaf - Objects)
- Container - Objekte (Zusammenfassung mehrerer Blattobjekte)

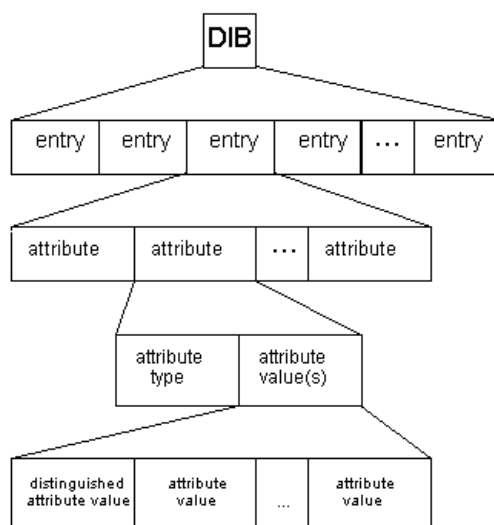


Beispiel:



Wenn man in dieser Struktur Information finden möchte (z.B. einen Usernamen, wenn dieser Benutzer sich einloggen möchte), muß neben dem Eigennamen des Objekts (sog. Common Name, kurz: CN) auch der Standort des Objekts in der Directory (der sog. Kontext) angegeben werden. Aus einem Benutzer Maier wird damit z.B. ein Benutzer .Maier.FB04.FHM . Dies wird als vollständiger Name (Full distinguished Name) bezeichnet. (Der führende Punkt des im an Novell NDS angelehnten Beispiels kennzeichnet den absoluten Pfadnamen des Objekts relativ zur Wurzel des Baumes. Alternativ dazu kann durch Weglassen eines führenden Punkts auch eine relative Pfadangabe zum momentan an der Workstation des Benutzers voreingestellten Kontext gemacht werden).

Den grundsätzlichen Aufbau eines Eintrags in der Directory zeigt nachfolgendes Bild (Quelle: D. Chadwick: Understanding X.500 - The Directory)



Die gesamte Information wird in einer sogenannten Directory Information Base (DIB) gespeichert. Zu jedem Objekt (hier Eintrag bzw. Entry genannt) werden entsprechende Eigenschaften (attributes) gespeichert, denen dann bestimmte Werte (Values) zugeordnet werden.

8.3.2 Navigation durch eine Directory

Directories arbeiten nach dem Prinzip von Verweisen (engl. Referrals). Nachdem die gesamte Informationsmenge unter Umständen auf verschiedenen Server aufgeteilt ist, verweist ein Server, der die von einem Client gewünschte Information nicht in seinem Datenspeicher findet, auf einen anderen. Welcher Server das ist, wird durch die Auflösung des Full Distinguished Names (also des vollständigen Namens) eines Objekts festgestellt. Hierbei wird der Namensbaum traversiert, dabei kann es sein, dass die entsprechende Datenbasis eines bestimmten Teilbaumes auf einem anderen Server liegt, auf den dann entsprechend verwiesen wird.

8.3.3 Aufgaben einer Directory

Wir haben im letzten Kapitel bereits die Notwendigkeit der Speicherung von Daten über die Netzwerkinfrastruktur kennengelernt (z.B. Adress und Namensinformation über Netzwerkteilnehmer). Die offene Struktur einer Directory ermöglicht grundsätzlich die Speicherung weiterer Information. Damit können über entsprechende Objekte im Directory Service folgende Dienste bzw. Funktionen realisiert werden:

- Benutzerverwaltung (durch Speicherung von Benutzerinformation)
 - Anwendungsmanagement (durch Steuerung des Ressourcenzugriffs einzelner Benutzer)
 - Softwareverteilung (durch Speicherung des aktuellen Software-Bestands einzelner)
 - Desktopmanagement (durch zentralisiertes Policy Management)
 - Erzeugung und Verteilung von Schlüsseln für die verschlüsselte Kommunikation
 - Management von Netzwerkhardware (die auch durch entsprechende Objekte repräsentiert wird)
- Näheres dazu im Kapitel Netzwerkmanagement

8.3.4 Realisierung in den verschiedenen Betriebssystemen:

Novell 4/5/6: NDS (Novell Directory Service), neuere Bezeichnung: eDirectory:

- Echte hierarchische Struktur (X.500)
- Login erfolgt über Angabe des kompletten Objektnamens in der Struktur:
Login .CommonName.OrganisationalUnit.Organisation
z.B. Login: .Maier.FB04.FHM
Vereinfachung: Voreinstellung von '.FB04.FHM' über lokale Konfiguration der Workstation des Benutzers oder über
Catalog Service: Vorhalten aller Benutzer in einem speziellen Katalog in der NDS
- Plattformübergreifend: Novell Netware, Windows NT (mit NDS for NT auch plattformübergreifende Benutzerverwaltung), Windows 2000, Sun Solaris, Linux, IBM OS/390 Großrechner

Microsoft Windows 2000 Active Directory:

- Keine echte hierarchische Struktur (Hierarchie nicht innerhalb von Domänen)
- Speicherung der Benutzer: Userverwaltung erfolgt nach wie vor in flacher Datenbankstruktur, d.h. Benutzer loggt sich nach wie vor mit einem eindeutigen Namen in die Domäne ein
- Trustbeziehungen zwischen Domänen durch Baumstruktur realisiert, Trusts sind dabei transitiv

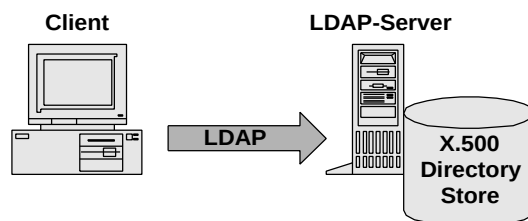
Microsoft Exchange Directory:

- Vorläufer von Active Directory für Enterprise Mailsystem Exchange
- Adress-Verwaltung von User Mailboxen
- Exchange 2000 ↗ Integration mit ADS

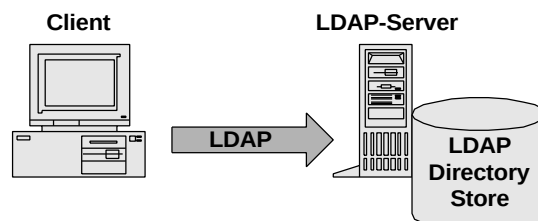
8.3.5 Universeller Zugriff auf Directories

X.500 setzt als Directory Standard auf OSI Protokollen auf (OSI: s. später), die sich nicht durchgesetzt haben. Proprietäre Zugriffsprotokolle der Directory Hersteller sind zwar unter Umständen sehr mächtig und nachdem sie der Struktur der einzelnen Directory Services angepasst sind, schon dadurch inkompatibel zueinander. Insofern hat man ein Standardprotokoll gesucht, das zunächst auf TCP/IP aufsetzen kann und zudem einfacher von der Struktur her ist.

Hierzu wurde das LDAP (Lightweight Directory Access Protokoll) standardisiert, dessen Version 2 ein Standardzugriffsprotokoll auf X.500 kompatible Directories ist.



LDAP wurde von einem reinen Zugriffsprotokoll weiterentwickelt zur Version 3, das eine eigene Directory standardisiert.



Die meisten der heute kommerziell einsetzbaren Directory Services (z.B. AD, NDS) sind LDAPv3 kompatible Directories. Dies ermöglicht den Zugriff eines LDAP Clients (über ein Benutzerinterface oder auch aus

Anwendungen heraus) auf verschiedene Directories. Nachdem eine Directory eine feste Struktur bietet, innerhalb der ganz unterschiedliche Informationen abgelegt werden können (Netzwerkinformation, aber auch andere, allgemeiner Information) bietet diese Entwicklung den gemeinsamen, standardisierten Zugriff auf Information auf unterschiedlichsten Systemen.

8.3.6 Metadirectories

In einem heterogenen Netzwerk kommen unter Umständen verschiedene Directories zum Einsatz, oft kommen Datenbanken zum Einsatz, die ebenfalls Informationen enthalten, die sich mit dem Inhalt einer unternehmensweiten Directory decken (z.B. Namen von Mitarbeitern). Insofern wäre es hilfreich, wenn eine Abgleichmöglichkeit zwischen den verschiedenen Directories und Datenbanken bestünde. Um bei n verschiedenen Datenbanken/Directories nicht n verschiedene Schnittstellen unterstützen zu müssen, bietet sich das Konzept einer übergeordneten Zwischendirectory an, die als Metadirectory bezeichnet wird. Die eigentliche Datenbank dieses Directory Services stellt sozusagen den kleinsten gemeinsamen Nenner aller austauschbaren Directory Information dar.

Ein Beispiel für eine Metadirectory ist z.B. DirXML von Novell, das auf den XML Standard zur Übertragung von Nachrichten zwischen den angeschlossenen Directories aufsetzt.

8.4 Zugriffskontrolle in Netzen

8.4.1 Kennwortschutz

Die erste Hürde für einen Benutzer, der auf Information im Netzwerk zugreifen will, ist der Passwortschutz, wobei häufig mehrere Passworte benötigt werden (und damit ggf synchronisiert werden müssen)

Häufig erfolgt ein Einloggen in die

- lokale Workstation (z.B. BIOS Passwort, Benutzerkennwort für lokales Windows NT/2000)

und in den

- Server bzw. das Netzwerk

Hier besteht die Problematik, dass das Kennwort zwischen Client und Server übertragen werden muß. Dazu unterscheidet man zwischen:

- Übertragung unverschlüsselt (frühere Netzwerkbetriebssysteme, Hostsysteme)
- symmetrische Passwort-Verschlüsselung
- Authentication (Beglaubigung)

Mehr dazu im Kapitel Netzwerksicherheit

8.4.2 Zugriffsrechte

Man unterscheidet grundsätzlich zwei verschiedene Methoden für die Steuerung des Zugriffs auf Information durch Benutzer:

- Das Capabilities Schema und
- Zugriff über Zugriffssteuerungslisten (Access Control Lists)

Bei Verwendung des Capabilities Schemas wird für jeden Benutzer im Netz gespeichert, was er darf und was nicht. Access Control Lists (kurz: ACL) speichern für jedes Objekt in der Directory bzw. auch im Filesystem ab, welcher Benutzer hier was darf und was nicht. Das Konzept ist allerdings von Benutzerrechten auf Rechte anderer Objekte übertragbar. So kann man z.B. den Zugriff eines Dienstes auf irgendwelche Daten auch über Rechte steuern etc.

Sinnvolle Rechte, die Objekte in einer Directory haben können, sind:

- Rechte auf Objekte
- Rechte auf Objekteigenschaften

Im Filesystem sind dies (je nach verwendetem Filesystem unterschiedlich):

- Rechte auf freigegebene Verzeichnisse
- Rechte auf Directory, Files (User - bezogen), je nach Filesystem unterschiedlich
- Rechte auf Directory, Fileattribute (Filebezogen), je nach Filesystem unterschiedlich

Üblicherweise können Rechte innerhalb der verschiedenen Ebenen des Filesystems und innerhalb der Directory Services vererbt werden. Dies erleichtert die Administration von Rechten erheblich. Allerdings bestehen zwischen den verschiedenen Netzwerkbetriebssystemen erhebliche Unterschiede bezüglich der Art und Weise der Vererbung. Man unterscheidet:

- dynamische Vererbung und
- statische Vererbung

Beispiel Novell Netware: Novell unterstützt sogenannte dynamische Vererbung, d.h. Rechte werden nur an wenigen Stellen im System gespeichert, bei jedem Datenzugriff wird dann errechnet, welcher aktuelle Stand der Rechtevererbung für einen bestimmten Benutzer besteht.

Beispiel Windows NT/2000: Windows verwendet statische Vererbung. Hier können z.B. Rechte von einer Ebene des Filesystems explizit per Knopfdruck auf darunterliegende Ebenen übertragen werden. Sie werden dann dort ebenfalls gespeichert. Der Vorteil ist, dass nicht bei jedem Datenzugriff die aktuellen effektiven Rechte eines Berechtigten ermittelt werden müssen, der Nachteil besteht darin, dass die zu speichernde Informationsmenge über Rechte um ein Vielfaches größer ist als im obigen Fall (Bei einer Directory Tiefe von z.B. 5 Ebenen wird 5 mal dieselbe Information gespeichert).

Single Sign On:

Heute sind Netze meist heterogen realisiert, d.h. es kommen unterschiedliche Betriebssysteme und Hardwareplattformen zum Einsatz. Dies bedeutet für den Benutzer, daß er in den unterschiedlichen Systemen bekannt sein muß.

Eine gemeinsame Administration von einem zentralen Punkt ist dabei über einen plattformübergreifenden Directory Service, eine Direkte Kopplung von einzelnen Directories oder über Metadirectories zu erreichen. Dabei ist auch ein gemeinsames Login für den Benutzer ein Ziel (Single Sign On). Durch entsprechenden Abgleich von Benutzerpassworten zwischen den Systemen wird so der Login Vorgang vereinheitlicht. Dies stellt allerdings gleichzeitig eine Erleichterung und eine Gefahr dar.

(↯ Kapitel Netzwerksicherheit).

9. Netzwerkmanagement

Man unterscheidet:

- Netzwerkmanagement
- Desktopmanagement

9.1 Netzwerkmanagement:

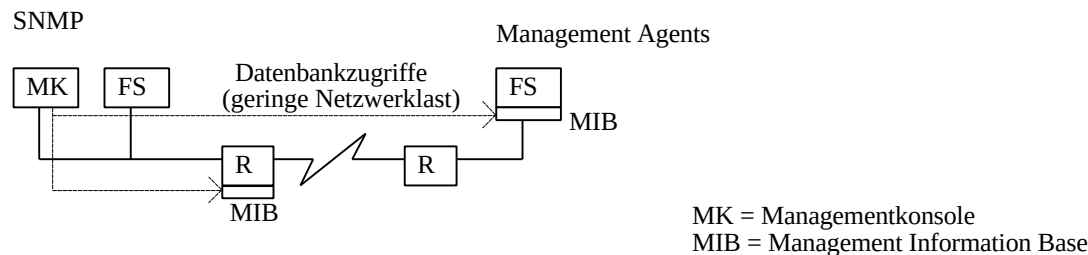
Standardmanagement Modell der ISO:

5 funktionale Bereiche:

- Fault Management
Fehlerbenachrichtigung, Fehlererkennung, Fehlervoraussage, (Proaktives Management)
- Performance Management
Überwachung der Performance (Leistungsfähigkeit) von Netzkomponenten der Netzverbindungen
- Configuration Management
Netzwerkübersicht (Map), Überwachung und Fernsteuerung von Netzwerkkomponenten, Fernkonfiguration
- Security Management
Reourcenverwaltung, Schutz vor unerlaubten Zugriff, Benutzerverwaltung
- Accounting Management
Berechnung oder Abrechnung von Services

Aufbau eines Managementsystems:

Im Allgemeinen besteht ein Netzwerkmanagementsystem aus zwei Komponenten, einer Managementstation und einem überwachten Rechner. Der Rechner wird durch eine spezielle Schnittstelle, dem sogenannten Management-Agent, managebar. Der Agent ist als Software auf dem zu überwachenden Rechner installiert.



In Band Management: Management über das LAN

Out Band Management: Management über redundanten Weg (Dedizierte Leitung / Modem)

Management Agents:

Die Agents haben unterschiedliche Funktionalität, z.B.:

Serverüberwachung:

- Server-Agent: z.B. Überwachung bestimmter Server-Hardware (z.B. Compaq)
- Agents zur Überwachung von Zusatzfunktionen wie z.. USV (Unterbrechungsfreie Stromversorgung)

Netzwerküberwachung:

- Überwachung der angeschlossenen Netzwerke von den Servern aus (Managewise LANalyzer Agent, Problematisch bei Switches)
- Agents zur Überwachung von Hubs, Bridges, Switches, Routern (Herstellereigen)

Standards:

ISO: CMIP (Common Management Information Protocol)
CMOT (CMIP over TCP/IP)
IETF: SNMP (Simple Network Management Protocol)
SNMP II

SNMP:

Jeder Agent stellt eine Datenbank (MIB=Management Information Base) zur Verfügung, in der spezifische Daten des überwachten Gerätes festgehalten sind. Die Kommunikation zwischen Netzwerkmanagementstation und den Agents (genauer mit der MIB der Agents) erfolgt im Allgemeine über das Protokoll SNMP (Simple Network Management Protocol).

SNMP erlaubt die Fernsteuerung der überwachten Geräte (Veränderung der Werte in der MIB) sowie das Versenden von Fehlermeldungen (sogenannte Traps) vom Agent zur Managementstation. SNMP kann, auch wenn es aus der IP Welt kommt, in verschiedenen Protokollwelten verwendet werden, es kann insbesondere auch auf das Novell IPX Protokoll aufsetzen.

SNMP kennt 5 Befehle: Get Request, Get Response, Get Next, Set, Trap
(Objekt lesen, Tabellen lesen, Schreiben von Information, Alarmmeldung)

Beispiel für eine Kommunikation zwischen Konsole und Agent:

Get Request: Kosten Routerverbindung
Get Response: Kosten = 12
Set: Setze Kosten = 15

Aufbau der MIBs

Die Agents enthalten Information über das zu überwachende Gerät in Form sogenannter MIBs. Es gibt zwei verschiedene Typen von MIBs:

Standard MIBs (Internet MIBs): standardisierte MIBs, hier ist der Aufbau der MIB sowie der Parameter innerhalb der MIB genau festgelegt.

Enterprise MIBs: Hier kann jeder Hersteller für sein Gerät (z.B. ein Routerhersteller für seine Router) genau festlegen, welche Parameter des Geräts überwacht werden sollen, und welche Traps (Fehlermeldungen) in welchen Fällen an eine Management Station geschickt werden.

Die MIBs folgen einer einheitlichen, hierarchischen Struktur
(SMI: Structure and Identification of Management Information)

Generell muß natürlich auch die Managementstation über die Informationen der einzelnen Parameter der MIBs aller Agents verfügen, sonst könnten unter Umständen Traps, die ankommen, nicht richtig interpretiert werden. Speziell bei nicht standardisierten MIBs z.B. bestimmten Enterprise MIBs muß eventuell die MIB an der Managementstation entsprechend kompiliert werden, damit die entsprechende Information hier vorliegen kann. Bei NMS erfolgt dies über einen speziellen MIB Compiler, der innerhalb der Management Konsole gestartet werden kann.

Wesentliche Unterschiede zu anderen Netzwerkmanagementprotokollen:

SNMP am weitesten verbreitet

CMIP ist komplexer; wesentlich: abwärtskompatibel zu SNMP

CMIP Managementstation kann auf SNMP Komponenten zugreifen

SNMP II

Weiterentwicklung für erweiterte Funktionalität

Kommunikation zwischen Managementkonsolen

Erweiterte Sicherheitsfunktionen

Get Bulk: lesen kompletter Tabellen in einem Befehl ? Performancegewinn

Beispiele für Netzwerk-Management Software

- Novell anageWise

LAN, WAN

- IBM Tivoli
- HP Openview

LAN, WAN
LAN, WAN

Alarm-Management:

Für das Alarmmanagement müssen die entsprechenden Agents konfiguriert werden:

- Schwellwerte für Alarmmeldungen bei Überlastung
- Fehlerzustände

Damit dann entsprechende Trap-Meldungen bei einer Management Konsole auch ankommen, muß die Adresse dieser für jeden Agent konfiguriert werden (z.B. Traptarg.cfg bei Novell). Da nicht unbedingt gewährleistet ist, daß bei Ausfall einer Komponente überhaupt noch eine Fehlermeldung geschickt werden kann, macht es unter Umständen Sinn, zumindest Kritische Systeme dauerzuüberwachen (z.B. über Ping).

Zusätzlich können die Agents konfiguriert werden, so daß nicht alle möglichen Störfälle automatisch zum Senden eines Traps führen, sondern nur ganz bestimmte. Dazu kann normalerweise jeder einzelne Trap festgelegt werden, alternativ, sind die möglichen Störfälle in Klassen eingeteilt (Severity Levels, typisch Level 1 bis 7) und es kann festgelegt werden, daß z.B. nur Störfälle ab einem bestimmten Level wie Severity Level 6 geschickt werden. Auf diese Weise kann der durch die Managementfunktionalität verursachte Netzwerkverkehr reduziert werden, was sich speziell bei WAN Verbindungen auszahlt.

9.2 Desktopmanagement

Hier können die Clients eines Netzes überwacht werden. Dazu wird allerdings nicht auf SNMP aufgesetzt, sondern über ein eigenes Protokoll auf den Client zugegriffen.

Typische Aufgaben umfassen:

- Softwareverteilung und File-Transfer
- Inventarisierung
- Remote Control

Hier geht es also nicht um Alarmmanagement, sondern um Fernkonfiguration und Benutzerunterstützung. Aufgrund der zunehmenden Komplexität heutiger Netze und Workstationbetriebssysteme kommt dem Desktopmanagement eine immer größer werdende Bedeutung zu.

Beispiele für Desktopmanagementsysteme:

- Novell Z.E.N.Works
- Microsoft Systems Management Server (SMS)
- Netinstall

9.3 Fehlertoleranz bei LAN Betriebssystemen

Die Grundproblematik der Zentralisierung von Daten ist wie vorher bereits beschrieben die Ausfallsicherheit des Systems. Aus diesem Grund müssen Ausfallsicherungssysteme zur Verfügung stehen, um die Verfügbarkeit von Servern zu gewährleisten. Inwieweit ein solches System notwendig ist ergibt sich durch die Kosten, die bei einem Systemausfall entstehen können. Hierbei muß man durchaus unterscheiden zwischen

- ungeplantem Systemausfall und
- Wartungszeiten, für die ein System für eine gewisse Zeit heruntergefahren werden muß, um Wartungsarbeiten, Upgrades und ähnliches durchführen zu können.

Grundsätzlich muß festgestellt werden, daß eine **relativ hohe** Ausfallsicherheit durch heutige Standardsysteme gegeben ist, daß aber **sehr hohe** Ausfallsicherheit unter Umständen sehr teuer erkaufte werden muß. Hier müssen die Kosten, die durch einen Systemausfall entstehen können (z.B. durch entgangene Umsätze, Kosten von Arbeit usw) gegen die Kosten der Ausfallsicherungslösung gestellt werden. Sehr viele Firmen können wahrscheinlich

einen Systemausfall für mehrere Stunden oder sogar Tage verkraften, andere Firmen kann ein Systemausfall von nur einer Stunde in den Konkurs stürzen. Entsprechende Ausfallsicherungssysteme sind nötig.

9.3.1 Kosten eines Systemausfalls

Die folgende Tabelle zeigt das Ergebnis einer Kostenanalyse, die bei ausgewählten Firmen in verschiedenen industriellen Bereichen durchgeführt wurde (Quelle: Novell Brainshare 98 aus Dataquest Contingency Planning Research):

Industriebereich	Geschäftsbereich	Kostenbereich pro Std. in US\$	Durchschnittl. Kosten pro Std. in US\$
Finanzen	Wertpapierhandel	5 600 000 – 7 300 000	6 450 000
Finanzen	Kreditkartenorganisation	2 200 000 – 3 100 000	2 600 000
Medien	Pay-TV	67 000 – 233 000	150 000
Handel	Radio/TV/Hifi	87 000 – 140 000	113 000
Handel	Kataloganbieter	60 000 – 120 000	90 000
Transport	Fluglinie (Reservierung)	67 000 – 112 000	89 000
Medien	Ticketverkauf	56 000 – 82 000	69 000
Transport	Paketdienst	24 000 – 32 000	28 000
Finanzen	Bankautomatengebühren	12 000 – 17 000	14 000

Auch eine sehr hohe Verfügbarkeit schließt einen kurzzeitigen Systemausfall nicht aus. Die folgenden Zahlen verdeutlichen dies:

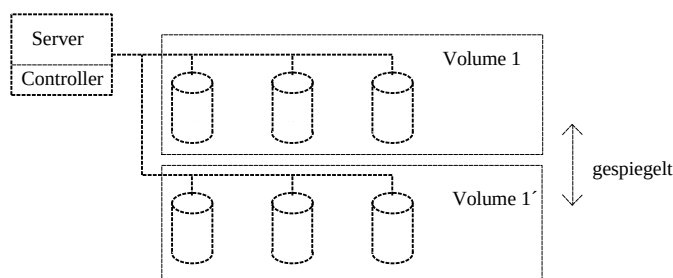
Verfügbarkeit	Ausfallzeit pro Jahr
96 %	350,4 Std.
98 %	175,2 Std
99 %	87,6 Std
99,9 %	8,7 Std
99,99 %	52,5 min
99,999 %	5,2 min

9.3.2 Realisierung von Ausfallsicherungssystemen

- ⚡ Redundanz der FAT, DET (Tabellen)
- ⚡ Festplattenspiegelung (1 Controller, mehrere Disks)
- ⚡ Festplattenduplexing (2 Controller, mehrere Disk)

Festplattenspiegelung und Duplexing wird oft kombiniert mit sogenanntem Disk-Striping (bei Novell: Volume Spanning)

Realisierung: Eine Partition wird über mehrere Platten verteilt. Dadurch überlappender Zugriff (quasi-parallel) möglich mit entsprechender Steigerung der Zugriffsgeschwindigkeit.



RAID

Verlegung des Disk Striping / Volume Spanning auf HW-Ebene

Erhöhung der Verfügbarkeit von Festplatten durch Verteilung auf möglichst viele Festplatten und Wiederherstellung von Daten bei Plattenausfall. Man unterscheidet verschiedene sog. RAID Level:

Level 1: Block Striping

Level 2: Bitstriping mit ECC

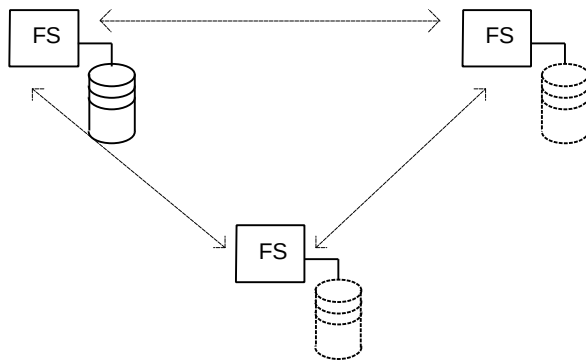
Level 3: Bitstriping mit Paritätsplatte

Level 4: Blockstriping mit separater Paritätsplatte

Level 5: Blockstriping mit verteilter Paritätssicherung

Datenspiegelung über verschiedene Server, eigentlich Shadowing bzw. Replication.

z.B. Replizierung von verteilten Datenbanken (z.B. Directory Services), Replizierung von Datenbereichen auf Festplatten (z.B. Novell Replication Service)

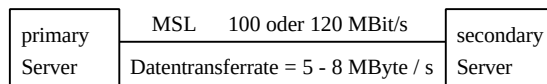


Server Spiegelung (derzeit nur Novell 3.11 SFT III bzw. 4.2 - SFT III)

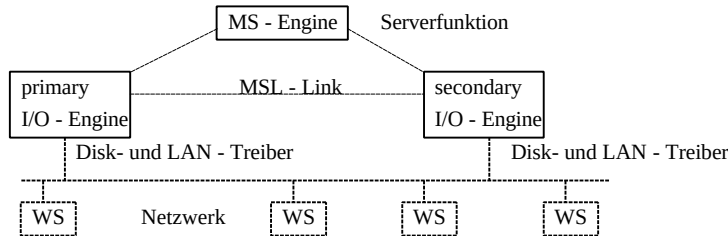
SFT III = System Fault Tolerance III:

Komplette Spiegelung zweier Server mit großen Teilen Ihrer Hauptspeicherbereiche, die Server präsentieren sich nach aussen als ein System. Die beiden Server sind über das LAN sowie eine dedizierte Verbindung (MSL: Mirrored Server Link) miteinander verbunden).

physikalischer Aufbau



logischer Aufbau



Kommunikation vom Client findet nur über die Primary I/O-Engine mit der MS-Engine statt, die Secondary I/O Engine wird über das MSL Link auf dem Laufenden gehalten. Im Fehlerfall des Primary Servers kann die Secondary Server die Verbindung zur MS-Engine herstellen. Um sicherzustellen, daß der Secondary Server unterscheiden kann, ob wirklich ein Ausfall des Primary Servers vorliegt oder ein Ausfall der MSL Verbindung, tauschen beide Server ständig über das LAN sogenannte I'm Alive Pakete aus.
Ausfallzeit: Konfigurierbar von 1s bis 10s.

Standby Server (Vina, Novell)

Ähnliches Konzept wie SFT III aber ohne Spiegelung der Speicherbereiche der Server. Stattdessen wird lediglich eine Festplattenspiegelung über eine dedizierte Verbindung zwischen den Servern durchgeführt. Bei Serverausfall wird der Standby Server automatisch aktiviert und übernimmt (transparent für die Benutzer) die Rolle des ersten Servers.

Hierbei kann ein einziger Standby Server die Backupfunktion für mehrere Server übernehmen.

Ausfallzeit: Gegeben durch Neustart des Standby Servers und Laden der Volumes einige Minuten)

Cluster

Ein Cluster ist eine lose Kopplung von Servern über schnelle Netzwerkverbindungen (SAN: Storage Area Network) mit gemeinsamem Speichersystem. Im Gegensatz zu den vorhergehenden Konzepten sind alle im Cluster beteiligten Server aktiv, fällt ein Server aus, übernimmt ein anderer Server aus dem Cluster die Funktionen des ersteren mit. Hierbei wird der Cluster dem Benutzer als ein einziger Server präsentiert „Single System Image“. Dies beinhaltet den Datenaustausch von Information im RAM der Server. Dadurch können Anwendungen (die für den Einsatz im Cluster entwickelt wurden) im Cluster sowie der Zugriff auf verschiedene Server verteilt werden (Load Balancing).

Ziele die durch Clustering erreicht werden sollen, sind:

- Erhöhung der Verfügbarkeit
- Erhöhung der Systemperformance
- Darstellung des Clusters von Servern als ein System (Single System Image)
- Skalierbarkeit
- Lastverteilung

Die Realisierung eines SANs kann bei heutigen Systemen erfolgen über:

- SCSI Verbindungen
- Fast Ethernet / Gigabit Ethernet (über Switch), ATM, FDDI (über eigenes Clustering Interconnect Protocol CICP)
- SCI (Scalable Coherent Interface, 200MB/s)
- Fibre Channel
- SSA (Serial Storage Architecture) (IBM)

Es gibt verschiedene Clusterkonzepte:

- Clustering von einzelnen Servern zur Realisierung einer gegenseitigen Ausfallsicherung von zwei Servern (Microsoft MS Cluster Server).
- Verteilter Cluster: (Novell Cluster Services) Hier ist eine Ausfallsicherung eines Servers durch mehrere Server im Cluster möglich (Novell Terminologie: Fan-Out Failover). Damit bietet dieser Cluster nicht nur Ausfallsicherung sondern gleichzeitig die Skalierbarkeit des Gesamtsystems.
- MultiSite Cluster: Cluster über mehrere Standorte hinweg

10. Besondere Aspekte von IP Infrastrukturen

10.1 Routing

10.1.1 Distanzvektorprotokolle

Distanzvektor (distance vector) Routingprotokolle zeichnen sich durch folgende Merkmale aus.

- Router erstellen aus Informationen die sie von anderen Routern erhalten, direkt eine Routingtabelle
- Die Routingtabelle wird vollständig in regelmäßigen Zeitabständen (RIP(IP) alle 30s, RIP(IPX) alle 60s im Netz per Broadcast bekannt gemacht.

Damit erhalten einzelne Router dieselbe Information von vielen verschiedenen Routern, auch wenn diese die Information selbst auch nur von anderen Routern erfahren haben (Information basiert sozusagen auf Hörensagen). Dies führt in größeren Netzen unter Umständen zu Konvergenzproblemen (unterschiedliche Routinginformation an verschiedenen Routern) und zum sog. Count to Infinity Problem, das sich wie nachfolgend dargestellt äußert.

Das Count to Infinity Problem lässt sich durch entsprechende Protokollergänzungen verhindern (beide RIP(IP) Versionen verwenden dazu Techniken wie Split Horizon bzw. Poison Reverse, die nachfolgend dargestellt sind).

10.1.2 Verbindungsstatusprotokolle

Das oben beschriebene Problem wird bei Verbindungsstatusprotokollen (Link State Protocols) wie folgt umgangen:

- Trennung von Routingtabelle von der Update-Information, die an andere Router geschickt wird
- Zentrale Verwaltung mehrerer benachbarter Router
- Updates nur bei Änderungen oder in langen Zeitintervallen

Der Unterschied zum Distance Vektor Routing wird im folgenden Bild klarer.

10.2 Planung der Netzwerkinfrastruktur

10.2.1 Subnetting

Obiges Beispiel zeigt, daß für getrennte Segmente auch getrennte Netzwerkanteile der Adressen nötig sind. Daher ist es z.B. schon aus administrativen Gesichtspunkten heraus sinnvoll, eine weitere Unterteilung einer Netzwerkadresse in weitere Teilnetze, sogenannte Subnetze, durchzuführen. Dies kann für Class A oder Class B Adressen z.B. so erfolgen, daß jeweils ein oder auch zwei Bytes zur Unterscheidung der Teilnetze herangezogen werden. Obiges Beispiel wäre also auch mit der Class B Adresse 162.170.0.0 möglich, wobei die Teilnetze 162.170.1.0, 162.170.2.0 und 162.170.3.0 heißen könnten.

Definiert würde das dann über eine Netzwerkmaske 255.255.255.0

Die Adresse 162 würde zunächst auf ein Class B Netz hindeuten, die Netzwerkmaske legt aber noch ein drittes Byte als Netzwerkadresse fest, diese dritte Byte wird dementsprechend zur Unterteilung der Subnetze herangezogen. Insgesamt könnten so 254 Subnetze in diesem Class B Netz gebildet werden.

Die Aufteilung der Adresse ergibt sich damit zu

	Binär:	Dezimal
Adresse:	10xxxxxx.xxxxxxxx. xxxxxxx. xxxxxxx	128-191.xxx.xxx.xxx
Netzwerkmaske:	11111111.11111111. 11111111. 00000000	255.255.255.0
	Netzwerk Subnet Host	

Grundsätzlich ist aber auch eine andere Netzwerkmaske möglich. Folgendes Beispiel zeigt Subnetting in einem Class C Netz:

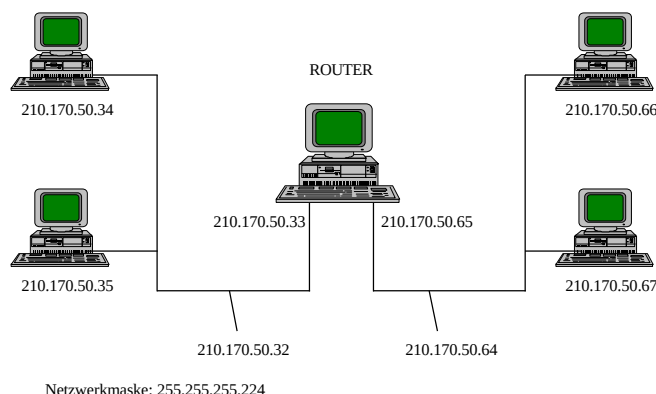
	Binär:	Dezimal
Adresse:	110xxxxx.xxxxxxxx.xxxxxxxx. xxx xxxxx	192-224.xxx.xxx.xxx
Netzwerkmaske:	11111111.11111111.11111111. 111 00000	255.255.255.224
	Netzwerk	Subnet Host

Die ersten 3 bit des letzten Byte stehen nun zur Unterscheidung der verschiedenen Subnetze zur Verfügung. Damit können 6 (2^3-2) Subnetze gebildet werden, jedes dieser Subnetze kann 30 (2^5-2) Hosts enthalten. Adressen, bei denen alle Bits 1 oder 0 sind, werden häufig nicht verwendet, da eine Adresse mit 1-Bits auch als Broadcast und eine Subnetz-Adresse mit 0-Bits auch als das "Hauptnetz" interpretiert werden kann. Dies hängt weitestgehend von den verwendeten Routern und Routingprotokollen ab (so überträgt RIP z.B. nur Netzwerkadressen ohne dazugehörige Netzwerkmasken, OSPF oder andere Routingprotokolle (oft Routerherstellerspezifische) dagegen übertragen beides, so daß man hier flexibler wird). Durch Subnetting gehen also eventuell Adressen verloren, und zwar umso mehr, je weniger Bits die Subnetzmaske umfaßt. Dies muß dann aber in Kauf genommen werden, wenn eine logische Trennung in Subnetze durch die Verwendung von bestimmten Routern bzw. Routingprotokollen erfolgen soll.

Die Nummern der Subnetze lauten nun am Beispiel der Class C Adresse 210.170.50.0:

Binär (letztes Byte)	Dezimal (vollständ. Adresse)	Mögliche Hostadressen
000 00000	210.170.50.0	i.A nicht verwendet
001 00000	210.170.50.32	210.170.50.33-62
010 00000	210.170.50.64	210.170.50.65-94
011 00000	210.170.50.96	210.170.50.97-126
100 00000	210.170.50.128	210.170.50.129-158
101 00000	210.170.50.160	210.170.50.161-190
110 00000	210.170.50.192	210.170.50.193-222
111 00000	210.170.50.224	i.A. nicht verwendet

Die folgende Abbildung zeigt nochmals unser erstes Beispiel, nun aber mit Subnetting.



Auch hier kann aus einer gegebenen IP Adresse der Netzwerkanteil wieder durch bitweise UND Verknüpfung gewonnen werden, wie folgendes Beispiel zeigt:

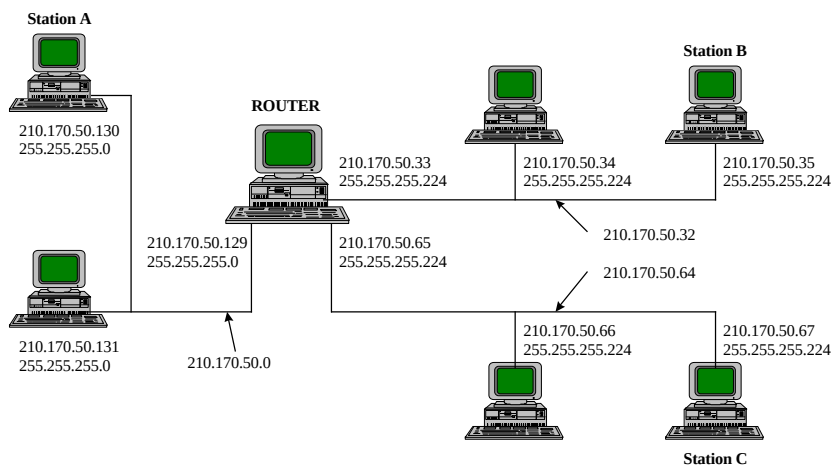
Besondere Aspekte von IP Infrastrukturen

Adresse:	10110111 . 01011110 . 00110011 . 01001101	183. 94. 51. 45
Maske:	<u>11111111 . 11111111 . 11111111 . 11100000</u>	<u>255.255.255.224</u>
Netzwerkanteil:	10110111 . 01011110 . 00110011 . 01000000	183. 94. 51. 32

10.2.2 Proxy ARP

Generell ist es auch möglich, den verschiedenen Netzwerksträngen unterschiedliche Netzwerkmasken zuzuteilen. So ist z.B. eine Aufteilung wie im folgenden Bild möglich. Hier werden die Subnetze 32 und 64 verwendet, alle anderen werden in einem Netzwerksegment zusammengefaßt.

Allerdings besteht hier die Problematik, daß Sender und Empfänger unterschiedliche Subnetzmasken besitzen. Damit gehen sie bei der Kommunikation von unterschiedlichen Voraussetzungen aus, was die Frage betrifft, ob sich ein Kommunikationspartner im lokalen oder einem entfernten Netzwerksegment befindet. Station B geht davon aus, daß sich Station A mit Adresse 130 im letzten Byte in einem anderen Subnetz als 64 befindet und schickt daher alle Pakete folgerichtig an den Router. Nur, Antwortpakete werden zunächst nicht empfangen. Station A geht nämlich aufgrund seiner Netzwerkmasken davon aus, daß sich alle Hosts mit den Adressen 1 bis 254 im letzten Byte im eigenen Segment befinden. Ein ARP Request für Station B oder C würde also zunächst keine Hardwareadresse liefern, womit diese Stationen zunächst unerreichbar wären.



Abhilfe schafft hier die Fähigkeit mancher Router, sogenanntes Proxy-ARP zu unterstützen. Ein Router, der Proxy-ARP unterstützt, ist in der Lage für alle Hosts in den angeschlossenen Subnetzen stellvertretend für diese Stationen ARP-Requests zu beantworten. Im konkreten Fall würde also der Router für Station B oder C seine eigene Hardwareadresse als die richtige angeben und sich dann um die Weiterleitung des entsprechenden Pakets kümmern. Proxy ARP wird von manchen UNIX-Routern, Hardwareroutern und Novell NetWare 4 Routern unterstützt, nicht jedoch von NetWare 3.

10.2.3 Supernetting

Supernetting erlaubt die Zusammenfassung von Netzwerkadressen in sog. Supernets. Hierzu können Gruppen von aufeinanderfolgenden Netzwerkadressen mit einer verkürzten Netzwerkmaske dargestellt werden.

Beispiel: Nehmen wir an, 4 Class C Netze (202.35.32.0 ... 35.0) sollen zu einem Supernet zusammengefasst werden. Die entsprechende Schreibweise wäre dann

Adresse: 202.35.32.0

Maske: 255.255.252.0

oder verkürzt geschrieben 202.35.32.0/22

Dies ermöglicht z.B. Routern, dieses Supernet mit einem einzigen Routingtabelleneintrag zu adressieren, sofern diese Router CIDR (Classless Interdomain Routing) unterstützen. Statt also hier vier Routingeinträge für jede der vier Netzwerkadressen zu haben, genügt nun einer. Dies ist speziell für ISPs (Internet Service Provider) interessant, die ganze Gruppen von Subnetzen für Ihre Kunden verwalten.

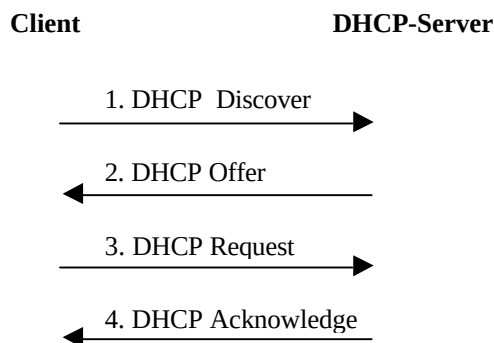
10.3 Dynamische IP-Adressvergabe über DHCP

In IP basierten Netzen ist es zunehmend üblich, IP Adressen dynamisch von einem zentralen Server an Clients im Netzwerk zu vergeben. Hierfür wurde zunächst BOOTP entwickelt, darauf aufsetzend wurde später das DHCP (Dynamic Host Configuration Protocol, RFC 1531) standardisiert, das heute weit verbreitet ist. DHCP unterstützt dabei drei verschiedene Arten der Adressvergabe:

- Manuelle Adressvergabe: Hier wird dem Client vom DHCP Server eine Adresse zugeteilt, die vorher von einem Netzwerkadministrator festgelegt wurde
- Permanente (oder auch automatische) Adressvergabe: Hier erhält der Client eine Adresse aus einem bestimmten Bereich, diese aber für unbegrenzte unbegrenzte Zeit
- Dynamische Adressvergabe: Hier erhält der Client eine Adresse aus einem bestimmten Bereich für eine begrenzte Zeit (sog. Lease Time)

10.3.1 Verfahren der Adressvergabe

Die Adressvergabe an sich läuft wie folgt in vier Schritten ab:



1. Der Client sucht zunächst per Broadcast (Quelladresse 0.0.0.0, Zieladresse 255.255.255.255) einen DHCP Server im lokalen Subnetz (Broadcasts werden nicht geroutet!). MAC Adresse und Computername (Netbios Name) werden übertragen.
2. Alle verfügbaren DHCP Server bieten eine freie IP Adresse an. (Mit Client MAC Adresse, Subnetzmaske, Leasedauer, Server IP Adresse). Antwortet kein DHCP Server, wiederholt der Client Schritt1 dreimal und dann alle 5 Minuten)
3. Der Client wählt sich aus den möglichen Adressen eine aus und bestätigt diese per Broadcast an den Server. Die Meldung ist gleichzeitig die Nachricht an mögliche andere DHCP Server, dass ihr Angebot abgelehnt wurde
4. Der DHCP Server bestätigt die Adresse und sendet zusätzliche Konfigurationsparameter für den Client. (s. später)

10.3.2 Ablauf der Lease Time

Bei begrenzter Lease Time der IP Adresse beginnt ein Client bei 50% der Zeit, eine Verlängerung der Lease Time bei „seinem“ DHCP Server zu beantragen (sog. DHCP Renew Request). Dies entspricht den Schritten 3 und 4 von oben. Ist er dabei nicht erfolgreich, fragt er wiederum nach der Hälfte der verbleibenden Zeit an. Für weitere Anfragen wird die restliche Zeit fortlaufend halbiert. Ab 87,5% der abgelaufenen Zeit beginnt der Client wieder den ursprünglichen Vorgang der Adressanfrage und ist dann bereit, jede Adresse von jedem beliebigen DHCP Server anzunehmen. Dies wiederholt er in weiterhin fortlaufend halbierten Intervallen, bis er entweder eine Adresse angeboten bekommt oder die Lease Time abgelaufen ist. In diesem Fall gibt er die Adresse auf und kann nicht mehr kommunizieren.

Microsoft Windows Rechner verhalten sich hier anders: nach Ablauf der Lease ohne Verlängerung suchen die Rechner sich selbst per Broadcast im Bereich 169.0.0.1 bis 169.254.255.254 (sog. APIPA Bereich, Automatic Private IP Addressing) eine freie Adresse selbst aus. Der APIPA Bereich ist von der IANA (Internet Assigned Numbers Authority) reserviert.

10.3.3 DHCP Frame Format

Nachfolgend ist das DHCP Frame gezeigt (Die Zahlen in Klammern geben die jeweilige Feldlänge in Byte an).

op (1)	htype (1)	hlen (1)	hops (1)
xid (4)			
secs (2)		flags (2)	
ciaddr (4)			
yiaddr (4)			
siaddr (4)			
giaddr (4)			
chaddr (16)			
sname (64)			
file (128)			
options (312)			

Hier die Bedeutung der einzelnen Felder:

Feld	Bytes	Beschreibung
op	1	Nachrichtentyp: 1 = BOOTREQUEST, 2 = BOOTREPLY
htype	1	Hardware Adresstyp (z.B. '1' = 10Mb Ethernet)
hlen	1	Hardware Adresslänge ('6' for 10Mb Ethernet)
hops	1	Client setzt den Wert auf 0, wird von sog. Relay Agents erhöht.
xid	4	Transaction ID. Eine zufällige Zahl, die für Client und Servernachrichten gleich ist.
secs	2	Zeit in Sekunden seit Start der Adressvergabe, durch den Client festgelegt
flags	2	Flags
ciaddr	4	Client IP Adresse. Vom Client vorgegeben, wenn dieser seine Adresse kennt (durch vorherige Adressanfragen)
yiaddr	4	'your' (client) IP Adresse. Die Antwort des Servers.
siaddr	4	Server IP Adresse..
giaddr	4	Relay Agent IP Adresse, verwendet bei Adressvergabe über Relay Agents
chaddr	16	Client Hardware Adresse.
sname	64	Optionaler Servername (Mit Null terminierter String)
file	128	Boot File Name (Mit Null terminierter String) "generic" name or null bei Requests, Vollqualifizierter Name des Directory Pfades bei Antwortframes
options	variab le	Optionale Parameter

Eine Liste der standardisierten Optionen findet sich im Anhang bzw. unter <http://www.iana.org/assignments/bootp-dhcp-parameters>. Für jede Option ist eine bestimmte Optionsnummer (Tag) festgelegt.

Beispiele für DHCP Options sind zusätzliche Konfigurationsparameter, wie z.B.

- 3 - Routeradresse
- 6 - DNS Server Adresse
- 85 - Novell NDS Server

und DHCP Kontrollmeldungen, wie z.B.

- Nachrichtentyp (das OP Code Feld, ein Überbleibsel von Bootp, ist hier zu unspezifisch)
- Parameter Request Liste (aktuell verwendete Optionen, die vom Client angefragt werden können).

Zudem gibt es einen Bereich TagNr: 128-254 von herstellerspezifischen Optionen, über die z.B. spezielle Clientkonfigurationsoptionen durchgeführt werden können. Microsoft verwendet dies z.B. zur Vergabe spezieller Parameter abhängig vom Rechnertyp (Notebook, Desktop etc.).

10.4 Multicast in IPNetzen

Grundsätzlich unterscheidet man bei der Kommunikation in Netzen:

- Unicast: Punkt zu Punkt Übertragung zwischen Netzwerkknoten.
- Broadcast: Rundruf an alle Teilnehmer im Netz.
- Multicast: Verbindung zu einer Gruppe ausgewählter Knoten innerhalb eines Netzes
- Anycast: Verbindung zu irgendeinem Knoten innerhalb einer Gruppe von Knoten

Gerade im Bereich Multimedia Anwendungen (Streaming bzw. Conferencing) spielt Multicast eine große Rolle.

Aufgabe: Verteilung von Daten über geeignete Router Infrastruktur an lokale Multicast Router, von dort an alle Clients einer Multicast Gruppe.

Typische Infrastruktur:

10.4.1 Multicast Adressierung

Adressbereich für IP- Multicast: 224.0.0.0 - 239.255.255.255

Spezielle Adressen sind u.A.:

Adresse	Bedeutung
224.0.0.1	All Host Group
224.0.0.2	All Router Group
224.0.0.4	All DMVRP Routers
224.0.0.5	All OSPF Routers
224.0.0.6	All OSPF Designated Routers
224.0.0.9	All RIP2 Routers
224.0.1.x	Versch. Anwendungsspez. Multicasts

DMVRP, OSPF, RIP sind sogenannte Routingprotokolle. Mehr dazu später.

Adresszuordnung:

Die Idee des Multicastings ist es, ein Paket ins Netz zu senden und im Netzwerk allen Knoten mit einer entsprechenden Multicast IP Adresse zuzuleiten. Dasselbe gilt für verschiedene Clients innerhalb eines Direktverbindungsnetzes. Insofern muß Multicast auch auf der MAC Ebene funktionieren. Dazu wird die IP-Multicast-Adresse der Rechner auf eine MAC Multicast Adresse abgebildet. Jeder Rechner speichert also neben seinen Unicast IP und MAC Adressen auch seine IP Multicast und MAC Multicast Adressen.

Multicast IP Adresse:

		1110 xxxx	x			
		9 bits		23 bit		
MAC Adresse						
0000 0001	0000 0000	0101 1110	0			
Organisation Unique ID :		01-00-5E				

Prinzipiell könnten Adressüberschneidungen auftreten, da nur 23 bit der MAC Adresse verwendet werden.

10.4.2 Aufbau der Infrastruktur

Registrieren von Clients:

IGMP: Internet Group Message Protocol

- Multicast-Router (M-Router) sendet in festen Zeitintervallen (100s) eine Membership Query an "All Hosts" mit einer TTL von 1 (d.h. an direkte Nachbarn)
- Clients antworten mit einem Membership Report (Beinhaltet Mitteilung, zu welcher Multicast Gruppe sie gehören)

Hinzufügen eines Clients: Client sendet MAC Multicast mit einem IGMP Membership Request

Verlassen einer Gruppe: IGMP Leave Group

Die Buchführung über aktuelle Verbindungen zu Clients liegt beim Router. Der Router teilt übergeordneten Routern nur die Änderung kompletter Multicast Gruppen mit, nicht jedoch wenn einzelne Clients hinzukommen bzw. die Multicast Gruppe verlassen.

11. Aufgaben bei der Paketübertragung

Die Paketübertragung in Netzen ist bestimmten Netzwerkeigenschaften unterworfen. Die wichtigsten sind:

- Latenzzeit: Zeit zur Übertragung von Paketen zwischen Sender und Empfänger
- Jitter: Pakete die vom Sender in gleichem Abstand geschickt werden, kommen beim Empfänger in unterschiedlichen Abständen an.
- Paketverluste: Pakete können unterwegs verloren gehen. GGf. muß ein Empfänger in der Lage sein, den Sender zur erneuten Übertragung aufzufordern
- Fehlerhafte Übertragung: Pakete können bei der Übertragung verfälscht werden
- Änderung der Paketreihenfolge: Pakete können im Netzwerk auf unterschiedlichen Pfaden übertragen werden und beim Empfänger in einer anderen Reihenfolge ankommen als abgesendet
- Bandbreite: Die Übertragungsbandbreite ist begrenzt, Sender und Empfänger müssen über entsprechende Mechanismen zur sog. Überlastkontrolle verfügen.

11.1 Fehlererkennung

Bei der Übertragung von Paketen können sog. Bitfehler (einzelne Bits werden verändert) oder sog. Burstfehler (Gruppen von Bits werden verändert) auftreten. Diese müssen vom Empfänger eines Pakets erkannt werden können. Zur Fehlererkennung gibt es entsprechende Möglichkeiten:

- Parität
- Zweidimensionale Parität
- Prüfsummen (engl. Checksum, Algorithmus der auf Addition basiert, Verwendung z.B. bei den Protokollen IP, UDP, TCP)
- Zyklische Redundanzprüfung (CRC, Cyclic Redundancy Check, Algorithmus basiert auf Polynomrechnungen, eine der am häufigsten verwendeten Technologien z.B. bei HDLC, Ethernet, Token Ring etc.)

Bei Erkennung eines Fehlers wird ein erhaltenes Paket üblicherweise verworfen und der Sender aufgefordert, das Paket noch einmal zu senden. Dies kostet natürlich zusätzliche Bandbreite, diese ist aber nur bei Auftreten eines Fehlers tatsächlich erforderlich.

Alternativ dazu gibt es Codes zur Fehlerkorrektur. Diese finden jedoch im LAN Bereich kaum Verwendung. Ein Grund dafür ist, dass bei Verwendung von Fehlerkorrekturcodes insgesamt eine größere Anzahl redundanter Bits gesendet werden müsste, was zusätzliche Bandbreite erfordern würde. In Protokollen für Leitungsgebundene Übertragung sind Bitfehler aber nicht so wahrscheinlich wie z.B. bei der nichtleitungsgebundenen Kommunikation.

11.2 Zuverlässige Übertragung

Je nach Dienst kommen in Netzwerken unterschiedliche Kommunikationsmuster zum Tragen. Bei den meisten Diensten besteht das Kommunikationsmuster aus Abfragen, die beantwortet werden.

Ein anderes Kommunikationsmuster besteht aus einem kontinuierlichen Datenstrom (z.B. bei Videoanwendungen). Je nachdem kann also die Übertragung mehr oder weniger Zuverlässig sein.

Um Pakete zuverlässig zustellen zu können, müssen fehlerhafte Pakete erkannt und verworfen werden können und der Sender aufgefordert werden, die betreffenden Pakete nochmals zu senden. Hierfür werden Bestätigungen verwendet, wobei es zwei Verfahren gibt:

- Der Empfänger quittiert erfolgreich gelesene Frames (Acknowledgment ACK), der Sender sendet Frames erneut, wenn das ACK innerhalb einer bestimmten Zeit (Timeout) nicht ankommt. Dieses Verfahren wird auch als ARQ (Automatic Repeat Request) bezeichnet.
- Der Empfänger quittiert fehlerlose und fehlerbehaftete Frames (letztere mit einem negative Acknowledgement (NAK), was dann zur erneuten Übertragung führt.

Bestätigungen können als kleine Kontrollframes gesendet werden, oder an andere Datenframes (im Hückepackverfahren engl. Piggyback) angehängt werden.

11.2.1 Stop-And-Wait Algorithmus

Hier wartet der Sender auf ein ACK, nach einem Timeout überträgt er den Frame erneut.

Das Grundproblem bei dieser Art der Übertragung besteht darin, dass die Übertragungsleitung nicht voll ausgelastet wird, da der Sender immer zunächst auf eine Bestätigung warten muß, bevor er den nächsten Frame sendet.

Zudem kann es natürlich vorkommen, dass der Empfänger einen zweiten Frame erhält obwohl er den ersten bereits bestätigt hat (aufgrund von Zeitüberschneidungen, oder weil der Ack Frame verloren ging). Dies wird so gelöst, dass der Header beim Stop and Wait Protokoll ein Sequenzbit bei jedem neuen Frame zwischen 0 und 1 ändert. Erhält der Empfänger zwei Frames mit gleichem Sequenzbit, weiss er, dass der Sendevorgang wiederholt wurde.

Die Details zeigt nachfolgendes Bild

11.2.2 Sliding Windows Algorithmus

Hier werden mehrere Frames gesendet, bevor der Sender ein Ack vom Empfänger erwartet. Im Gegensatz zum Stop-And-Wait Algorithmus ist hier der Übertragungskanal besser ausgelastet. Hierzu werden zunächst alle Pakete mit einer Sequenznummer versehen. 1) beschreibt den Algorithmus wie folgt:

Der Sender verwaltet:

- Die Größe des Sendefensters (Send Windows Size, SWS)
- Die letzte Empfangsbestätigung (Last Ack Received, LAR)
- Den zuletzt gesendeten Frame (Last Frame Sent, LFS)

Der Sender hält zudem die Bedingung $LFS - LAR \leq SWS$ aufrecht

Der Empfänger verwaltet

- Die Größe des Empfangsfensters (Receive Windows Size, RWS)
- Die größte anzunehmende Sequenznummer (LargestAcceptable Frame, LAF)
- Die Nummer des zuletzt empfangenen Frames (Last Frame Received, LFR)

Zudem hält er die Bedingung $LAF - LFR \leq RWS$ aufrecht

Kommt beim Sender eine Bestätigung an, schiebt der Sender sein LAR auf einen höheren Wert und ermöglicht damit dem Sender, neue Frames zu übertragen.

Kommt beim Empfänger ein Frame an, schaut er nach, ob die Sequenznummer im akzeptablen Bereich liegt, wenn ja, akzeptiert er ihn.

Dann entscheidet er, ob er eine Bestätigung sendet. Hier gibt es mehrere Implementationen:

- Bestätigung des Frames dieser Sequenznummer sowie aller darunterliegenden, wenn sie bereits angekommen sind, und erhöht seine LFR und damit LAF. Sollte in der Reihenfolge noch ein Frame fehlen, speichert er die Frames zwischen, bis er eine Sequenz von aufeinanderfolgenden Frames bestätigen kann.
- Selektive Bestätigung jedes einzelnen Frames mit der entsprechenden Sequenznummer

Eine Einschränkung besteht darin, dass Sequenznummern nicht unendlich groß sein können und sich damit im Laufe einer Übertragung wiederholen können. Mehr dazu unter 1).

Manche Protokolle verwenden den Sliding Windows Algorithmus mit einer festeingestellten Fenstergröße (z.B. SDLC). Damit kann der Algorithmus zunächst für folgende Aufgaben benutzt werden:

- die Zuverlässige Übertragung von Paketen sowie
- die Einhaltung der Reihenfolge von Frames

Andere Protokolle können die Größe des Empfangsfensters während der Übertragung verändern und dies dem Sender bekannt machen (z.B. TCP). Damit kann

- die Flusskontrolle zwischen Empfänger und Sender

mit übernommen werden, d.h. der Empfänger teilt dem Sender mit, für wie viele Frames (bzw. Bytes bei TCP) er momentan jeweils Platz hat.

11.2.3 Übertragung über mehrere logische Kanäle

Eine Alternative stellt die Verwendung gleichzeitiger logischer Kanäle bei der Paketübertragung dar. Hier kann in mehreren Kanälen einer der obigen Mechanismen verwendet werden. Bei Verwendung des Stop-And-Wait Algorithmus auf jedem dieser Kanäle kann dann wieder eine bessere Auslastung des Übertragungskanals erreicht werden (Beispiele: Sicherungsschicht im ARPANET, Novell SPX).

Aufgaben bei der Paketübertragung

11.3 Flußkontrolle bei Ende zu Ende Übertragung

Neben dem Verbindungsaufbau sind bei der Ende zu Ende Übertragung folgende grundsätzliche Fragen zu klären:

- Je nach Entfernung der Kommunikationspartner sind unterschiedliche Round Trip Times (RTT) im Netzwerk zu erwarten
- Die Paketreihenfolge muß kontrolliert und ggf. beim Empfänger korrigiert werden
- Es müssen Mechanismen zur Flusskontrolle vorhanden sein, d.h. der Empfangspufferplatz muß bestimmbar sein
- Eine maximale Übertragungsbandbreite soll genutzt werden können, andererseits muß eine Überlastung des Netzes verhindert werden.

Die verwendeten Mechanismen werden im Folgenden am Beispiel von TCP erklärt.

11.3.1 Berücksichtigung der Round Trip Time (RTT)

11.3.1.1 TCP Originalalgorithmus zur adaptiven Neuübertragung

Bei jedem Sendevorgang wird eine Bestimmung der aktuellen RTT durchgeführt. Der Algorithmus war:

Sample_RTT = Zeitpunkt Empfang ACK – Sendezeitpunkt eines Segments

Estimated_RTT = $\alpha \times \text{Estimated_RTT} + (1-\alpha) \times \text{Sample_RTT}$

mit α Glättungsfaktor, in ursprünglicher TCP Spec: 0,8 - 0,9

Timeout = $2 \times \text{Estimated_RTT}$

Der Timeout dient dazu, dass Pakete, die nicht innerhalb der Zeit des Timeouts bestätigt werden, erneut übertragen werden. Zur besseren Auslastung des Netzes wurden später weitere Algorithmen entwickelt.

11.3.1.2 Karn/Partridge

Aussetzen der Messung bei Neuübertragungen. Problemlösung für: 2. ACK bei Neuübertragung führt zu falscher Sample RTT in obiger Formel.

11.3.1.3 Jacobsen/Karels

Wesentliches Ziel des Algorithmus war die Vermeidung von Überlastsituationen (s. später). Hierfür wurde durch eine genauere Bestimmung des Timeouts nötig. Wenn der Timeout zu klein ist, muß ein Segment evtl unnötigerweise nochmal übertragen werden, ist er zu hoch, sinkt die Übertragungsrate. Ein Timeout kann als Zeichen einer Netzwerküberlast gesehen werden (Eine Quelle sendet ein Paket $\rightarrow$ ein Router unterwegs verwirft das Paket $\rightarrow$ Timeout tritt auf).

Ziel des Algorithmus: Bei großen Schwankungen des Sample_RTT, sollte die Sample_RTT nicht zu Nahe bei der Estimated_RTT liegen. Bei geringen Schwankungen ist aber ein Timeout der 2 fachen Estimated_RTT zu hoch.

Insgesamt muß die Quelle ermitteln können, wie viel Kapazität im Netzwerk frei ist. Dazu wurde ein Congestion Window (Sendefenster) eingeführt, das über den Mechanismus „Additive Increase/Multiplicative Decrease“ gesteuert wird:

- Wird ein Paket erfolgreich gesendet: Vergrößerung der Größe eines Pakets
- Tritt ein Timeout aus: Halbierung des Sendefensters

Später kamen noch zwei weitere Verfeinerungen hinzu:

- Slow Start: Exponentielle Erhöhung des Sendefensters, um die langsame additive Vergrößerung des Sendefensters in der Startphase einer Verbindung zu vermeiden
- Fast Retransmit: Senden von (doppelten) ACKs auch dann, wenn ein Paket außerhalb der Reihenfolge ankommt. Empfängt der Sender drei gleiche ACKs, sendet er das dazugehörige Segment nochmals, auch wenn der Timeout noch nicht abgelaufen ist.

12. Echtzeitübertragung in Netzen

12.1 Übertragung von Echtzeitdaten

12.1.1 RTP (Real-Time Transport Protocol)

Das RTP Protokoll wird verwendet zur Übertragung von Echtzeitdaten (Video bzw. Audio).

Architekturprinzip: Application Level Framing:

Anstatt auf die entsprechenden Mechanismen von TCP zuzugreifen, segmentiert hier die Anwendung die Daten selbst und kümmert sich auch um verlorene Pakete. Wie das jeweils geschieht, hängt von der Anwendung ab.

Zwei grundsätzliche Anwendungsgebiete:

- Konferenzanwendungen: Hier sind Anforderungen an möglichst geringe Latenz hoch, dafür kann die Bildqualität gering sein
- Streaming: Hier ist die Latenz zweitrangig, dafür bestehen höhere Anforderungen an die Qualität der übertragenen Daten.

RTP ist als sehr einfaches, aber universell einsetzbares Protokoll konzipiert. Unterschiedliche Anwendungen können durch verschiedene RTP Profile für unterschiedliche Daten charakterisiert werden.

V	P	X	CC	M	Payload Type	Sequence Nr			
Time Stamp									
SSRC-ID				Synchronization Source					
CSRC-ID				Contributing Sources					
...									
...									

V:	Version, derzeit 2
P:	Füllbits vorhanden (verweist darauf dass am Ende des UDP Pakets eine Anzahl von Füllbits festgelegt wurde, z.B. um eine bestimmte UDP Paketlänge zu erreichen)
X:	Erweiterungsheader vorhanden
CC:	Anzahl beitragender Quellen (nur bei Einsatz eines Mixers)
M:	Marker Bit (Framekennzeichnung), stellt zusammen mit dem
PT:	Payload Type ein bestimmtes Anwendungsprofil dar
Sequence Nr:	Erkennung fehlender oder falsch angeordneter Pakete
Time Stamp:	Zeitstempel = Zeit zwischen Paketen / Sampling Intervall
SSRC-ID:	Eindeutige Kennzeichnung einer Quelle des RTP Stroms (wird von der Quelle zufällig gewählt, ist unabhängig von der IP-Adresse und erlaubt damit mehrere Quellen pro Knoten)
CSRC-ID:	Kennzeichnungen weiterer Quellen in einem gemischten Datenstrom

12.1.2 RTCP (RTP Control Protocol)

Neben dem RTP Protokoll wird für jede Übertragung ein eigener Kontrollkanal über das RTCP Protokoll aufgebaut.

Aufgaben:

- Überwachung der Dienstqualität (QoS) über sog. Empfänger Reports (Rückmeldung über z.B. Qualitätsverlust oder Jitter etc. an den Sender)
- Überlastkontrolle: Der Sender informiert den Empfänger mittels sog. Sender Reports über die zu erwartende Datenrate
- Zusätzliche ID der Quelle: Hier wird ein kanonischer Name (CNAME) verwendet, um eine Quelle eindeutig zu identifizieren. Leibt im Gegensatz zur SSRC-ID immer fest
- Zusätzliche Information über eine RTP Sitzung (z.B. Name eines Konferenzteilnehmers)

12.2 Sitzungskontrolle und Signalisierung

Echtzeitübertragung in Netzen

RTP und RTCP verfügen nicht über Möglichkeiten zum Auf bzw. Abbau von Sessions bzw. zur Signalisierung. Hier wird auf andere Protokolle zurückgegriffen, wobei es hier zwei Ansätze gibt:

12.2.1 H.323

Empfehlung der ITU für die Signalisierung für Multimediakommunikation in Paketnetzen

H.323 definiert RTP zur Echtzeitübertragung, zudem werden Standards für Audiocodecs (G.711 bzw. weitere) und Videocodecs (H.261 bzw. weitere) festgelegt.

Zur Signalisierung werden u.A. folgende Protokolle definiert:

- H.225:
Hierzu wird eine Kommunikationszentrale eingerichtet (Gatekeeper), der unter anderem die IP Adressen in Telefonnummern der Clients abbildet. Ein H.323 Client (bezeichnet als H.323 Terminal) kann eine Verbindung nur nach einer Registrierung beim Gatekeeper aufbauen. Die benötigte Bandbreite der Verbindung wird über das sog. RSVP (Resource Reservation Protocol) beim Gatekeeper reserviert. Im wesentlichen umfasst H.225 zwei Protokolle
 - o Q.931: Call Signalisierung
 - o RAS (Registration Admission, Status) Steuerung.

- H.245: wird nach dem Verbindungsaufbau verwendet, um die Clients Eigenschaften und Charakteristika der Sitzung auszuhandeln
- H.235: Sicherheitsaspekte von H.323 Terminals (z.B. Verschlüsselung)
- H.450.x: Verschiedene Standards (.x) für zusätzliche Services in H.323 (Call transfer, Call diversion, Call hold, Call park & pickup etc.)
- T.120 Serie: Protokolle für Multimedia Conferencing

H.323 umfasst weitere Protokolle für verschiedenste Aufgaben im Bereich VoIP.

12.2.2 IETF Protokolle

Die SIP Architektur sieht auf den ersten Blick ähnlich wie die H.323 Architektur aus.. Calls können wie folgt aufgebaut werden:

- zwischen Clients direkt
- zwischen Clients unter Verwendung eines sogenannten SIP Proxy Servers.

Mögliche Standorte eines Clients werden durch einen sog. Location Server (oder Registrar) gespeichert, ein Standortwechsel eines Zielclients kann bei direktem Verbindungsaufbau von einem sog. Redirect Server, bei Verbindung über einen SIP Proxy vom Proxy Server beim Location Server angefragt werden.

Verwendete Protokolle innerhalb der SIP Architektur sind:

- SIP: Session Initiation Protocol (in etwa äquivalent zur Empfehlung H.225 Protokoll (d.h. Q931 und RAS) der H.323 Architektur)
- SDP: Session Description Protocol (Stellt Information für die Session zur Verfügung : Übertragungsmedium (Codecs, Sampling Rate), Addressinformation, Session Name und Zeiten, Kontaktinfo etc.). Damit ist SDP eigentlich eher ein Datenformat als ein Protokoll (vergleichbar mit ASN.1 bei H.323).

Echtzeitübertragung in Netzen

Für Videoconferencing kommen zwei weitere Protokolle hinzu:

- SAP: Session Announcement Protocol (Wird verwendet um Conference Calls in einer Multicast Umgebung anzukündigen, basiert auf Multicast, daher nicht geeignet für 1 zu 1 IP-Verbindungen)
- SCCP: Simple Conference Control Protocol (Management von Multimedia Sessions)

Mehr Information zur SIP Architektur findet sich unter www.sipcenter.com bzw. www.sipforum.org .

13. Ipng (IP next generation, IPv6)

Die größten Probleme des IP Protokolls in der derzeitigen Version 4 sind der begrenzte Adressraum und, für die Verwendung im Internet, das Fehlen von Verschlüsselungsalgorithmen. Beide Probleme werden in der nächsten Version IP Version 6 bzw. Ipng adressiert. Ipng wurde als Nachfolger des jetzigen IP von der IETF im Juli 1994 verabschiedet.

In der neuen IP Version sollten u.A folgende weitere Anforderungen gelöst werden:

- Unterstützung von Echtzeitsdiensten
- Unterstützung von Sicherheitsmerkmalen
- Automatische Konfiguration
- Erweiterte Routing Funktionalität

Diese Merkmale fehlten zu Beginn der Ipv6 Entwicklung, sind inzwischen aber in der Ipv4 Welt verfügbar. Dies mag ein Grund sein, warum die Umstellung auf Ipv6 langsamer vonstetten geht als zunächst angenommen wurde.

13.1 IPv6 Paketformat

Trotz der Erweiterungen, die IPv6 mit sich bringt ist das Paketformat einfacher als bei IPv4:

0	4	12	16	24	31
Version	Traffic Class	Flow Label			
Payload Length			Next Header	Hop Limit	
Source Address					
Destination Address					
Next Header / Data					

Die Felder bedeuten im Einzelnen:

- Traffic Class/ Flow Label: Unterstützt einen Mechanismus zur Dienstgüteunterstützung (Quality of Service QoS)
- Payload Length: Paketlänge des IP Pakets ohne Header
- Next Header: Ersetzt die Felder IP-Options (z.B. zur Fragmentierung von Paketen) und das Protocol Feld des Ipv4 Headers. Wenn Optionen erforderlich sind, wird kurzerhand ein weiterer Header nach dem Destination Address Feld hinzugefügt, in dem die Erweiterungen dann behandelt werden. Dies können zunächst beliebige Header, die zu Beginn wiederum ein Feld Next Header enthalten müssen. Beispiele sind ein eader zur Fragmentierung oder ein spezieller Routing Header.

13.2 Adressierung

Bei IPv6 wird zwischen folgenden Adress-Kategorien unterschieden:

- Unicast
- Multicast
- Anycast

Anycast Adressen ermöglichen den Versand von Paketen über eine festgelegte Stelle an eine Gruppe von Stationen. Im Gegensatz zu Multicast Paketen, die an alle Knoten einer bestimmten Multicast Adresse zugestellt

werden, werden Pakete an eine Anycast Adresse zunächst nur einem der Knoten mit der entsprechenden Anycast Adresse zugestellt.

Broadcast Adressen gibt es nicht, sie werden durch Multicasts an eine „All Nodes“ Gruppe ersetzt.

Der Adressraum von IpnG ist 16Byte. Bei einer 100% Ausnutzung von Adressen sind so $3,4 \times 10^{38}$ Adressen verfügbar. Selbst wenn man 100% Adresszuweisung nicht erreichen wird hat man aufgrund empirischer Zahlen aus französischen und US Amerikanischen Telefonnetzen sowie bei IPv4 ermittelt, dass pro Quadratmeter Erdoberfläche ca 15000 Adressen zur Verfügung stehen (Nach Peterson/Davie: Computernetze).

Die Schreibweise wird von einer byteweise Schreibweise mit „.“ Trennung in eine wortweise Schreibweise mit „:“ Trennung verändert.

Beispiel für eine IPv6 Adresse:

2E00:0000:0000:0000:0001:00E4:1011

Diese Adresse kann kompakter geschrieben werden. Hierzu kann man führenden Nullen weglassen und mehrere aufeinanderfolgende Null-Werte mit „:“ abkürzen. Damit ergibt sich aus obiger Adresse z.B.

2E00::1:E4:1011

Die Abkürzung „:“ darf nur einmal in der Adresse verwendet werden. Eine Adresse

::AB45:1234:: als 0:0:0: AB45:1234:0:0:0 wäre nicht eindeutig.

IPv6 unterscheidet nicht nach festen Adressklassen, sondern unterteilt den Adressraum flexibler: Die allg. Architektur der Adressierung findet sich in RFC2373. Der Adressraum ist danach wie folgt aufgeteilt:

Zuordnung	Präfix	Adressanteil
Reserved	0000 0000	1 / 256
Unassigned	0000 0001	1 / 256
Reserved (NSAP Allocation)*	0000 001	1 / 128
Reserved (IPX Allocation)*	0000 010	1 / 128
Unassigned	0000 011	1 / 128
Unassigned	0000 1	1 / 32
Unassigned	0001	1 / 16
Aggregateable Global Unicast	001	1 / 8
Unassigned	010	1 / 8
Unassigned	011	1 / 8
Unassigned	100	1 / 8
Unassigned	101	1 / 8
Unassigned	110	1 / 8
Unassigned	1110	1 / 16
Unassigned	1111 0	1 / 32
Unassigned	1111 10	1 / 64
Unassigned	1111 110	1 / 128
Unassigned	1111 1110 0	1 / 512
Link Local Unicast	1111 1110 10	1 / 1024
Site Local Unicast	1111 1110 11	1 / 1024
Multicast	1111 1111	1 / 256

Die mit * markierten Adressen dienen der Einbettung anderer Adresstypen (OSI bzw. Novell IPX Adressen)

13.2.1 Aggregatable Global Unicast Addresses

Diese Adressen sind die Adressen für die „normale“ Nutzung von Punkt zu Punkt Verbindungen. Der Adressraum wird hierarchisch aufgeteilt, nachdem davon ausgegangen wird, dass die gesamte Internetwelt organisatorisch gesehen hierarchisch aufgebaut ist. Folgendes Bild zeigt die Aufteilung:

3 Bits	13 Bits	8 Bits	32 Bits	16 Bits	64 Bits			
001	TLA-ID	Res.	NLA-ID	SLA-ID	OID	xFF	xFE	Ext. ID
	Öffentliche Struktur			Private Struktur	Interface-ID			

- TLA-ID (Top Level Aggregation ID): Identifikation Internationaler bzw. Nationaler Organisationen wie z.B. Anbieter von Internet Diensten
- NLA-ID (Next Level Aggregation ID): Identifikation erlaubt weitere Strukturierung der Hierarchie
- SLA-ID (Site Level Aggregation ID): Identifikation eines Kunden (Endbenutzers). Kann weiter hierarchisch strukturiert werden für Subnetz Hierarchie innerhalb eines privaten Netzes
- Interface ID: Physikalische Adresse eines Interfaces in einem Endsystem. Um bei einer lokalen Adressierung auf ARP verzichten zu können, wird die MAC Adresse hier eingebettet. (aufgeteilt in OID organisation Unique ID des Herstellers - 1111 1111 1111 1110 – Hersteller Selected Extension ID).

Unicast Adressen von lokaler Bedeutung

Link Local Unicast Adressen sind Adressen, die nur innerhalb eines isolierten Subnetzes verwendet werden und von Routern nicht weitergeleitet werden.

Präfix 10Bits	70 Bits	48 Bits
1111 1110 10	0000.....0	MAC Adresse

Site Local Unicast Adressen sind strukturierte Adressen, die ein Subnetting innerhalb eines solchen isolierten Netzes erlauben

Präfix 10Bits	54 Bits	16 Bits	48 Bits
1111 1110 11	0000.....0	Subnet ID	MAC Adresse

13.2.2 Spezielle Unicast Adressen

Unspezifizierte Adresse

- 0:0:0:0:0:0:0:0 oder kurz „::“
ist die Absenderadresse eines Systems, dem noch keine IPv6 Adresse zugewiesen wurde, z.B. bei DHCP Requests

Loopback Adresse

- 0:0:0:0:0:0:0:1 oder kurz „::1“

IPv4 kompatible Adressen

Grundsätzlich gibt es zwei Möglichkeiten, um eine Kompatibilität der IPv4 und IPv6 Adressen zu ermöglichen. Eine besteht darin, die IPv4 Adressen in IPv6 einzubetten und parallel zu verwenden. Hierzu werden die IPv4 Adressen einfach mit führenden Nullen aufgefüllt. Aus 210.70.25.86 wird damit 0:0:0:0:0:210.70.25.86 bzw. ::210.70.25.86

IPv4 mapped IPv6 Adressen:

Diese dienen der Übersetzung von Adressen in Fällen, in denen die IPv4 Kommunikation durch ein IPv6 Netzwerk getunnelt wird. Hierzu wird ein IPv4 Frame in einen IPv6 Header automatisch eingepackt, wobei der IPv4 Anteil der Adresse 210.70.25.86 hier 0:0:0:0:FFFF:210.70.25.86 bzw. ::FFFF:210.70.25.86 wird. Die durch die Ende-zu-Ende Kommunikation vorgegebenen Quell bzw. Zieladressen werden damit in einem dazwischen befindlichen IPv6 Netz auf entsprechende IPv6 Adressen gemapped.

IPv6 Kommunikation durch ein IPv4 Netz:

Im umgekehrten Fall einer IPv6 Kommunikation durch ein IPv4 Netz werden die IPv6 Adressen im Datenfeld eines IPv4 Frames huckepack transportiert, wobei dann in den Routern, die den Tunnelanfang bzw. Tunnelendpunkt repräsentieren, über entsprechende Adressumsetzungstabellen eine echte Adressumsetzung von der Ziel-IPv6 auf die IPv4 Adresse des Tunnelendpunkts durchgeführt wird.

13.3 Autokonfiguraion

Neben der klassischen automatischen Adressierung via DHCP, das für IPv6 ebenfalls verfügbar ist und als zustandsbehaftete Autokonfiguration (stateful Autoconfiguration) bezeichnet wird, unterstützt IPv6 sogenannte zustandslose Autokonfiguration (stateless Autoconfiguration). Diese geschieht in zwei Schritten:

- Bekanntmachen der eigenen Link Adresse und Überprüfen ihrer Einmaligkeit (normalerweise die eigene MAC Adresse) als ICMP Request an eine spezielle Multicast Adresse.
- Einholen des korrekten Adress Präfixes für dieses Subnetz vom nächsten Router. Diese Nachricht wird als ICMP Request an die „All Routers“ Multicast Adresse

13.4 ICMPv6

ICMPv6 ist etwas umfangreicher als sein Vorgänger aus der IPv4 Welt. Hier kommen neben Fehlermeldungen und Echo Requests noch Group Membership Messages (für Multicast) und die Unterstützung der automatischen zustandslosen Adresskonfiguration hinzu.

14. Übertragungsmedien für Netze

14.1 Leitungsgebundene Medien

- Koax - Verkabelung
 - 50 ? - Ethernet, Thickwire - Yellow Cable
 - 50 ? - Cheapernet, Thinner, Thinwire Ethernet
- Twisted Pair:

			Wellenwiderstand	Leitungswiderstand
IBM-LAN	Typ 1	STP	150 ? / 4 MHz	50 m? / m
IBM Host-Kanal	Typ 2	STP	150 ? / 4 MHz	120 m? / m
IBM Patchkabel	Typ 3	UTP	84 - 113 ? / 1 MHz	85 m? / m

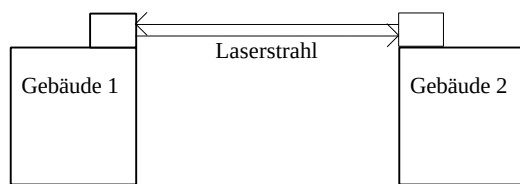
UTP Kategorien (Levels)

Cat 1	Telefonkabel	< 1 MHz
Cat 2	Datenkommunikation	< 4 MHz
Cat 3	Datenkommunikation	< 16 MHz
Cat 4	Datenkommunikation	< 20 MHz
Cat 5	Datenkommunikation	< 100 MHz
Cat 6	Datenkommunikation	< 300 MHz

- Glasfaser (Monomode-, Multimodefasern)
- Einsatz bei:FDDI, auch Ethernet, Token Ring

14.2 Nicht Leitungsgebundene Medien

- Richtfunk
- Satellitenübertragung
- Funkübertragung (LAN)
- Infrarotübertragung
- Laserstrecke

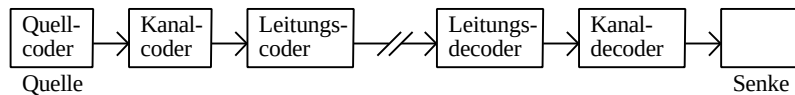


15. Grundlagen und Standards für Direktverbindungsnetze

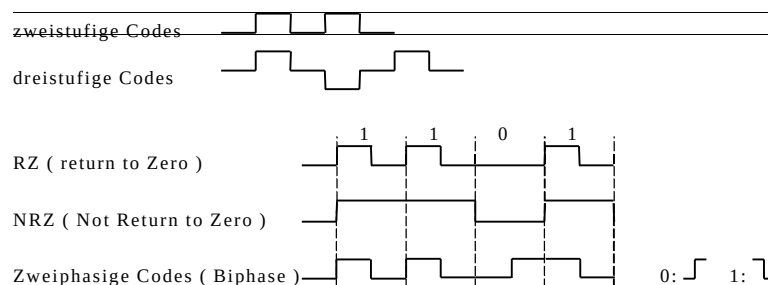
Während Punkt zu Punkt Übertragung vor allem in vermittelten Netzwerken (Switched LANs) zustande kommt, wurden die meisten LAN Zugriffsprotokolle zunächst für Mehrpunktverbindungen geschaffen. Netze, in denen der gemeinsame Medienzugriff durch Einsatz entsprechender Zugriffsprotokolle gesteuert wird, werden als sogenannte „shared LANs“ bezeichnet.

15.1 Leitungscodierung:

Ein typisches Übertragungssystem sieht wie folgt aus. Nachrichten einer Quelle werden zunächst quellcodiert. Zur sicheren Übertragung über einen Übertragungskanal werden Codes zur Fehlererkennung bzw. -Korrektur verwendet, durch die zusätzliche Redundanz hinzugefügt wird.



Zur Übertragung digitaler Signale ist eine Leitungscodierung notwendig, die festlegt, wie bestimmte Folgen von 0 und 1 physikalisch durch bestimmte Spannungspegel repräsentiert werden. Dabei unterscheidet man:



Es gibt zwei grundsätzlich unterschiedliche Arten der Leitungscodierung

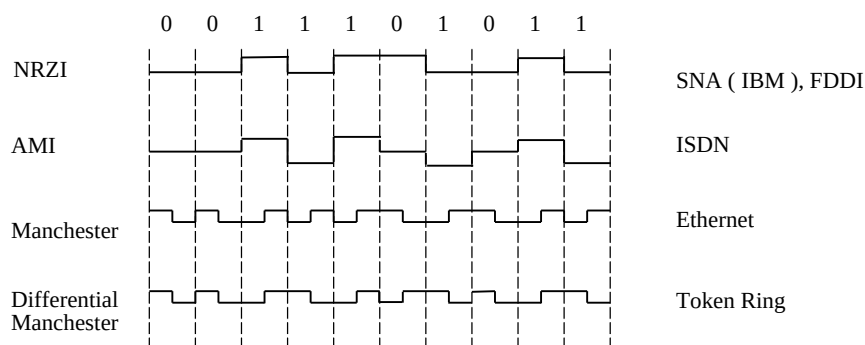
- Codierung der einzelnen Bits

Hierbei wird jedes Bit codiert, z.B. NRZ: 0=0V, 1= 5V

- Codierung der Übergänge zweier aufeinanderfolgender Bits

Hierbei wird z.B. eine 0 durch Wiederverwendung des vorangegangenen Symbols, eine 1 als Umkehrung des vorangegangenen Symbols codiert, z.B NRZI: 0 Signal bleibt konstant, 1: Signal ändert sich

Die meisten der gängigen Leitungscodes sind aus Kombinationen dieser grundsätzlichen Verfahren entstanden. Speziell im Netzwerkbereich verwendete Leitungscodes sind z.B:



NRZ z.B. 0 = 0V 1 = 5V

NRZI z.B. 0 = konst. 1 = Pegeländerung

Es gibt folgende Bewertungskriterien für Leitungscodes:

- Synchronisierbarkeit
- Bandbreitenausnutzung

- Gleichspannungsfreiheit
- Störempfindlichkeit
- Implementierungsaufwand

Anpassung des Kanalcodes an den Leitungscode

Gerade bei Verwendung von Codes, die eine geringe Bandbreiteausnutzung haben, aber damit evtl. auch bei langen Folgen von gleichen Bits nicht genügend Flanken im Signal zur Taktrückgewinnung bereitstellen (wie z.B. NRZ, NRZI oder andere) wird häufig die zu übertragende Bitfolge umcodiert, um ein Signal mit genügend Flanken zu erhalten.

Beispiel: Verwendung 4B/5B Codierung mit NRZI bei FDDI:

Hier werden jeweils Halbbytes durch 5 bit Worte ersetzt und zwar so, daß den 16 möglichen Halbbytes 16 geeignete 5bit Codeworte (aus 32 Möglichkeiten) zugeordnet werden. Geeignete Codeworte sind solche, die bei der Leitungscodierung genügend viele Signalfanken beinhalten.

Ein anderes Beispiel ist die Verwendung eines 3B/4B Codes beim ISDN.

15.2 Frameerzeugung

15.2.1 Asynchrone Übertragung

Zeichenweise Informationsübertragung (Start - Stop Verfahren). Kein kontinuierlicher Datenstrom zwischen Sender und Empfänger.

Start	Zeichen	Parity	Stop
-------	---------	--------	------

Problematik: 20% - 30% Overhead

15.2.2 Synchrone Übertragung

Kontinuierlicher Datenstrom zwischen Sender und Empfänger.

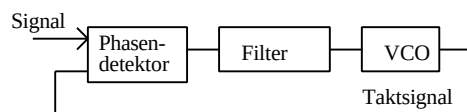
Zeichenorientierte Übertragung (verwendet z.B. bei BSC - Binary Synchronous Control)

ältere Host Systeme - heute kaum mehr

Bitorientierte Übertragung (verwendet bei SDLC, HDLC, LAP-B)

Synchronisierung Sender / Empfänger durch ein Taktsignal im Datenstrom

Empfänger: Taktrückgewinnung - Prinzip PLL



Notwendig zur Taktrückgewinnung:

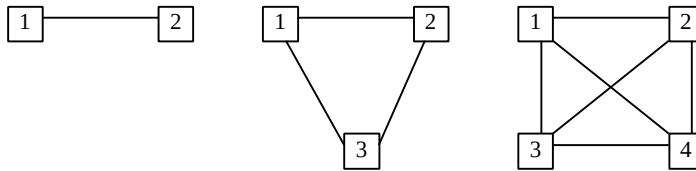
genügend Flanken im Signal ? Leitungs - Code

15.3 Physikalische Netztopologien

Netze werden heute in folgenden physikalischen Topologien verkabelt.

Maschentopologie (In öffentlichen Netzen verwendet)

Punkt zu Punkt Verbindungen zwischen Rechnern bzw.
zwischen Rechnern und Peripherie

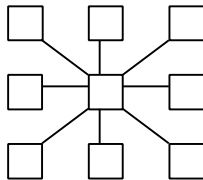


Sterntopologie (Basis für eine sog. Strukturierte Verkabelung, heute sehr häufig in LANs verwendet)

Punkt zu Punkt Verbindungen mit einem zentralen Punkt

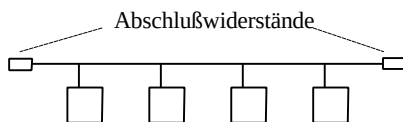
(HUB, Repeater, Concentrator, MSAU)

MSAU - Multiple Station Access Unit - Ringleitungsverteiler (Token Ring)



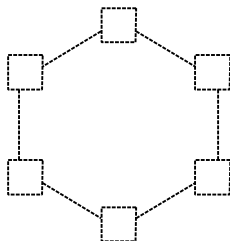
Bustopologie

Multipunktverbindung zwischen verschiedenen Knoten



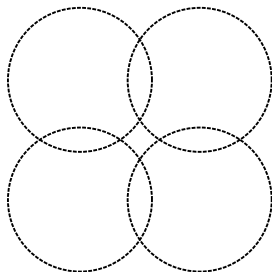
Ringtopologie

Punkt zu Punkt Verbindungen zwischen Knoten, die als Zwischenverstärker das Signal regenerieren können (Repeater).



Bei Funknetzen kommt eine weitere Topologie hinzu, die

Zelltopologie



Meistens findet man in heutigen Netzen verschiedene der vorgestellten Topologien gleichzeitig. Die verschiedenen Standards für Medienzugriffsprotokolle legen i.A: auch die zu verwendete Topologie fest (z.B. spezifiziert IEEE 802.3 10Base T Ethernet eine Sterntopologie). Die typischen Konfigurationsregeln (z.B. Längen) sind durch die entsprechenden Protokolle ebenfalls vorgegeben.

15.4 Physikalischer Aufbau von Netzen

Heute werden Netze üblicherweise in einer hierarchische Struktur aufgebaut, bestehend aus zwei oder auch drei Ebenen:

Ebene 1: Backbone (Firmenweites Netzwerksegment). Hier wird eine häufig Netzwerktopologien verwendet, die hohe Übertragungsraten erlauben (z.B. Token Ring, FDDI, zunehmend auch Switches (s. später)).

Ebene 2: Stockwerksverkabelung (fällt manchmal mit dem Backbone zusammen)

Ebene 3: Abteilungsverkabelung (getrennter Aufbau von Netzwerksegmenten abteilungsweit. Trennung von der Stockwerkssegmenten oder Backbone durch Geräte zur Lasttrennung (Router, Bridges/Switches s. später).

15.5 Multiplexing

Werden in einem Netzwerk mehrere Kommunikationskanäle geschaffen, muß durch Multiplexing eine Mehrfachnutzung ermöglicht werden. Man unterscheidet grundsätzlich zwei Übertragungsmethoden:

- Breitbandübertragung
auf verschiedenen Frequenzen verschiedene Information übertragen
- Basisbandübertragung
ohne Träger - Signal benötigt die gesamte Bandbreite

In lokalen Netzen findet man vor allem die Basisbandübertragung, die Übertragungsstrecke wird zeitlich auf verschiedenen Kanäle aufgeteilt (Zeitmultiplex).

Insgesamt gibt es folgende Varianten des Multiplexings:

Zeitmultiplex - Time Division Multiplex – TDM

Zeitliche Verschachtelung digitaler Signale auf einem Übertragungsmedium

- Synchron: feste zeitliche Zuordnung von Kanälen (z.B. im ISDN verwendet)
- Asynchron: Variable Aufteilung, Verwendung von Zellen (z.B. ATM)

Frequenzmultiplex - Frequency Division Multiplex – FDM

Übertragung mehrere analoger Signale auf einem Übertragungsmedium in versch. Frequenzlagen

Wellenlängenmultiplex – Wavelength Division Multiplex – WDM

Übertragung versch. Informationskanäle bei verschiedenen Wellenlängen gleichzeitig in einem Lichtwellenleiter.

- Typische Bandbreiten: 96 Kanäle a 10Gbit/s, Nortel Networks: 160 Kanäle a 10Gbit/s,
- Testphase: 40Gbit/s pro Wellenlänge, Labor: 80Gbit/s(Stand April 2000)

15.6 Medienzugriffsverfahren

15.6.1 Logische Topologien

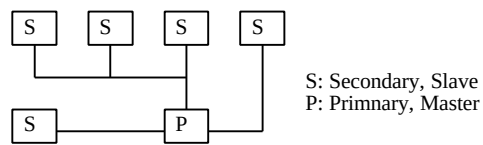
Unabhängig von der physikalischen Topologie eines Netzes gibt es verschiedene Zugriffsverfahren, um zu steuern, wie Informationen auf das Medium gebracht werden können. Hierbei gibt es im Grunde drei Möglichkeiten:

- ☞ Steuerung des Medienzugriffs von einem zentralen Punkt (Polling). Nachdem hier ein zentraler Kontrollpunkt existiert, der logische Verbindungen zu jedem Teilnehmer aufbauen muß, um den Medienzugriff zu regeln, spricht man hier von einer logischen Sterntopologie.
- ☞ Konkurrenz der Teilnehmer untereinander um den Medienzugriff (Konkurrenzverfahren). Hier ist jeder Teilnehmer gleichberechtigt, man spricht von einer logischen Bustopologie.
- ☞ Steuerung des Medienzugriffs durch Weitergabe einer Sendeberechtigung von Teilnehmer zu Teilnehmer, d.h. in einem logischen Ring.

Es ist damit möglich, ein Netzwerk in physikalischer Sterntopologie aufzubauen (heute häufig eine sternförmige Verkabelung mit UTP Cat. 5 oder 6), und dann je nach verwendetem Zugriffsprotokoll ein logisches Busnetz (z.B. Ethernet) oder ein logisches Ringnetz (z.B. Token Ring) zu implementieren. Allerdings muß dann die Verkabelung hinsichtlich physikalischer Voraussetzungen (z.B. Leitungslängen) für die entsprechenden Zugriffsprotokolle verwendbar sein.

15.6.2 Polling (Logische Sterntopologie)

Grundlagen und Standards für Direktverbindungsnetze



zentrale Zugriffssteuerung

Vorteil: Deterministisches Verfahren (Zeitscheibenverfahren)

Nachteil: Overhead

Anwendung: Host - Umgebung z.B. IBM

15.6.3 Konkurrenzsystem (Contention, logische Bustopologie)

Grundregel: Teilnehmer können bei Bedarf Daten sofort senden ? es können Kollisionen auftreten.

Abhilfe gegen Kollisionen:

- Trägererkennung CSMA (Carrier Sense Multiple Access)

- Kollisionserkennung CD (Collision Detection)

? CSMA / CD

Nach Kollision wird eine zufällige Zeitspanne gewartet

abhängig von Anzahl bisher erfolgloser Versuche

Kollisionserkennung durch Spannungsüberhöhung auf der Leitung (analoger Prozess)

- Kollisionsvermeidung CA (Collision Avoidance)

? CSMA / CA

kurzes RTS (Request to Send) Paket wird zum Empfänger geschickt und quittiert

dann erst Datenübertragung

Vorteile: Einfaches Protokoll, geringer Overhead

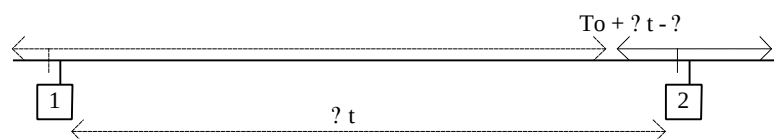
Nachteile: nicht deterministisch (Antwortzeiten nicht kalkulierbar, keine Prioritäten möglich)

Anwendung:

CSMA / CD ? Ethernet, IEEE802.3

CSMA / CA ? LocalTalk (Apple), IEEE 802.11

Problematik Kollisionserkennung



To: Station 1 sendet

To + ?t - ? : Station 2 sieht das Paket gerade noch nicht und sendet ebenfalls

? Kollision

To + 2?t : Sender erfährt von Kollision

2?t = Konkurrenzperiode

erst nach dieser Zeit kann der Sender sicher sein, daß keine Kollision erfolgt ist

Ausbreitungsgeschwindigkeit typ. 200m / μ s

Bei 500 m Länge: 2?t = 5 μ s

Sender sendet in 2?t: ?B = 50 Bit

?B? bei größeren Leitungslängen

?B? bei größeren Übertragungsraten

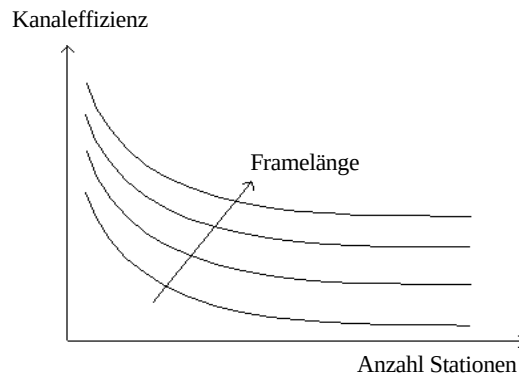
Grundlagen und Standards für Direktverbindungsnetze

Kanaleffizienz: [Tannenbaum, Kap. 3.4]

Vorraussetzung: konst. Wahrscheinlichkeit für nochmalige Übertragung

$$K_{eff} = \frac{1}{1 + 2 \frac{Ble}{cF}}$$

B = Bandbreite
c = Ausbreitungsgeschwindigkeit
l = Länge der Leitung
e = nat. Zahl
F = Framelänge



Das CSMA Verfahren ist skalierbar, d.h. bei höheren Datenraten kann man durch Verringerung der Leitungslänge oder durch Erhöhung der minimalen Paketgröße die Konkurrenzperiode konstant halten.

15.6.4 Token Passing (logische Ringtopologie)

Ein spezielles Bitmuster (Token) wird als Sendeberechtigung von Station zu Station weitergereicht. Das Token kann von jeder Station durch einen Frame ersetzt werden. Der Frame wird durch das Netz zum Empfänger geschickt, dieser quittiert den Erhalt und sendet den Frame zurück zum Empfänger. Dieser nimmt den Frame vom Netz. Der Sender kann nun ein weiteres Paket senden, oder er gibt den Token wieder frei. Eine Station behält den Token, d.h. die Sendeberechtigung nur für eine bestimmte Zeit, die Tokenhaltezeit, danach muß sie den Token auf jeden Fall weitergeben..

Vorteile:

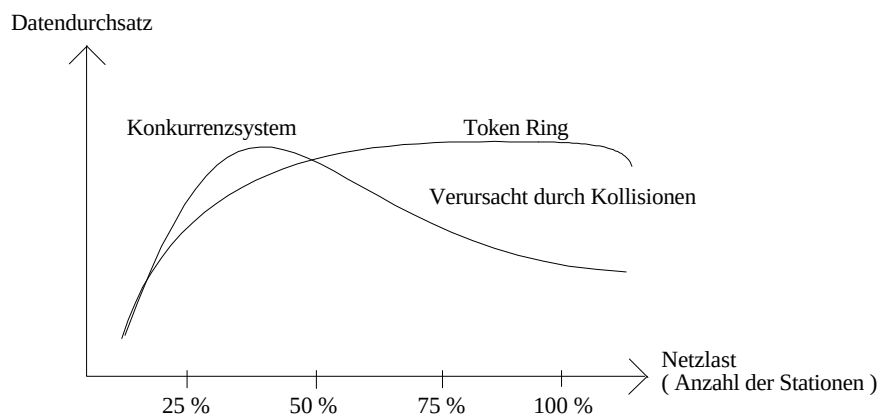
- ☒ Deterministisches Verfahren (d.h. Antwortzeiten kalkulierbar),
- ☒ Eine Prioritätensteuerung ist möglich, d.h. es können wichtigere und unwichtigere Stationen definiert werden.

Nachteil:

- ☒ Protokollaufwand hoch, z.B. muß eine Station im Netz Aufgaben der Ringwartung übernehmen (wie Erzeugung von Tokens, Verhinderung endlos kreisender Pakete, Neustart wenn Token verlorengeht usw.)

Anwendung: Token Ring, Arcnet, Token Bus (IEEE 802.4), FDDI

15.6.5 Vergleich Token Passing / Konkurrenzsystem



Konkurrenzsystm: Protokoll-Limitierung

Token Passing: Limitierung durch evtl. Netzwerkhardware

Token Passing verträgt höhere Netzlast

15.7 IEEE Projektgruppen

Im Bereich der Data Link Layer hat die IEEE (das Institute of Electrical and Electronical Engineers) die Standardisierung im wesentlichen beeinflusst. Sie unterteilt sich in verschiedenen Projektgruppen, die sich den verschiedenen Themen in diesem Bereich annehmen.

Network	IEEE 802.1 Internetworking		
LLC	802.2 LLC		
MAC	802.3 (Ethernet)	802.4 (Token Bus)	802.5 (Token Ring)
PHYS.			

Weitere IEEE 802 Projektgruppen sind folgende (nicht vollständig):

802.6	MAN
802.7	Breitbandtechnologie
802.8	Fiber Optics
802.9	Integration Sprache - Daten
802.10	LAN Sicherheit
802.11	Wireless LAN
802.12	Demand Priority Netze
802.15	Wireless PAN

15.8 IEEE 802.3 und Ethernet

15.8.1 Entwicklung des Standards

Ethernet ist sich zunächst die klassische Variante eines physikalischen Busnetzes und wurde von Digital, Intel und Xerox (DIX Standard) entwickelt, unter IEEE 802.3 werden eine ganze Reihe von Varianten von CSMA/CD Netzen unter dem auch dafür üblicherweise verwendeten Begriff "Ethernet" zusammengefaßt. Diese Varianten unterstützen verschiedene physikalische Topologien und Übertragungsraten. Die gängigsten sind:

Standard	Verkabelung
10 Base 5	Koaxverkabelung mit 500 m Segmentlänge (Thickwire)
10 Base 2	Koaxverkabelung mit 185 m Segmentlänge (Thinwire)
10 Base T	Twisted Pair
10 Base F	Fiber Optic 2 strangig (FB = synchron, FL = asynchron)
100 Base TX	Twisted Pair bidirektional, 4 adrig, UTP Cat5, STP Typ 1
100 Base T4	Twisted Pair 8 adrig (2x4), UTP Cat 3 und höher
1000 Base TX	Twisted Pair bidirektional , 4 adrig

Bezeichnungen:

~~10~~10, 100, 1000: Übertragungsrate in Mbit/s
~~Base~~Base: Basisbandübertragung
~~T~~T: Twisted Pair

Die klassische Ethernet Spezifikation und die IEEE 802.3 Standards unterscheiden sich zudem bei üblichen Bezeichnungen für die Geräte:



AUI = Attachment Unit Interface
 MAU = Medium Attachment Unit

Die Twisted Pair Varianten des IEEE802.3 Standards 10 Base T bzw. 100BaseT verwenden eine Verkabelung mit 2 Adernpaaren. In den üblicherweise verwendeten RJ45 Anschlußsteckern werden hierzu für die beiden Adernpaare die Pins 1/2 sowie 3/6 verwendet. Üblicherweise wurden jedoch beim Verlegen von UTP 5 Kabel alle 4 Adernpaare angeschlossen, insofern konnte man sich bei der Entwicklung der 1000 Base T und 10Gbase T Standards darauf stützen und alle 4 Adernpaare verwenden.

Ein sehr weitgehender weiterer Unterschied ist die Verwendung unterschiedlicher Frameformate. IEEE 802.3 kennt neben dem im klassischen Ethernet verwendeten Frameformat Ethernet_II noch drei weitere verschiedene Frameformate. Im folgenden sind die Formate gegenübergestellt:

Ethernet_II

Präambel	Ziel-adresse	Quell-adresse	Typ	Daten	CRC
----------	--------------	---------------	-----	-------	-----

Ethernet_802.3 Rohformat

Präambel	Ziel-adresse	Quell-adresse	Länge	Daten	CRC
----------	--------------	---------------	-------	-------	-----

Ethernet 802.2 (eigentlich IEEE802.3 und IEEE802.2)

Präambel	Ziel-adresse	Quell-adresse	Länge	LLC	Daten	CRC
----------	--------------	---------------	-------	-----	-------	-----

Ethernet_SNAP (Subnetwork Addressing Protocol)

Präambel	Ziel-adresse	Quell-adresse	Länge	LLC	Typ	Daten	CRC
----------	--------------	---------------	-------	-----	-----	-------	-----

8 Byte 6 Byte 6 Byte 2 Byte 46 – 1500 Byte 4 Byte

Die Präambel dient durch die Bitfolge 10101010 pro Byte im wesentlichen zur Synchronisation bei der Übertragung (Bei Ethernet II handelt es sich um 7 Byte und einen SFD (Start of Frame Delimiter) Frame, der die Bitfolge 10101011 besitzt, und den Beginn des Frames markiert). Ohne Präambel muß der Frame, um auf eine minimale Paketlänge von 64 Byte (min. Paketlänge für die Kollisionserkennung) zu kommen, evtl mit bis zu 46 sogenannten Füllbytes aufgefüllt werden. Die maximale Paketlänge ergibt sich zu 1518 Byte (manchmal findet sich auch die Angabe 1514 Byte, hier wird dann die CRC nicht mitgerechnet).

Der wesentliche Unterschied liegt zunächst in der Verwendung eines Längenfeldes statt eines Typ Feldes. Das Typ Feld verweist auf das darüberliegende Protokoll (üblicherweise IP oder IPX), bei Verwendung des Längenfeldes wird üblicherweise über LLC die Adressierung des darüberliegenden Protokolls vorgenommen (s. später). Aufgrund dieser unterschiedlichen Adressierung des nächsthöheren Protokolls können die verschiedenen Frameformate jeweils nur für unterschiedliche Netzwerkprotokolle verwendet werden, da die Netzwerkprotokolle unterschiedliche Adressierungsarten von der Data Link Layer her unterstützen. Die nachfolgende Tabelle zeigt, welche Frameformate für welche gängigen Netzwerkprotokolle verwendet werden können.

Frameformate	Gängigste Netzwerkprotokolle
Ethernet_II	IPX, TCP/IP, NetBeui
Ethernet_802.3	IPX, NetBeui
Ethernet_802.2	IPX, SNA, NetBeui
Ethernet_SNAP	IPX, TCP/IP, AppleTalk, NetBeui

15.8.2 CSMA/CD Zugriffsverfahren

Alle der vorgestellten Ethernet Varianten verwenden das CSMA/CD Zugriffsprotokoll. Nachdem jeder sendewillige Sender sofort (d.h. mit einer Wahrscheinlichkeit von 1) sendet, wenn der Kanal frei ist, wird diese Variante des CSMA Protokolls auch als *1-persistent CSMA* Protokoll bezeichnet (Ja, demnach gibt es auch CSMA Varianten, bei denen die Wahrscheinlichkeit kleiner 1 ist, die Stationen also freiwillig warten, bevor sie senden *non-persistent CSMA*).

Die maximale Ausdehnung eines klassischen 10Mbit/s Ethernets ergibt sich wie folgt:

- 5 Segmente a 500m maximal können über
- 4 Repeater miteinander verbunden werden, insgesamt dürfen
- 3 Segmente mit Workstations belegt sein (sog. 5-4-3 Regel).
- Die maximale Menge von Workstations im Ethernet ist 1000.
-

Damit die Kollisionserkennung auch bei diesen großen Ausdehnungen noch funktionieren kann, wird die minimale Paketlänge auf 64 Byte (bzw. 512 Bit) festgelegt. Dies führt zu einer Konkurrenzperiode (Round Trip Delay) von 51,2 μ s (worin die mögliche Latenzzeit bei der Übertragung über die Repeater berücksichtigt ist).

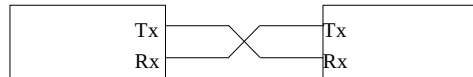
Sobald eine Netzwerkkarte erkennt, dass ihr Frame mit einem anderen kollidiert, bricht sie die Übertragung ab und sendet stattdessen ein Warnsignal (12 Byte: 8 Byte Präambel und 4Byte Warnsignal), engl. auch als Runt-Frame bezeichnet.

Nach einer Kollision beginnt ein Verfahren, das als exponential Backoff bezeichnet wird. Dazu wird die Zeit in Zeitschlitze (Timeslots) von 51,2 μ s eingeteilt, beide an der Kollision beteiligten Stationen warten eine bestimmte Zeit (0 μ s oder 51,2 μ s zufällig gewählt). Bei einer erneuten Kollision warten die Adapter 0 μ s, 51,2 μ s, 102,4 μ s oder 153,6 μ s (zufällig gewählt). Nach weiteren Kollisionen warten sie per Zufallsentscheidung $(0 \dots 2^n - 1) \times 51,2\mu s$, wobei n die Anzahl der erneuten Versuche ist. Wenn 10 Kollisionen erreicht sind, bleibt n konstant. Nach 16 Versuchen gibt der Netzwerkadapter die Übertragung auf und meldet einen Fehler.

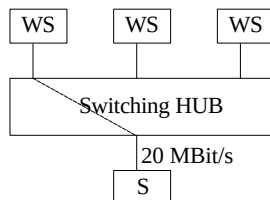
Wie früher bereits erwähnt, ist das Verfahren skalierbar, d.h. es kann auch bei höheren Übertragungsraten und entweder durch Vergrößerung der minimalen Paketlänge (Vergrößerung der Zeitschlitze) oder Verkürzung der Leitungsausdehnung weiter verwendet werden.

15.8.3 Full Duplex Ethernet

Beim Full Duplex Ethernet werden die beiden Adernpaare kurzerhand als Sende und Empfangskanal verwendet, wodurch nur Punkt zu Punkt Verbindungen möglich sind. Auf die klassische Ethernet Zugriffskontrolle wird hier verzichtet.



Die Idee ist, dass damit Ein- und Ausgang parallel arbeiten können, wodurch eine Verdopplung der Übertragungsraten erreicht werden soll (dies funktioniert jedoch nur bei der Kommunikation zwischen Switches und Servern (s. Bild), und auch dann nur, wenn man davon ausgehen kann, dass durch die von den verschiedenen Clients aus aufgebauten Verbindungen eine Gleichverteilung der Auslastung von Sende und Empfangskanal besteht. Üblicherweise sind die meisten Netzwerkumgebungen leseintensive Umgebungen, d.h. hier fließen im wesentlichen Nutzpakete zum Client und kleine Kontrollpakete zurück. Durch diese ungleichmäßige Aufteilung kann dann auch keine Verdopplung der Übertragungsrate erreicht werden.



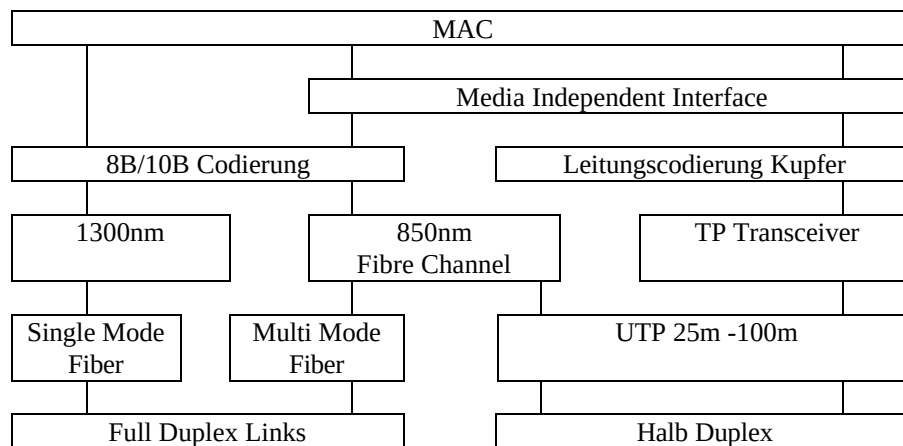
Für Full Duplex Token Ring und Full Duplex FDDI haben sich ebenfalls Standards verabschiedet. Auch hier ist das grundsätzliche Verfahren das Abschalten der entsprechenden Zugriffskontrolle.

15.8.4 Gigabit Ethernet (IEEE 802.3z, 802.3ab)

1000 Base T	Cat 5 UTP
1000 Base CX	STP/Twinax
1000 Base SX	Multimode Fiber (850nm)
1000 Base LX	Monomode Fiber oder Multimode (1300nm)

Gigabit Ethernet unterstützt STP und Fiber Optik Verkabelung (IEEE802.3z), sowie UTP Cat. 5 mit bis zu 100m Länge (entspricht 100Base T Anforderungen) unterstützt (IEEE802.3ab).

Die verschiedenen Varianten sind nachfolgend zusammengestellt:



Fiber Optik Standard (IEEE 802.3z)

Für die Optik Version bestehen folgende Randbedingungen:

Grundlagen und Standards für Direktverbindungsnetze

Bezeichnung	Realisierung	Medium	Max. Länge
1000 Base SX	Multimode Fiber (850nm)	62,5? m	260m
1000 Base SX	Multimode Fiber (850nm)	50? m	550m
1000 Base LX	Multimode Fiber (850nm)	62,5? m	440m
1000 Base LX	Multimode Fiber (850nm)	50? m	550m
1000 Base LX	Monomode Fiber	8,3 ? m	3km

1000 Base T auf UTP (IEEE 802.3ab):

Als Zugriffsverfahren wird weiterhin das CSMA/CD verwendet.

Problematik: Bei Erhöhung der Übertragungsrate auf 1Gbit/s möchte man auf die vorhandenen Verkabelungsstrukturen aufsetzen, d.h. die Leitungslängen sollen gleich sein wie bei 100 Base T. Daher wurde bei gleicher Leitungslänge die minimale Paketgröße auf 512 Byte vergrößert.

Da hierdurch ein Nachteil bei kleinen Paketen besteht (die auf die 512 Byte aufgefüllt werden müssen), wird auf der anderen Seite ein Burst Modus eingeführt, der es erlaubt, mehrere Pakete bis auf eine Gesamtlänge von 1500 Byte bis 8kByte zusammenzufassen.

Randbedingungen:

- 100m maximale Länge (90m + 2 Patchkabel mit je 5m)
- gleichzeitig Verwendung von 4 Adernpaaren (250Mbit/s pro Paar)
- Vollduplex Betrieb auf allen 4 Paaren (Echo Cancellation)
- Trellis Codierung (nicht 8B/10B) (Verteilung der 8 Datenbits und 1 Paritybit auf 5 verschiedene Level der 4 Kanäle (4D-PAM5 Codierung)

Zur Codierung: Alle 8ns (=125MHz) wird ein Byte an den Controller übergeben, insgesamt existieren also $2^8 = 256$ verschiedene Datenworte. Durch die 4D-PAM5 Codierung können $5^4 = 625$ mögliche Datenworte gebildet werden, wobei die (bezüglich des Signalabstands zwischen den Adern) geeignetsten verwendet werden. Pro Taktperiode werden 2 Bits übertragen, so dass sich die benötigte Bandbreite auf 62,5MHz reduziert. Durch diese Codierung wird die Übertragung von 1Gbit/s auf einem UTP Kabel Cat5 ermöglicht, dessen maximale Bandbreite mit 100MHz festgelegt ist.

Weitere Eigenschaften:

Gigabit Ethernet besitzt folgende weitere Eigenschaften:

- Automatische MDI/MDI-X Einstellung (Normal/Crossover)
- Auto-Negotiation (10Mbit/s – 100Mbit/s – 1000Mbit/s)
- Festlegung einer Master-Slave Rolle (bei Taktbildung und Taktrückgewinnung)
- Automatische Konfiguration eines DSP Filters (Equalizer, Scrambler)

Diese Eigenschaften werden während der Warmup Phase einer Netzwerkkarte mit dem jeweiligen Gegenüber ausgehandelt.

15.8.5 10 Gigabit Ethernet (802.3ae)

Der 10Gigabit Ethernet Standard wurde im Juni 2002 verabschiedet, näheres findet sich unter <http://grouper.ieee.org/groups/802/3/ae/>. Hier wurde erstmals ein reiner Fiber Optik Ethernet Standard verabschiedet. Überlegungen, auch Kupfer zu unterstützen wurden in der Entwicklungsphase des Standards fallengelassen.

Der Standard unterstützt 2 physikalische Schnittstellen, im LAN Bereich mit 10Gigabit /s, im WAN Bereich wird die Standardbitrate gemäß SONET OC192 bzw. SDH 64 (knapp 10Gbit/s) unterstützt.

15.9 IEEE 802.5 und Token Ring

	802.5	Token Ring
max. Stationen	250	260 / 72
Datenrate	1 MBit/s - 4 MBit/s	4 MBit/s - 16 MBit/s
Medium	not specified	TP
Topologie	not specified	Stern

Token Ring Verkabelung: Star-wired Ring

Token Ring Frame Formate:

Token_Ring

Token_Ring SNAP (vgl. Ethernet)

Frameaufbau Token Ring

AC = Access Control
 FC = Frame Control
 FS = Frame Status
 DA = Destination Address
 SA = Source Address

AC - Byte:

P	P	P	T	F	R	R	R
---	---	---	---	---	---	---	---

P = Prioritätsbits

R = Reserviert für Priorität

T = Token Bit; (0 = Token; 1 = Frame)

M = Monitorbit

sendende Station setzt M = 0

active Monitor setzt M = 1

wenn ein Frame bei dem M = 1 am Active Monitor vorbeikommt

? Frame wird vernichtet (endlos kreisend)

FC - Byte:

Datenfeld enthält Daten oder Control Informationen

FS - Byte

2 Bit - Address Resolution Bits

Empfänger sagt: habe an mich adressierte Nachricht gefunden

2 Bit - Frame Copied Bits

Empfänger sagt: habe Nachricht erfolgreich in meinen Speicher kopiert

Protokollaufgaben:

zentrale Ringwartung ? Überwachungsstation (Active Monitor)

~~steuert~~ die Erzeugung eines Tokens (Claim Token Process)

~~bei~~ Verlust des Tokens ? Ring leeren ? neues Token erzeugen

~~Vernichten~~ endlos kreisender Pakete (M- Bit)

~~Erkennen~~ verstümmelter Frames

~~Ringzusammenbruch~~

~~einfügen~~ von 24 Zusatzbits

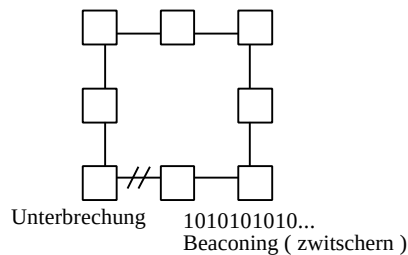
~~Erzeugen~~ eines Taktsignals (Ringtakt)

Mechanismus zur Fehlersuche im Token Ring:

Jede Station überprüft dauernd ob sie vom vorherigen Knoten (NAUN= Nearest Active Upstream Neighbour)

Datenpakete erhält. Bei Leitungsunterbrechungen (Unterbrechung des Rings), sendet sie ein Dauersignal (Beacon Frame). Mit Hilfe eines Netzwerkanalysertools kann dann festgestellt werden, welche Station die meisten Beacon

Frames gesendet hat, die damit am nächsten an der Fehlerstelle positioniert ist.



Physikalische Länge eines Bits:

Datenübertragungsrate 4 MBit/s
Ausbreitungsgeschwindigkeit ca. 200 m/μs
? Bitlänge = (200 m/μs) / (4 MBit/s) = 50 m

Datenübertragungsrate 16 MBit/s
? Bitlänge = (200 m/μs) / (16 MBit/s) = 12,5 m

Das bedeutet z.B., daß ein 4kByte großer Frame im 4Mbit/s Token Ring eine physikalische Länge von 820km besitzt. Bei Ringdurchmessern, die kleiner sind (und das sind sie typischerweise), bedeutet das, daß eine sendende Station quasi gleichzeitig senden und empfangen muß.

Damit dies nicht eventuell auch bei Token selbst geschieht, fügt der active Monitor eine Verzögerung von 24 bit (entspricht der Länge des Tokens selbst) ein.

15.10 FDDI (Fiber Distributed Data Interface)

ANSI - Standard seit 1986

Erster 100 Mbit/s LAN-Standard überhaupt, ursprünglich konzipiert für Glasfaser-Übertragung

Multimodefaser: max Entfernung = 2 km zwischen 2 Stationen

Monomodefaser: max Entfernung = 60 km zwischen 2 Stationen

Max. Anzahl der Stationen = 1000 (effektiv nur 500 ? Doppelring)

Max. Ringdurchmesser = 200 km (effektiv 100 km)

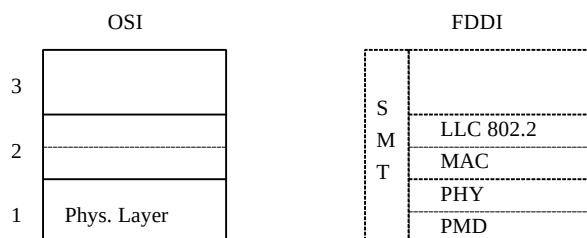
Leitungscodierung:

Token Ring Diff. Manchester

FDDI NRZI 4B / 5B

Umcodierung: Halbbytes werden ersetzt durch 5 Bit - Worte

z.B. 0000 ? 11110 [Rechnernetze nach OSI, Kap. 13.6.3]



PHY = Physical Layer Protocol (Kodierung, Synchronisation)

PMD = Physical Medium Dependend (Zugriff auf optisches Medium)

SMT = Station Management (Asynchrone und / oder synchrone Bandbreite)

FDDI:

Jede Station ist active Monitor

Beaconing wie bei 802.5

Grundlagen und Standards für Direktverbindungsnetze

Framelänge max. 4500 Byte

Varianten bzw. Weiterentwicklungen:

FDDI-2

Token Passing wird ersetzt durch echte synchrone Übertragung, dabei Aufteilung der Bandbreite in feste Slots mit ca 6MB/s Übertragungsrate pro Kanal, Rahmenlänge 12500 bit mit Taktrate 8kHz. Synchrone Übertragung über feste Kanäle, restliche frei Slots im Rahmen können für Paketverkehr verwendet werden.

FDDI auf Twisted Pair 100 MBit/s

Green Book Vorschlag 1991

Weiterentwickelt SDDI

auf STP 150Ohm mit NRZI

CDDI Copper DDI 1990

Umkodierung auf UTP Cat 5

TP-PMD (TP-Physical Medium Dependet -FDDI)

? UTP Cat 5

? STP 150 Ohm IBM Typ 1

Umkodierung auf MLT - 3 (Multilevel3)

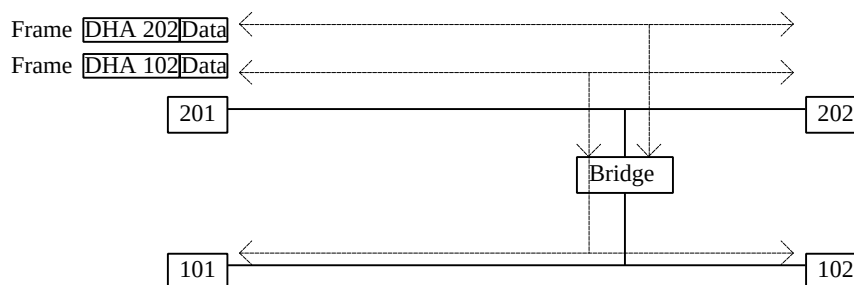
16. Internetworking

16.1 Brücken

Die generelle Aufgabe von Brücken ist die Lastverteilung durch Trennung von Teilnetzen. Ziel ist es, dass der Netzwerkverkehr in einem Teilnetz unabhängig vom Netzwerkverkehr in einem anderen Teilnetz abgewickelt werden kann. Dazu ist es natürlich notwendig, dass die Aufteilung der Stationen entsprechend vorgenommen wird. Nachdem die Trennung so vorgenommen wird, dass die Brücken einzelne Stationen dissesits und jenseits der Brücke anhand ihrer MAC Adressen identifizieren, spricht man von einer Brücke auch von einer MAC Layer Bridge. Der IEEE 802.1d Standard beschreibt die Aufgaben von MAC Layer Bridges im Detail.

2	2	LLC / MAC
1	1	PHY

Die generelle Funktionsweise von Bridges verdeutlicht nachfolgendes Bild:



DHA = Destination Hardware Address

Im Bild sendet die Station mit der (verkürzt dargestellten) MAC Adresse 201. Ein Paket an die Station 202 bleibt auf das obere Segment beschränkt (die Brücke sperrt das Paket), ein Paket an die 102 wird dagegen weitergeleitet. Eine solche Aufteilung ist sinnvoll, wenn der Verkehr zu mehr als 80% innerhalb eines Segments abgewickelt werden kann und weniger als 20% des Verkehrs über die Brücke abgewickelt wird (sog. 80/20 Regel).

Die Hardwareanforderungen an Brücken sind hoch, da jedes Paket gelesen und ausgewertet werden muß. Eine Brücke ist gekennzeichnet durch die sog.

- Filtering Rate (Pakete/s die ausgewertet werden)
- Forwarding Rate (Pakete/s die weitergeleitet werden)

heute (Stand 2002) erreichte Werte in pps(Pakete/s) sind:

Datenrate	Filtering/Forwarding Rate
10 Mbit/s	15000pps
100 Mbit/s	150000 pps
1 Gbit/s	1,5 Mio pps

16.1.1 Transparente bzw. selbstlernende Brücke

Eine transparente Bridge lernt selbstständig durch Lesen von Quelladressen in Paketen (sie werden daher auch als selbstlernende Brücken bezeichnet) und bildet dann interne Tabellen. Hier wird üblicherweise wie folgt verfahren:

- Zieladresse nicht bekannt: weiterleiten und lernen aus der Sourceadresse des Antwortpaketes
- Zieladresse bekannt: weiterleiten oder filtern des Pakets, je nach dem, ob DHA vor oder hinter der Bridge

Nach Erreichen einer max. Anzahl von Einträgen werden die ältesten Einträge gelöscht (LRU Last Recently Used Algorithmus. Transparente Brücken findet man in Ethernets.

16.1.2 Source Routing Bridge

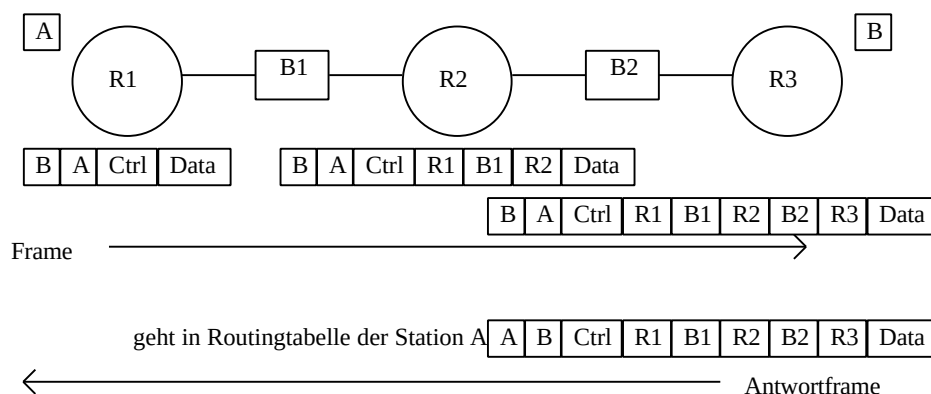
In Token Ring bzw. FDDI Umgebungen findet man einen anderen Typ von Brücken. Ausgehend von der Maßgabe, dass die Rechenleistung der Netzwerkstationen zur Paketvermittlung mitverwendet werden könnte, hat man Brücken entwickelt, die nicht über eigene Adresstabellen verfügen. Statt dessen befinden sich die Adresstabellen in den Netzwerkstationen.

Zunächst wird jede Source Routing Bridge (SR-Bridge) so konfiguriert, dass jeder angeschlossene Port eine eindeutige Ring-ID erhält die mit den entsprechenden Ports anderer Brücken, die ebenfalls mit diesem Port verbunden sind, übereinstimmt. Zudem besitzt jede Brücke eine eindeutige Bridge ID.

Die Frage lautet nun, wie die Adresstabellen in den Netzwerkstationen erstellt werden. Dazu läuft im Netz folgender Discovery Prozeß ab:

- Eine Workstation, die ein Paket senden möchte, sendet zunächst ein Discovery Paket mit der Adresse der Zielstation ins Netz.
- Dieses Paket wird nun überall weitergeleitet, bis es bei der Zielstation ankommt. Die Bridges auf dem Weg fügen dem Frame Information über die Ring-IDs und Bridge IDs auf dem Weg hinzu ? ergibt Bridgereihenfolge von Quelle zum Ziel
- Die empfangende Station sendet das Paket auf dem kürzesten Weg zurück, der sich aufgrund der Source Route Information ergibt.
- Die sendende Station speichert diesen Weg für zukünftige Sendevorgänge

Das nachfolgende Bild zeigt den Discovery Prozess für Station A:



Die Grundproblematik von Source Route Bridging besteht darin, dass bei zunehmender Anzahl von Stationen im Netz (und der damit verbundenen Netzwerklast) durch die steigende Anzahl von Discovery – Paketen eine weitere Steigerung der Netzlast nachfolgt.

16.1.3 SRT Bridges

Brücken, die eine Verbindung von Ethernet und Token Ring Netzen ermöglichen (in Token Ring Netzen wird Source Routing (SR), verwendet, im Ethernet transparente Brücken (T)) nennt man auch SRT Brücken. Hierbei muß beachtet werden, dass MAC Adressen in Token Ring Netzen und Ethernet Netzen unterschiedlich übertragen werden. Ethernet überträgt die einzelnen Bytes der MAC Adressen startend mit dem Least significant Bit (LSB Mode, canonical mode), in Token Ring bzw. FDDI wird für jedes Byte das Most Significant Bit (MSB Mode, noncanonical mode) zuerst übertragen. So muß eine Token Ring Adresse 400011112222 einer Station, die aus dem Ethernet erreicht werden soll im Zieladressfeld des Ethernet Frames byteweise vertauscht werden, also mit 020088884444 adressiert werden.

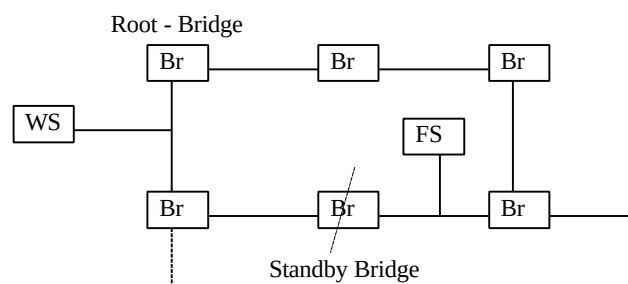
IEEE 802.1d Spanning Tree Protocol

16.1.4 Grundlagen

Nachdem sich der Datenverkehr an den Bridges/Switches im Netz konzentriert, muß ein Ausfall dieser Geräte abgesichert werden. Moderne Switches verfügen an sich schon über interne Redundanzmechanismen (doppelte Backplane, Netzteile etc.). Trotzdem wird man oft einen zweiten Switch neben den ersten stellen bzw in einem LAN mit mehreren Bridges/Switches redundante Wege planen. Hierbei entsteht allerdings dann das Problem von Schleifen (Bridging Loops). Durch diese redundanten Wege kann es vorkommen, dass bestimmte Knoten über mehrere Ports der Brücke erreichbar sind und entsprechende Pakete nicht mehr gefiltert werden. Besonders gefährlich ist das Versenden von Broadcasts, da diese endlos im LAN kreisen können und der Broadcast Verkehr kontinuierlich zunimmt (sog. Broadcast-Sturm).

Abhilfe schafft hier das IEEE802.1d Spanning Tree Protocol, das festlegt, wie durch eine automatische (Re-)Konfiguration von Brücken solche Schleifen verhindert werden können. Durch den Einsatz dieses Protokolls wird veranlasst, dass bestimmte Brücken bestimmte Ports in einen Standby Modus (Backup bzw. Standby Bridge) schalten, so dass es von jedem Punkt im Netz zu jedem anderen genau einen Weg gibt. Dazu wird ausgehend von einer Hauptbrücke (Root Bridge) genau eine Verbindung zu anderen (sog. Designierten) Brücken so zugelassen, dass jedes Netzwerksegment genau einmal erreicht wird. Wenn man sich das LAN als Graphen vorstellt, ist ein Spanning Tree ein Teilgraph, der alle Knoten umfasst, aber nicht alle Kanten. Die sich ergebende Baumstruktur gibt dem Protokoll seinen Namen. Bei Ausfall einer Brücke wird das Netz automatisch rekonfiguriert.

Nutzung mehrer Bridges/Switches im LAN



Das Spanning Tree Protokoll definiert sog. BPDUs (Bridge Protocol Data Units) zur Kommunikation zwischen den Brücken. Ausgangspunkt für die Konfiguration sind zwei Faktoren:

- die sogenannte Bridge ID (8bytes, davon zwei konfigurierbar, die restlichen 6 byte entsprechen der MAC Adresse der entsprechenden Bridge Ports)
- Portkosten, die jedem Bridgeport zugeordnet werden können

Die Format einer BPDU (übertragen im Datenfeld eines Ethernet Frames) sieht wie folgt aus:

	4 byte	2 byte	8 byte	4 byte	8 byte	10 bytes
Ethernet Header	Constants	TC / TCA	Root ID	Port Cost	Bridge ID	Misc.

Die Bedeutung der einzelnen Felder ist folgende :

Constants:

Protocol ID(2), Version(1), Message Type(1)

TC: topology change notification (1)

TCA: topology change notification acknowledgement (1)

Root ID: ID der Root Bridge (8)

Port Cost: Kostenfaktor des Pfads zur Root Brücke in Hops (4)

Bridge ID: ID der sendenden Brücke (8)

Miscellaneous:

Port ID (2): verwendet, wenn zwei Brücken bei der Wahl der Root Bridge dieselbe Bridge ID besitzen

Message Age (2): seit der letzten Konfigurationsnachricht

Maximum Age (2): bis das Paket gelöscht werden sollte

Hello Time (2): seit der letzten Konfigurationsnachricht der Root Bridge

Forward Delay (2): Verzögerung zum Wechsel in den nächsten Status bei einer Topologieänderung

16.1.5 Autokonfiguration

Die Autokonfiguration läuft nun wie folgt ab:

1. Wahl der Root Bridge

Jede Brücke konkurriert mit den anderen um die Funktion der Root Bridge, indem sie Multicast Frames sendet, bei denen sie ihre eigene Bridge ID als Root ID und sich selbst als 0 Hops entfernt von der Root Bridge bekannt macht. Empfängt sie ein Paket mit einer niedrigeren Root ID, stoppt sie den Multicast und leitet stattdessen die empfangenen Pakete mit der niedrigeren Root ID auf die anderen Ports weiter. Schließlich bleibt eine Brücke als Root Bridge übrig, diese sendet nun in regelmäßigen Zeitabständen (IEEE default ist 2s) BPDUs an alle Ports.

2. Bestimmung der designierten Bridges:

Jede Brücke, die BPDUs empfängt, leitet diese an die anderen Ports weiter. Dabei erhöht sie den Wert des Port Cost Feldes (das die Entfernung zur Wurzel angibt) um 1 bzw. um den für diesen Port konfigurierten Kostenwert. Empfängt eine Bridge auf einem Port eine BPDUs, in der angegeben ist, dass die sendende Bridge näher an der Wurzel liegt oder bei gleichem Abstand zur Wurzel eine kleinere Bridge ID besitzt, hört die Bridge mit dem Senden von BPDUs über diesen Port auf (Port geht in den Blocking Status). Die übriggebliebenen Bridges sind die sogenannten Designated Bridges, über die der Weg in die einzelnen Segmente vorgegeben ist.

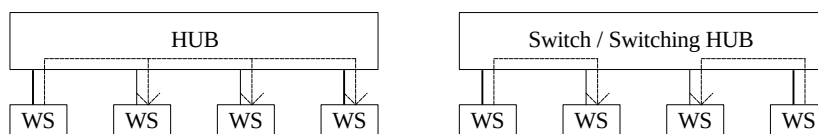
Beispiel:

16.1.6 Topologieänderung

Ein Ausfall einer Bridge führt jetzt aber nicht etwa zu einer Umleitung von Paketen über alternative Wege (dazu ist der Algorithmus nicht in der Lage) sondern zu einer Rekonfiguration des Netzes nach obigem Schema. Backup Bridges lesen dazu den Datenverkehr an den angeschlossenen Segmenten mit. Sieht nun eine Backup Brücke in einem Segment keine BPDUs mehr, geht sie von einem Ausfall der entsprechenden Designated Bridge aus und sendet eine Topology Change Notification (TC) in Richtung Root Bridge. Die Root Bridge sendet ein Topology Change Notification Acknowledgement (TCA) und gibt dabei eine Zeit vor, bis eine neue Rekonfiguration einsetzen soll (Forward Delay). Jede Brücke, die eine BPDU mit dem gesetzten TCA Bit empfängt, löscht alle nicht-permanenten Einträge aus ihrer Datenbasis und wartet die spezifizierte Zeit. Nach Ablauf dieser Zeit konkurrieren wieder alle Bridges um den Status von Designierten Bridges entsprechend ihrer Portkosten wie oben unter Punkt 2 beschrieben. Nachfolgendes Beispiel zeigt den Vorgang:

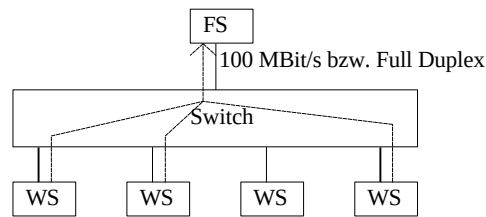
16.2 Frame Switching

In den letzten Jahren hat sich ein Wandel von Hub-basierten shared LANs hin zu switched LANs vollzogen. Hierbei wird die Technologie der Lasttrennung von Bridges verwendet, um den Datenverkehr zwischen zwei Stationen auf die entsprechenden Ports eines Switches zu beschränken und so parallel andere Kommunikationskanäle über andere Ports zu ermöglichen.



Mit einem Switchport werden wenige Stationen (als Shared LAN) verbunden (bei Ethernet Switches spricht man in diesem Zusammenhang von einer sog. Collision Domain). Im Extremfall werden einzelnen Stationen mit den Ports eines Switches verbunden, dies gilt vor allem für Server um die volle Bandbreite des Kanals für den Server zur Verfügung stellen zu können.

Switches erlangten im Ethernet zum Zeitpunkt der Umstellung von Netzwerken von 10Mbit/s auf 100Mbit/s Bedeutung. Hier hat man oft die Serververbindung auf 100Mbit/s aufgerüstet, und die vorhandene Verkabelung und Netzwerkkarten auf Clientseite weiterverwendet. Durch das Multiplexen mehrerer 10Mbit/s Kanäle auf 100Mbit/s ergab sich so ein Geschwindigkeitsvorteil in Netzen, die bis dahin eine hohe Auslastung zeigten.



Die grundsätzliche Funktionsweise eines Switches entspricht dem einer Multiport Bridge. Allerdings gibt es hier häufig unterschiedliche Funktionsweisen des Switching Vorgangs an sich, bzw. unterschiedliche Switch Typen.

Man unterscheidet:

- Store-And-Forward Switching
- Cut-Through Switching

16.2.1 Store-And-Forward Switching

Klassische Brücken speichern ein Paket zwischen, bevor sie es weiterleiten. Dies ermöglicht es einer Brücke zunächst, das weiterzuleitende Paket auf seine Integrität zu prüfen. Die führt jedoch zu einer Latenzzeit bei der Weiterleitung von Paketen (je nach Framelänge bis zu 1200?s). Bei Switches heißt dieser Modus Store-And-Forward Switching. Zum Vergleich: Router brauchen üblicherweise noch etwas länger (bis zu 1600?s). Manche Switches erlauben die Zwischenspeicherung nur eines Teils der Daten (z.B. 512bit) um zumindest Fragmente einer Kollision als solche zu erkennen und zu filtern. Dadurch werden kurze Latenzzeiten (64?s) erreicht.

16.2.2 Cut-Through Switching

Bei dieser Technologie wird die Latenzzeit verkürzt, indem ein Paket sofort an den entsprechenden Zielport weitergeleitet wird. Latenzzeiten liegen hier in einer Größenordnung von etwa 40?s. Diese Switches werden vor allem dort eingesetzt, wo innerhalb von LANs Lasttrennung erfolgen muß, aufgrund entsprechender Anwendungen (z.B. Videoconferencing, LAN Telefonie) längere Latenzzeiten aber nicht vertretbar sind.

16.2.3 Intelligent Switching

Moderne Switches sind in der Lage, den Paketdatenverkehr zu überprüfen und sozusagen nebenbei die Qualität der Übertragung zu überprüfen. Bleibt die Anzahl der fehlerhaften Pakete, die von einem Switch erkannt werden, unterhalb einer bestimmten Schwelle, ist der Switch im Cut-Through Modus, steigt die Fehlerrate, geht er automatisch in den Store-And-Forward Modus.

16.2.4 Token Ring Switching

Auch in Token Ring bzw. FDDI Umgebungen ist zunehmend Switching im Einsatz. Hier muß aber eine Zwischenspeicherung von Paketen vorgenommen werden, da durch das Token Ring protokoll am Zielport ein sofortiges Weiterleiten nicht möglich ist. Ähnlich kurze Latenzzeiten wie beim Cut-Through Switching im Ethernet lassen sich hier nicht erreichen.

16.2.5 Quality of Service

Speziell mit der zunehmenden Anforderung an geringe Latenzzeiten für bestimmte Dienste (QoS: Quality of Service) ist speziell für Switches in LANs der Ruf nach Prioritätssteuerung immer größer geworden. So sehen heutige Switches entsprechende Mechanismen vor, mit IEEE802.1p wurde hierfür ein eigener Standard für QoS auf MAC Ebene definiert. Der Standard sieht die Aufteilung des Datenverkehrs in 8 verschiedene Prioritätsstufen vor, Pakete niedrigerer Priorität werden nicht gesendet, solange Pakete höherer Priorität in entsprechenden Warteschlangen vorhanden sind. Die Priorität wird dabei vom Netzwerkkartentreiber der Netzwerkstationen vorgegeben. Das entsprechende Frameformat wurde mit IEEE 802.3ac definiert (s. später).

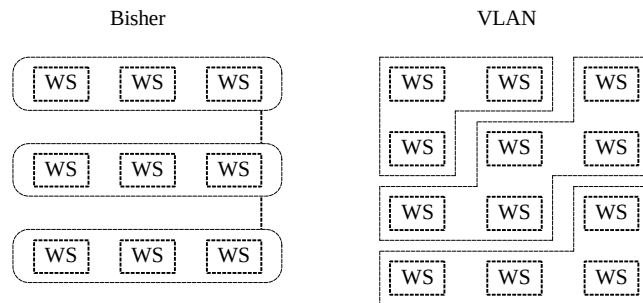
16.3 Virtuelle LANs

16.3.1 Funktion von VLANs

Heutige Switches unterstützen die Aufteilung des Netzes in sogenannte Virtuelle LANs z.B. um den Netzwerkverkehr zwischen Abteilungen eines Unternehmens zu trennen. Hierbei werden die einzelnen Ports

Mitglieder von getrennten VLANs, die als eigenes Netzwerksegment betrachtet werden. Sendet eine Station z.B. einen Broadcast, bleibt dieser auf alle Knoten innerhalb des VLANs beschränkt. Mehrere VLANs werden dann über Router zu einem Internetwork verbunden. Dieser Router kann ein eigenes Gerät sein, das zusammen mit den VLAN Switches eingesetzt wird, modernere Router ermöglichen das Routing im Switch (sog. Layer 3 Switch).

Dies ermöglicht eine grundsätzlich neue Organisationsform von Netzen. Bisher galt, dass mit einer logischen Aufteilung in Netze über Router auch eine physikale Aufteilung in verschiedene geographische Bereiche einher ging.



Durch die Verwendung von VLANs ist nun die logische Ebene von der physikalischen Ebene völlig getrennt. Durch Layer-3 Switching können innerhalb eines Switches mehrere log. Netze koexistieren. Damit sind z.B. physikalische Änderungen des Netzes bei Änderung der Arbeitsgruppen innerhalb eines Netzes nicht mehr nötig.

Layer 3 Switches sind hochkomplexe Geräte. So stehen oft getrennte Prozessoren für die einzelnen Aufgaben wie Routing, Layer 2/3 Switching und die einzelnen Ports zur Verfügung.

Die generelle Konfigurations- und Funktionsweise von VLANs (Architekturüberblick), Forwarding-Regeln, Topologie- und Bridgemanagement (z.B. VLAN Tagging) sind im IEEE 802.1q Standard festgelegt.

16.3.2 Statische und Dynamische VLANs

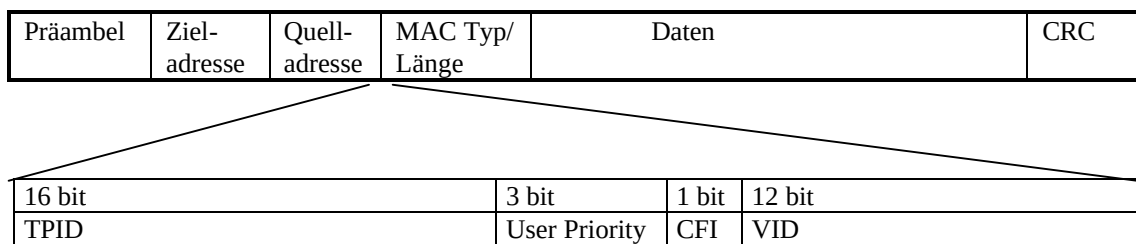
VLANs können an den zu verwendenden Switches von einem Administrator direkt konfiguriert werden (statische VLANs) oder durch den Einsatz spezieller Protokolle (GARP: Generic Attribute Registration Protocol bzw. GVRP: GARP VLAN Registration Protocol) als dynamische VLANs. GVRP erlaubt es Information über VLANs zwischen Netzwerkknoten und Switches zu verteilen. Dies ermöglicht es z.B. mobilen Benutzern, deren Rechner als Teil eines bestimmten VLANs konfiguriert ist, sich mit beliebigen Ports eines Switches zu verbinden. Dieser Port wird dann dynamisch dem entsprechenden VLAN zugeordnet.

16.3.3 VLAN Tagging

Heutige Switches erlauben die Aufteilung von VLANs über Switchgrenzen hinweg. Dazu muß ein Mechanismus existieren, der es erlaubt, Pakete zwischen den Switches zu senden und dabei zu einem bestimmten VLAN gehörig zu kennzeichnen (engl. Tagging). Jedes VLAN erhält dazu eine bestimmte VLAN ID (VID). Diese wird den einzelnen Frames bei der Kommunikation zwischen den Switches hinzugefügt. Inzwischen wurde hierfür mit IEEE 802.3 ac eine Erweiterung für den Ethernet Standard für das sogenannte Tag Switching entwickelt. Durch den Einsatz von Tagging kann nun jeder Port des Switches zu einem oder auch mehreren VLANs mit unterschiedlichen IDs gehören.

Nachfolgendes Bild zeigt ein Beispiel. Hier sind 3 VLANs über zwei Switches realisiert. VLAN 1 ist innerhalb des ersten Switches realisiert, VLAN 2 und 3 über Switchgrenzen hinweg. Innerhalb der VLANs 2 und 3 werden nun alle Ports speziell gekennzeichnet, für die eine Kommunikation über Switchgrenzen hinweg auf den jeweils anderen Teil von VLAN2 bzw. 3 ermöglicht werden soll. Die Verbindung zwischen den beiden Switches wird als VLAN-Trunk bezeichnet. So ist hier der Server1 an Port 2 aus allen VLANs erreichbar.

UM VLAN Tagging zu ermöglichen, muß das Ethernet-Frameformat entsprechend erweitert werden. Der IEEE standard 802.3ac spezifiziert vier zusätzliche Byte zur Kapselung (encapsulation) zusätzlicher Daten zur Verwendung für die Verwendung in VLAN Umgebungen vor dem Längenfeld eines IEEE802.3 Frames. Die Verwendung der zusätzlichen Felder ist in IEEE 802.1q festgelegt:



Die einzelnen Felder bedeuten:

TPID: Protokolltypfeld, das von IEEE802.1q als 0x8100 festgelegt wurde

User Priority: Legt fest welche Quality of Service (Level 0 bis 7) für den Frame anzusetzen ist.

CFI: Canonical Flag Indicator, zeigt an ob die MAC Adressinformation im Frame kanonisches Format repräsentiert

VID: Gibt die VLAN ID an. Reservierte Werte sind:

- 0: Header beinhaltet nur User Prioritätsinformation und keine gültige VLAN ID
- 1: Default VLAN, wird an alle „untagged“ Ports geschickt
- FFF: Reserviert.

Damit sind konfigurierbare VIDs nach IEEE802.1q VID 2 bis 4094

VLANs sind bei verschiedenen Switches jedoch nicht durchgehend gleich realisierbar. Speziell bei älteren Switches gibt es unterschiedlichste Realisierungen der VLAN Technik. Die Möglichkeit, VLANs über Switchgrenzen hinweg über VLAN Trunks zu realisieren, setzte, sofern dies überhaupt möglich war, auf proprietären Mechanismen auf.

Teilweise wird der Begriff VLAN nicht nur zur Trennung von Netzen über Router verwendet, sondern auch über Bridges, was ja z.B. für nicht routingfähige Protokolle durchaus sinnvoll ist.

Teilweise wurde hier dann Ports innerhalb eines VLANs als Cut-Through Ports realisiert, über VLAN Grenzen hinweg wurde dann z.B. Store-And-Forward Switching eingesetzt.

16.4 Einsatzgebiete von Bridges, Switches und Routern

Reine Bridges findet man heute so gut wie nicht mehr, wenn dann nur innerhalb von sogenannten Broutern. Brouter sind Multiprotokollrouter, die alle Protokolle routen, die sie kennen, und für den übrigen Netzwerkverkehr als Bridge arbeiten. Ansonsten spricht man heute von Switches.

Switches werden heute sehr weit innerhalb von LANs eingesetzt, das typische Design ist hierarchisch zu immer höheren Übertragungsraten hin zum sogenannten Backbone des Netzes. Aufgrund von Möglichkeiten zur Lasttrennung von Bereichen durch den Einsatz von VLANs findet man klassische Router an den Grenzen zu Abteilungen (evtl. mit zusätzlichen Filterfunktionen als sog. Abteilungsfirewalls (Siehe Kap. Sicherheit in Netzen)), dies dann oft als Layer 3 Switches. bzw. an der Grenze zwischen LAN und WAN.

Typische Unterschiede zwischen Bridges, Switches und Routern sind nachfolgend kurz zusammengestellt:

Router	Bridge/Switch
Begrenzung auf bestimmte Protokolle (Network-Layer) evtl. zusätzliche Bridgefunktion	Netzwerk-Protokolltransparenz
Wird aktiv, wenn Paket an Router adressiert	liest jedes Paket
Leitet Pakete aufgrund einer Routingtabelle weiter	Filtert Pakete aufgrund von Zieladressen
Netzwerkadressen	MAC Adressen
Trennung eines Netzes in log. Subnetze	1 logisches Netz

	IEEE 802.1d Bridge	Switch
Lernen von Adressen	Alle Ports	Konfigurierbar
Weiterleitung von Paketen	Store and Forward Mode	Verschiedene Modi
Aktion bei vollem Puffer	Vernichtung von Paketen	Abhängig vom Switch Design
Spanning Tree	unterstützt	Teilweise unterstützt
Aktion bei unbekannter Zieladresse	Weiterleiten an alle Ports	Konfigurierbar

17. Wireless LAN

17.1 IEEE 802.11 Wireless LAN

Der 802.11 Standard definiert ein MAC protokoll und drei verschiedene Physical Layer Interfaces. Verwendete Übertragungsraten sind bei allen drei Medien 1 oder 2 Mbit/s (durch verschiedene Modulationsarten).

Die drei möglichen Medien sind:

- Funkübertragung mit Frequency Hopping Spread Spectrum (FHSS)
- Funkübertragung mit Direct Sequence Spread Spectrum (DSSS)
- Infrarotübertragung (bei 850nm ...950nm)

Für die Funkübertragung wird das sog. ISM Band (Industrial, Scientific, Medical 2,4 ...2,4835 GHz) verwendet.

17.1.1 Spreizspektrum Übertragung

Um bei der Funkübertragung eine größere Störsicherheit zu erreichen, wird die Übertragung über ein breiteres Frequenzband durchgeführt als nötig (Spreizspektrum, Spread Spectrum). Hier unterstützt 802.11 zwei verschiedene Verfahren:

- Frequency Hopping Spread Spectrum (FHSS)
- Direct Sequence Spread Spectrum (DSSS)

Beim FHSS (Frequenzsprungverfahren) wird das zu übertragende Signal auf ständig wechselnden Trägerfrequenzen moduliert (2,5 Frequenzwechsel pro Sekunde). Der Frequenzwechsel geschieht nach einem bestimmten Algorithmus, basierend auf Verwendung eines Pseudozufallszahlengenerators. Sender und Empfänger verwenden den gleichen Startwert und Algorithmus. 802.11 spezifiziert 79 1MHz Kanäle.

Beim DSSS wird jedes Bit des zu übertragenden Frames durch mehrere Bits (EXOR des Bits mit n Zufallsbits (sog. Chipping Code)) ersetzt. Dadurch wird das Frequenzband n mal breiter. Bei IEEE802.3 ist n=11.

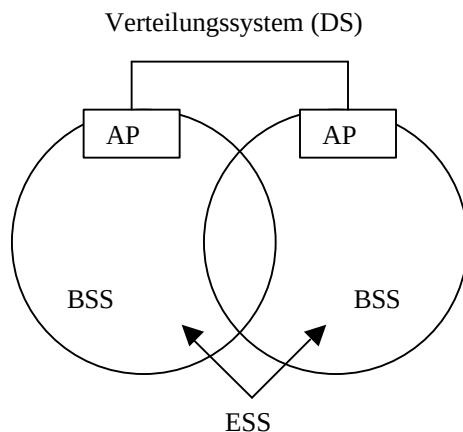
Die Infrarotübertragung erfolgt diffus, wodurch keine klare Sichtverbindung zwischen Sender und Empfänger existieren muß. Allerdings ist die Reichweite bei dieser Technologie auf 10m begrenzt. IEEE802.11 hat sich aufgrund der geringen Übertragungsraten nicht sehr weit durchgesetzt.

Die Übertragung erfolgt verschlüsselt, wobei das WEP Protokoll (Wired Equivalent Privacy) eingeführt wurde, das i.A. inzwischen jedoch nicht als besonders sicher gilt.

17.1.2 Kopplung von Knotengruppen

Die Knoten innerhalb eines Wireless LANs werden in Knotengruppen (Basic Service Sets, BSS) zusammengefasst. Die geographische Fläche die diese Knotengruppe umfasst wird als Basic Service Area (BSA) bezeichnet. Die Rechner innerhalb einzelner Knotengruppen können dabei als Ad-Hoc Netzwerk auf Peer to Peer Basis arbeiten (Independent-BSS, IBSS), oder sie werden sogenannten Anschlusspunkten (Access Points, AP) zugewiesen, die als wireless Hubs arbeiten. Diese sind häufig gleichzeitig Übergangspunkte (der IEEE 802.11 Standard nennt sie Portale) in ein leitungsgebundenes LAN und sind damit funktional mit Brücken (Bridges) vergleichbar.

Innerhalb der Knotengruppe arbeiten die Knoten wie soeben beschrieben, über Knotengruppen-grenzen hinweg erfolgt die Kommunikation über die jeweiligen Anschlusspunkte über das dazwischenliegende LAN (das in diesem Zusammenhang dann auch als Verteilungsnetzwerk oder Distribution Network bzw. DS bezeichnet wird). Ein WLAN, das aus mehreren Knotengruppen mit dazwischenliegendem Verteilungsnetzwerk besteht, wird auch als Extended Service Set ESS bezeichnet.



Die WLAN Knoten suchen sich dazu einen verfügbaren Anschlusspunkt per *Scanning* Verfahren wie folgt:

1. Der Knoten sendet einen *Probe* Frame
2. Alle Anschlusspunkte innerhalb der Reichweite antworten mit einem *Probe-Response* Frame
3. Der WLAN Knoten wählt sich den Anschlusspunkt mit der stärksten Sendeleistung aus und sendet ihm einen *Association Request* Frame
4. Der Anschlusspunkt antwortet mit einem entsprechenden *Association Response* Frame

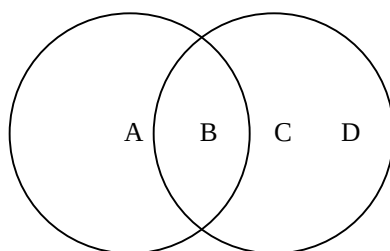
Dieser Vorgang wiederholt sich immer dann, wenn ein Knoten wandert und irgendwann nicht mehr mit seinem momentanen Anschlusspunkt „zufrieden“ ist. Dabei sendet er kontinuierlich Probe Frames, die ihn irgendwann zu einem besser erreichbaren Anschlusspunkt führen.

17.1.3 Zugriffsverfahren

802.11 verwendet als Zugriffsprotokoll ein CSMA/CA, das hier als MACAW (Multiple Access Collision Avoidance for Wireless LANs) bezeichnet wird. Der Ablauf ist wie folgt:

1. Vor dem eigentlichen Sendevorgang schickt der Sender ein RTS (Request to Send) an den Empfänger, der eine Angabe über die Länge des zu übertragenden Datenframes enthält.
2. Der Empfänger antwortet mit einem CTS (Clear to Send) Frame (mit dem gleichen Längensfeld). Dies ist das Signal für alle anderen Knoten, die von dieser Übertragung Kenntnis haben, dass sie selbst nicht senden dürfen. Jeder Knoten verwaltet dazu den sogenannten Network Allocation Vector (NAV), der die verbleibende Besetztzeit des Übertragungskanal in μ s anzeigt.
3. Der Sender sendet den Frame mit der vorher vereinbarten Länge
4. Der Empfänger quittiert den erhaltenen Frame mit einem ACK. Dies ist gleichzeitig für die anderen Knoten im sichtbaren Umfeld eine zusätzliche Bestätigung, dass sie wieder senden dürfen.

Aufgrund der begrenzten Reichweiten von Funknetzen (100m, bis zu 2km mit Richtantennen) kann es vorkommen, dass sich die Stationen A und C aufgrund Ihrer begrenzten Reichweite (angedeutet durch einen Kreis im nachfolgenden Bild) gar nicht sehen können, sie aber beide mit einer Station B in der Mitte kommunizieren wollen. A und C sind damit für einander verborgene Knoten (Hidden Nodes).



Umgekehrt könnte es sein, dass B mit A kommuniziert. Nun wäre es falsch, wenn C annehmen würde er könne nicht senden, nur weil er B hört. C könnte in diesem Szenario durchaus an D (exponierter Knoten, exposed Node) senden. Insgesamt müssen also folgende Fälle unterschieden werden:

Wireless LAN

- Ein Knoten sieht einen CTS Frame: Er weiß, dass er in der Nähe eines Empfängerknotens liegt und daher jetzt nicht senden darf.
- Ein Knoten sieht einen RTS Frame, nicht aber den CTS Frame. Er weiß, dass er zwar in der Nähe eines anderen Senders liegt, den entsprechenden Empfänger der Nachricht aber nicht sehen kann. Er kann daher selbst senden.

Natürlich kann es nach wie vor zu Kollisionen kommen, allerdings nur zu Kollisionen zwischen RTS Frames. Diese Kollision wird allerdings nur dadurch erkannt, dass die Sender nach einer bestimmten Zeit immer noch kein CTS erhalten haben. In diesem Fall warten Sie eine zufällige Zeitspanne (Exponential Backoff wie beim Ethernet), bis sie es erneut versuchen.

Dieses Zugriffsprotokoll übernimmt also wie auch in einem leitungsgebundenen Shared LAN die Aufgabe, die verfügbare Bandbreite (hier von 1 oder 2 Mbit/s) innerhalb eines BSS auf die verschiedenen, gleichzeitig sendenden Knoten aufgeteilt. Je nach Anwendung (übertragener Datenmenge) ergibt sich so eine praktische Grenze für die sinnvolle Höchstzahl von Knoten innerhalb eines BSS, die üblicherweise unter der Höchstzahl von Knoten liegt, die von einem Access Point unterstützt werden könnten (ein üblicher Maximalwert ist 256). Sollen mehr Knoten unterstützt werden, empfiehlt sich die Aufteilung auf mehrere BSS durch Verwendung mehrerer Access Points.

17.1.4 Frame Format

Das 802.11 Frameformat sieht wie folgt aus:

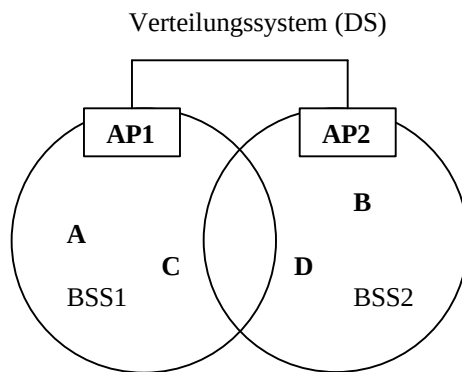
2 Byte	2 Byte	6 Byte	6 Byte	6 Byte	2 Byte	6 Byte	0 – 2312 Byte	4 Byte
Frame Control	Duration / Conn. ID	Addr1	Addr2	Addr3	Sequence Control	Addr4	Daten	CRC

Die einzelnen Felder werden wie folgt verwendet:

- Daten: besitzt eine variable Länge von 0 bis 2312 Byte
- Duration/Connection ID: gibt die Übertragungsdauer in ?s an
- Frame Control: Kennzeichnung der Adressen, Frame Management wie folgt:
 - Type/Subtype (6 bit): Kennzeichnet Datenframe, RTS, CTS oder ACK, sowie die gesamte Steuerinformation zum Scanning nach Access Points etc.
 - ToDS (1 bit): Wird gesetzt, wenn ein Frame an einen Access Point adressiert wird
 - FromDS (1 bit): ist gesetzt, wenn ein Frame von einem Access Point kommt.
 - More Fragments (erlaubt Fragmentierung von Frames)
 - WEP: weist darauf hin, wenn die Daten mit WEP verschlüsselt sind
- Addr1-4: Auch hier werden die IEEE typischen 6Byte (48Bit) MAC Adressen verwendet, hier allerdings vier, da die Möglichkeit besteht, dass ein Frame über ein Verteilsystem an eine andere Knotengruppe weitergeleitet wird. Die Steuerung erfolgt über die ToDS bzw From DS Bits. Addr1: Ist die Zieladresse eines Sendevorgangs innerhalb eines BSS
 Addr2: Ist die Quelladresse eines Sendevorgangs innerhalb eines BSS
 Addr3: Ist die fehlende Anfangs bzw. Endadresse der gesamten Kommunikationsbeziehung

Die Verwendung der Adressen sowie der Steuerbits ToDS bzw. FromDS wird am nachfolgenden Beispiel einer Kommunikation innerhalb einer BSS (von A $\not\rightarrow$ C) und über ein Verteilsystem (von A $\not\rightarrow$ D) deutlich. Bei letzterer teilt sich der Kommunikationspfad auf drei Anteile auf (Von A zu AP1, von AP1 zu AP2 innerhalb des LANs (in der Tabelle nicht gezeigt), sowie von AP2 zu D).

Sendevorgang	FromDS	ToDS	Addr1	Addr2	Addr3	Addr4
A $\not\rightarrow$ C	0	0	C	A	AP1	
A $\not\rightarrow$ D (innerhalb BSS1)	1	0	AP1	A	D	
A $\not\rightarrow$ D (innerhalb BSS2)	0	1	D	AP2	A	
A $\not\rightarrow$ D (innerhalb DS)	1	1	AP2	AP1	D	A



Addr4 wird nur für den speziellen Fall verwendet, dass das Verteilungssystem auch ein Wireless LAN ist, und wird dann für die Quelladresse des ursprünglichen Senders verwendet. Obige Tabelle zeigt den entsprechenden Frame zwischen den beiden Zugriffspunkten innerhalb des Wireless Verteilungssystems.

17.2 IEEE802.11 Nachfolgestandards

IEEE 802.11b

Der jetzt aktuelle Standard 802.11b überträgt im 2,4GHz Band bis zu 11Mbit/s (1, 2, 5,5 oder 11Mbit/s). Dabei wird nur noch Funkübertragung mit Direct Sequence Spread Spectrum (DSSS) unterstützt. Um die Kompatibilität der unterschiedlichen Produkte zu gewährleisten, wurde ein Interoperability Standard Wi-Fi (Wireless Fidelity) eingeführt, nach dem entsprechende Produkte zertifiziert werden (<http://www.wi-fi.org>). Die Wireless Ethernet Compatibility Alliance ([WECA](http://www.wi-fi.org)) zertifiziert die Interoperabilität der 802.11-kompatiblen Geräte. In diesem Zusammenhang werden die Geräte nach IEEE802.11 auch unter dem Markennamen Wi-Fi (Wireless Fidelity) vermarktet.

IEEE 802.11a

Dieser zu den anderen 802.11 Standards inkompatibler Standard sieht die Übertragung von bis zu 54Mbit/s bei 5GHz vor. Allerdings ist dieser Standard in Deutschland nicht zugelassen, da der entsprechende Frequenzbereich für die Satellitenkommunikation reserviert ist. Für die störungsfreie Zusammenarbeit mit diesen Diensten fehlen Ergänzungen, die mit dem Standard IEEE 802.11h geschaffen werden sollen.

IEEE 802.11g

Ein neuerer Standard wird 802.11g sein, der abwärtskompatibel zu 802.11b ist und durch andere Modulationsarten ebenfalls bis zu 54Mbit/s Übertragungsraten aufweist.

17.3 Weitere Wireless Übertragungsstandards

Bluetooth

Die Bluetooth Special Interest Group (BSIG) wurde erst Anfang 1998 von fünf Firmen (IBM, Toshiba, Intel, Ericsson, Nokia) gegründet. Nur kurze Zeit später konnte der Standard verabschiedet werden.

Die wichtigsten Eigenschaften von Bluetooth sind:

- Frequenzbereich: Bluetooth wird im 2,4 GHz-ISM-Band betrieben.
- Modulationsverfahren: Um in diesem lizenz- und genehmigungsfreien Frequenzbereich eine zuverlässige Übertragung zu erreichen, setzt Bluetooth das oben erwähnte Frequency Hopping Spread Spectrum (FHSS) ein. Hierbei wird die Trägerfrequenz alle 625 µs nach einem zwischen Sender und Empfänger vereinbarten Ablauf gewechselt.
- Reichweiten: Die Reichweite der Bluetooth-Systeme ist ohne Richtantenne auf etwa 10 m beschränkt.
- Datenraten und Verkehrstypen: Bluetooth unterstützt sowohl synchrone als auch asynchrone Übertragungsmodi. Somit ermöglicht Bluetooth sowohl eine Sprachübertragung mit einer Bandbreite von 64 kbps in beide Übertragungsrichtungen, als auch die Datenübertragung mit einer Bandbreite von 865,2 kbps.

- Dienste: Der Bluetooth-Standard beschreibt nicht nur die beiden untersten Ebenen des Protokollstapels, sondern auch Dienste, die auf den höheren Schichten beschrieben werden. Auf diese Weise kann Bluetooth so genannten Ad-Hoc-Netzwerke unterstützen.

IrDA

hat sich inzwischen als Infrarot Standard im PAN (Personal Area Network) Bereich durchgesetzt.

Die Infrarot-Aktivitäten müssen beim Blick in die Zukunft außen vorgelassen werden. Trotz der kostengünstigen Hardware, der großen Verbreitung und der Verankerung in den bestehenden Betriebssystemen von Notebooks, PDAs und Handys sieht deren Zukunft düster aus. Insbesondere die notwendige Ausrichtung der sendenden und empfangenden Geräte ist für die Vernetzung von mehr als zwei Stationen außerordentlich lästig.

HiperLAN/2

HIPERLAN/2 ist der europäische Standard für wireless LANs, entwickelt vom European Telecommunication Standards Institute (ETSI), verabschiedet im April 2000. Ziel war es, einen Standard zu schaffen, der so nahe wie möglich an die Übertragungsraten leitungsgebundener Netze herankommt. Es wird eine Bitrate von 54 MBit/s bei 5GHz zur Verfügung gestellt. Der Standard steht sozusagen in Konkurrenz mit dem 802.11b.

18. Anhang

18.1 Verwendete Abkürzungen

ABR	Available Bit Rate (ATM)
ACK	Acknowledgement Paket
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation 1 (Maschinenunabhängige Darstellung von Information, OSI Protokoll)
ATM	Asynchronous Transfer Mode
AMI	Alternate Mark Inversion (Code)
ANSI	American National Standards Institute
APPC	IBM Advanced Program to Program Communication
APPN	IBM Advanced Peer to Peer Networking
AUI	Attachment Unit Interface
BPDU	Bridge Protocol Data Unit
CAT	Category (Kabelkategorie)
CBR	Constant Bit Rate
CCITT	Consultative Committee for International Telegraphy and Telephony
CDDI	Copper Distributed Data Interface
CLNP	Connectionless Network Protocol (OSI Layer 3)
CMI	Coded Mark Inversion (Code)
CMIP	Common Management Information Protocol (OSI)
CMOT	CMIP over TCP/IP (OSI)
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DARPA	Department of Defense Advanced Research Projects Agency
DAS	Dual Attachment Station
DHCP	Dynamic Host Configuration Protocol
DIX	Digital-Intel-Xerox
DoD	Department of Defense (USA)
DNA	Digital Network Architecture (DEC)
DS	Directory Services (OSI Protokoll)
EBCDIC	Extended Binary Coded Decimal Interchange Code (IBM)
ES-IS	End System-Intermediate System (OSI Routing Protocol)
FAT	File Allocation Table
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
FDM	Frequency Division Multiplexing
FTAM	File Transfer, Access and Management (OSI-Protokoll)
FTP	File Transfer Protocol (Internet Protokoll)
HPFS	High Performance Filing System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IAB	Internet Architecture Board
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronical Engineers
IETF	Internet Engineering Task Force

IP	Internet Packet
IPng	IP Next Generation
IPX	Internet Packet Exchange (Novell)
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System-Intermediate System (OSI Routing)
ISO	International Standards Organisation
ITU	International Telecom Union
LAN	Local Area Network
LAN-E	LAN Emulation (ATM)
LAT	Local Area Transport (DEC- Terminalserver- Protokoll)
LLC	Logical Link Control
LSP	Link State Protocol
LU	Logical Unit (IBM-Welt)
LU6.2	IBM Peer to Peer Protokoll
MAC	Media Access Control
MacOS	Macintosh Operating System
MAN	Metropolitan Area Network
MAU	Medium Attachment Unit (Transceiver)
MDT	Mittlere Datentechnik
MIB	Management Information Base
MHS	Message Handling System
MPOA	Multiprotocol over ATM
MSAU	Multiple Station Access Unit (Ringleitungsverteiler)
NAK	Negative Acknowledgement Paket
NAU	Network Addressable Unit (z.B. PU, LU, SSCP)
NCP	NetWare Core Protocol
NDS	Novell Directory Services
NFS	Network File System
NLSP	Netware Link Services Protocol (IPX Routing)
NNTP	Network News Transfer Protocol
NRZ	Non Return to Zero
NRZI	Non Return to Zero Inverted
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First (Routing)
PAN	Personal Area Network
PHY	Physical Layer Protocol
PMD	Physical Layer Medium Dependent
POP	Post Office Protocol
PU	Physical Unit (IBM-Welt)
PUCP	Peripheral Unit Control Point (Lokaler SSCP in Cluster Controller, IBM-Großrechnerwelt)
RAID	Redundant Array of Inexpensive Disks
RIP	Routing Information Protocol
RPC	Remote Procedure Call
RS	Return to Zero
SAP	Service Advertising Protocol (Novell)
SAP	Service Access Point
SAS	Single Attachment Station
SSCP	System Services Control Point (Steuert Session-Aufbau in IBM Großrechnern)
SDDI	Shielded Distributed Data Interface

SDH	Synhronous Digital Hierarchy (Eoropäisches Equivalent zu SONET)
SFD	Start of Frame Delimiter (Frame-Anfangsfeld)
SMI	Structure and Identification of Management Information
SMT	Station Management (FDDI)
SMTP	Simple Mail Transfer Protocol (Internet Protokoll)
SNA	Systems Network Architecture (IBM)
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network (High Speed Standard bis 2.4GBit/s)
SPX	Sequenced Packet Exchange (Novell)
SUTP	Screened Unshielded Twisted Pair
STMP	Simple Mail Transfer Protocol
STP	Shielded Twisted Pair
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
TELNET	Protokoll zur Terminal Emulation (Internet Protokoll)
TPDU	Transport Protocol Data Unit (Internet Protokollwelt)
TP4	Transport Protocol 4 (eines der OSI Transport Protokolle)
TP-PMD	Twisted Pair-Physical Medium Dependent (Norm für FDDI über Twisted Pair)
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol (Internet Protokoll)
UTP	Unshielded Twisted Pair
VBR	Variable Bit Rate
VLAN	Virtual LAN
VLM	Virtual Loadable Module
VC	Virtual Channel
VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier
VT	Virtual Terminal (OSI-Protokoll)
VTAM	Virtual Telecommunications Access Method („Netzwerkbetriebssystem“ der IBM Großrechnerwelt)
WAN	Wide Area Network
WLAN	Wireless LAN
XDR	External Data Representation (Internet Protokoll)

18.2 BOOTP AND DHCP PARAMETERS

<http://www.iana.org/assignments/bootp-dhcp-parameters>

(last updated 2002-10-10)

The Bootstrap Protocol (BOOTP) [RFC951] describes an IP/UDP bootstrap protocol (BOOTP) which allows a diskless client machine to discover its own IP address, the address of a server host, and the name of a file to be loaded into memory and executed. The Dynamic Host Configuration Protocol (DHCP) [RFC2131] provides a framework for automatic configuration of IP hosts. The "DHCP Options and BOOTP Vendor Information Extensions" [RFC2132] describes the additions to the Bootstrap Protocol (BOOTP) which can also be used as options with the Dynamic Host Configuration Protocol (DHCP).

BCP 29, RFC 2489, "Procedure for Defining New DHCP Options" [RFC2489], describes the procedure for defining new DHCP options. This new procedure provides guidance to IANA in the assignment of new option codes. The DHCP option number space (1-254) is split into two parts. The site-specific options (128-254) are defined as "Private Use" and require no review by the DHC WG. The public options (1-127) are defined as "Specification Required" and new options must be reviewed prior to assignment of an option number by IANA.

BOOTP Vendor Extensions and DHCP Options are listed below:

Tag ---	Name ----	Data Length -----	Meaning -----
0 [RFC2132]	Pad	0	None
1 [RFC2132]	Subnet Mask	4	Subnet Mask Value
2 [RFC2132]	Time Offset	4	Time Offset in Seconds from UTC
3 [RFC2132]	Router	N	N/4 Router addresses
4 [RFC2132]	Time Server	N	N/4 Timeserver addresses
5 [RFC2132]	Name Server	N	N/4 IEN-116 Server addresses
6 [RFC2132]	Domain Server	N	N/4 DNS Server addresses
7 [RFC2132]	Log Server	N	N/4 Logging Server addresses
8 [RFC2132]	Quotes Server	N	N/4 Quotes Server addresses
9 [RFC2132]	LPR Server	N	N/4 Printer Server addresses
10 [RFC2132]	Impress Server	N	N/4 Impress Server addresses
11 [RFC2132]	RLP Server	N	N/4 RLP Server addresses
12 [RFC2132]	Hostname	N	Hostname string
13 [RFC2132]	Boot File Size	2	Size of boot file in 512 byte chunks

14 [RFC2132]	Merit Dump File	N	Client to dump and name the file to dump it to
15 [RFC2132]	Domain Name	N	The DNS domain name of the client
16 [RFC2132]	Swap Server	N	Swap Server addeess
17 [RFC2132]	Root Path	N	Path name for root disk
18 [RFC2132]	Extension File	N	Path name for more BOOTP info
19 [RFC2132]	Forward On/Off	1	Enable/Disable IP Forwarding
20 [RFC2132]	SrcRte On/Off	1	Enable/Disable Source Routing
21 [RFC2132]	Policy Filter	N	Routing Policy Filters
22 [RFC2132]	Max DG Assembly	2	Max Datagram Reassembly Size
23 [RFC2132]	Default IP TTL	1	Default IP Time to Live
24 [RFC2132]	MTU Timeout	4	Path MTU Aging Timeout
25 [RFC2132]	MTU Plateau	N	Path MTU Plateau Table
26 [RFC2132]	MTU Interface	2	Interface MTU Size
27 [RFC2132]	MTU Subnet	1	All Subnets are Local
28 [RFC2132]	Broadcast Address	4	Broadcast Address
29 [RFC2132]	Mask Discovery	1	Perform Mask Discovery
30 [RFC2132]	Mask Supplier	1	Provide Mask to Others
31 [RFC2132]	Router Discovery	1	Perform Router Discovery
32 [RFC2132]	Router Request	4	Router Solicitation Address
33 [RFC2132]	Static Route	N	Static Routing Table
34 [RFC2132]	Trailers	1	Trailer Encapsulation
35 [RFC2132]	ARP Timeout	4	ARP Cache Timeout
36 [RFC2132]	Ethernet	1	Ethernet Encapsulation
37 [RFC2132]	Default TCP TTL	1	Default TCP Time to Live
38 [RFC2132]	Keepalive Time	4	TCP Keepalive Interval
39 [RFC2132]	Keepalive Data	1	TCP Keepalive Garbage
40 [RFC2132]	NIS Domain	N	NIS Domain Name
41 [RFC2132]	NIS Servers	N	NIS Server Addresses
42 [RFC2132]	NTP Servers	N	NTP Server Addresses
43 [RFC2132]	Vendor Specific	N	Vendor Specific Information
44 [RFC2132]	NETBIOS Name Srv	N	NETBIOS Name Servers

45	NETBIOS Dist Srv	N	NETBIOS Datagram Distribution
[RFC2132]			
46	NETBIOS Node Type	1	NETBIOS Node Type
[RFC2132]			
47	NETBIOS Scope	N	NETBIOS Scope
[RFC2132]			
48	X Window Font	N	X Window Font Server
[RFC2132]			
49	X Window Manager	N	X Window Display Manager
[RFC2132]			
50	Address Request	4	Requested IP Address
[RFC2132]			
51	Address Time	4	IP Address Lease Time
[RFC2132]			
52	Overload	1	Overload "sname" or "file"
[RFC2132]			
53	DHCP Msg Type	1	DHCP Message Type
[RFC2132]			
54	DHCP Server Id	4	DHCP Server Identification
[RFC2132]			
55	Parameter List	N	Parameter Request List
[RFC2132]			
56	DHCP Message	N	DHCP Error Message
[RFC2132]			
57	DHCP Max Msg Size	2	DHCP Maximum Message Size
[RFC2132]			
58	Renewal Time	4	DHCP Renewal (T1) Time
[RFC2132]			
59	Rebinding Time	4	DHCP Rebinding (T2) Time
[RFC2132]			
60	Class Id	N	Class Identifier
[RFC2132]			
61	Client Id	N	Client Identifier
[RFC2132]			
62	Netware/IP Domain	N	Netware/IP Domain Name
[RFC2242]			
63	Netware/IP Option	N	Netware/IP sub Options
[RFC2242]			
64	NIS-Domain-Name	N	NIS+ v3 Client Domain Name
[RFC2132]			
65	NIS-Server-Addr	N	NIS+ v3 Server Addresses
[RFC2132]			
66	Server-Name	N	TFTP Server Name
[RFC2132]			
67	Bootfile-Name	N	Boot File Name
[RFC2132]			
68	Home-Agent-Addrs	N	Home Agent Addresses
[RFC2132]			
69	SMTP-Server	N	Simple Mail Server Addresses
[RFC2132]			
70	POP3-Server	N	Post Office Server Addresses
[RFC2132]			
71	NNTP-Server	N	Network News Server Addresses
[RFC2132]			
72	WWW-Server	N	WWW Server Addresses
[RFC2132]			
73	Finger-Server	N	Finger Server Addresses
[RFC2132]			
74	IRC-Server	N	Chat Server Addresses
[RFC2132]			
75	StreetTalk-Server	N	StreetTalk Server Addresses
[RFC2132]			
76	STDA-Server	N	ST Directory Assist.
Addresses[RFC2132]			

77	User-Class	N	User Class Information
[RFC3004]			
78	Directory Agent	N	directory agent information
[RFC2610]			
79	Service Scope	N	service location agent scope
[RFC2610]			
80	Naming Authority	N	naming authority
81	Client FQDN	N	Fully Qualified Domain Name
[DRAFT-IETF-DHC-FQDN-OPTION]			
82	Relay Agent Information	N	Relay Agent Information
[RFC3046]			
83	Agent Remote ID	N	Agent Remote ID
84	Agent Subnet Mask	N	Agent Subnet Mask
85	NDS Servers	N	Novell Directory Services
[RFC2241]			
86	NDS Tree Name	N	Novell Directory Services
[RFC2241]			
87	NDS Context	N	Novell Directory Services
[RFC2241]			
88	IEEE 1003.1 POSIX	N	IEEE 1003.1 POSIX Timezone
89	FQDN	N	Fully Qualified Domain Name
90	Authentication	N	Authentication
[RFC3118]			
91	Vines TCP/IP	N	Vines TCP/IP Server Option
92	Server Selection	N	Server Selection Option
93	Client System	N	Client System Architecture
94	Client NDI	N	Client Network Device Interface
95	LDAP	N	Lightweight Directory Access
Protocol			
96	IPv6 Transitions	N	IPv6 Transitions
97	UUID/GUID	N	UUID/GUID-based Client Identifier
98	User-Auth	N	Open Group's User Authentication
[RFC2485]			
99	Unassigned		
100	Printer Name	N	Printer Name
101	MDHCP	N	DHCP multicast address
102-107	REMOVED/Unassigned		
108	Swap Path	N	Swap Path Option
109	Unassigned		
110	IPX Compatability	N	IPX Compatability
111	Unassigned		
112	Netinfo Address	N	NetInfo Parent Server Address
113	Netinfo Tag	N	NetInfo Parent Server Tag
114	URL	N	URL
115	Failover	N	DHCP Failover Protocol
116	Auto-Config	N	DHCP Auto-Configuration
[RFC2563]			
117	Name Service Search	N	Name Service Search
[RFC2937]			
118	Subnet Selection Option	4	Subnet Selection Option
[RFC3011]			
119	Domain Search	N	DNS domain search list
[RFCAB0B]			
120	SIP Servers DHCP Option	N	SIP Servers DHCP Option
[RFC3361]			
121	Classless Static Route	N	Classless Static Route Option
[RFCCSR7]			
	Option		
122-125	Unassigned		
126	Extension	N	Extension
127	Extension	N	Extension
128-254	Private Use		
255	End	0	None
[RFC2132]			

DHCP Message Type 53 Values

Value	Message Type	Reference
-----	-----	-----
1	DHCPDISCOVER	[RFC2132]
2	DHCPOFFER	[RFC2132]
3	DHCPREQUEST	[RFC2132]
4	DHCPDECLINE	[RFC2132]
5	DHCPACK	[RFC2132]
6	DHCPNAK	[RFC2132]
7	DHCPRELEASE	[RFC2132]
8	DHCPINFORM	[RFC2132]
9	DHCPFORCERENEW	[RFC3203]

Network/IP Option Type 63 Sub-Option Codes

Value	Message Type	Reference
-----	-----	-----
1	NWIP_DOES_NOT_EXIST	[RFC2242]
2	NWIP_EXIST_IN_OPTIONS_AREA	[RFC2242]
3	NWIP_EXIST_IN_SNAME_FILE	[RFC2242]
4	NWIP_EXIST_BUT_TOO_BIG	[RFC2242]
5	NSQ_BROADCAST	[RFC2242]
6	PREFERRED_DSS	[RFC2242]
7	NEAREST_NWIP_SERVER	[RFC2242]
8	AUTORETRIES	[RFC2242]
9	AUTORETRY_SECS	[RFC2242]
10	NWIP_1_1	[RFC2242]
11	PRIMARY_DSS	[RFC2242]

DHCP Agent Sub-Option Codes per [RFC3046]

Code	Sub-Option Description	Reference
-----	-----	-----
1	Agent Circuit ID Sub-option	[RFC3046]
2	Agent Remote ID Sub-option	[RFC3046]
3	Sub-option 3 is reserved and should not be assigned at this time; proprietary and incompatible usages of this sub-option value have been seen limited deployment.	[Droms]
4	DOCSIS Device Class Suboption	[RFC3256]

REFERENCES

-
- [RFC951] Croft, B., and J. Gilmore, "BOOTSTRAP Protocol (BOOTP)", RFC-951, Stanford and SUN Microsystems, September 1985.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", Bucknell University, March 1997.
- [RFC2132] Alexander, S., and R. Droms, "DHCP Options and BOOTP Vendor Extensions", Silicon Graphics, Inc., Bucknell University, March 1997.
- [RFC2241] D. Provan, "DHCP Options for Novell Directory Services", RFC 2241, November 1997.

- [RFC2242] R. Droms and K. Fong, "NetWare/IP Domain Name and Information", RFC 2242, November 1997.
- [RFC2485] S. Drach, "DHCP Option for The Open Group's User Authentication Protocol", RFC 2485, January 1999.
- [RFC2489] Droms, R., "Procedure for Defining New DHCP Options", BCP 29, RFC 2489, January 1999.
- [RFC2563] R. Troll, "DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients", RFC 2563, May 1999.
- [RFC2610] C. Perkins and E. Guttman, "DHCP Options for Service Location Protocol", RFC 2610, June 1999.
- [RFC2937] C. Smith, "The Name Service Search Option for DHCP", RFC 2937, September 2000.
- [RFC3004] G. Stump, R. Droms, Y. Gu, R. Vyaghrapuri, A. Demirtjis, B. Beser, and J. Privat, "The User Class Option for DHCP", RFC 3004, November 2000.
- [RFC3011] G. Waters, "The IPv4 Subnet Selection Option for DHCP", RFC 3011, November 2000.
- [RFC3046] M. Patrick, "DHCP Relay Agent Information Option", RFC 3046, January 2001.
- [RFC3118] R. Droms and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC3203] Y. T'Jones, C. Hublet, and P. De Schrijver, "DHCP Reconfigure Extension", RFC 3203, December 2001.
- [RFC3256] D. Jones and R. Woundy, "The DOCSIS Device Class DHCP Relay Agent Information Sub-option", RFC 3256, April 2002.
- [RFCAB0B] B. Aboba and S. Cheshire, "DHCP Domain Search Option", RFC XXXX, Month Year.
- [RFC3361] H. Schulzrinne, "DHCPv4 Option for SIP Servers", RFC 3361, August 2002.
- [RFCCSR7] T. Lemon, S. Cheshire, and B. Volz, "The Classless Static Route Option for DHCPv4", RFC XXXX, Month Year.

PEOPLE

[Droms] Ralph Droms, <rdroms@cisco.com>, January 2001.

[]

18.3 Fragen zum Inhalt

Fragen zur Prüfungsvorbereitung:

Die folgenden Fragen sollen vor allem einen Überblick über den möglichen Fragestil geben, nicht so sehr einen vollständigen Katalog für eine Prüfung. Es sind Beispielfragen enthalten, die aus Praktikumsversuchen stammen, die in Ihrem Semester evtl gar nicht durchgeführt wurden.

Grundsätzlich müssen nicht alle Fragen beantwortet werden, um die volle Punktzahl zu erhalten, die Prüfung wird einen Fragenüberhang von ca 10% beinhalten.

OSI Referenzmodell

Vervollständigen Sie das folgende Bild des OSI-Referenzmodells. Sie können dabei die deutschen oder englischen Begriffe verwenden

Application Layer
Data Link Layer

In welche beiden Teilschichten kann man die Data Link Layer unterteilen?

Codierung

Was ist das wesentliche Kennzeichen von Codierungsverfahren wie Manchester bzw. Diff. Manchester?

Bei FDDI wird eine andere Art der Codierung verwendet. Welche? Was ist der wesentliche Unterschied zu Manchester bzw. Diff. Manchester Code?

Performance

Mit welchem Programm/Tool kann man die Auslastung/Utilization eines Netzwerksegments messen bzw. anzeigen lassen.

Directory Services

Welches sind die beiden wesentlichen Unterschiede einer Bindery eines Fileservers und Directory Services? Welcher grundsätzliche Unterschied besteht für den Benutzer beim Einloggen in einen einzelnen Fileserver gegenüber einem Netzwerk mit Directory Services?

Nennen Sie 5 verschiedene Blattobjekte (leaf objects) in der Novell NDS.

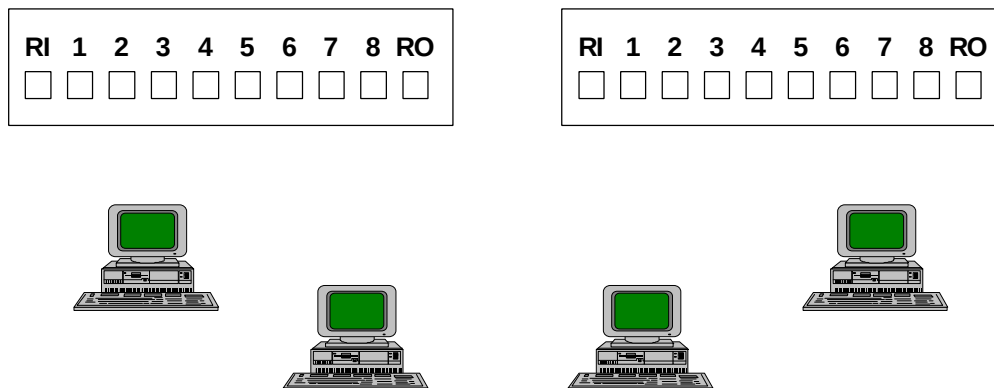
Standardprotokolle

Wieviele Subnetze kann man in einem Netz mit einer Class C Adresse maximal bilden, wenn an allen Stationen die Subnetzmaske FF.FF.FF.F0 verwendet wird.

Erklären Sie kurz den Verwendungszweck des ARP Protokolls.

Token Ring

4a. Vervollständigen Sie in nachfolgender Skizze den Aufbau eines Token Ring Netzes mit zwei Ringleitungsverteilern und vier Netzwerkstationen:

**LAN Zugriffsverfahren**

Beschreiben Sie in drei Schritten die Funktionsweise des CSMA/CD Zugriffsverfahrens beim Senden.

Ordnen Sie die folgenden Merkmale und Begriffe den Zugriffsverfahren Ethernet (802.3), Token Ring (802.5) und FDDI zu. Kreuzen Sie dazu entsprechend an (Mehrfachnennungen sind möglich):

	Ethernet	Token Ring	FDDI/CDDI
Deterministisches Verfahren			
Class A (bzw. DAS) -Station			
Eine Station als Active Monitor			
Manchester Codierung			
Twisted Pair Verkabelung möglich			
Ringleitungsverteiler			
Transceiver bzw. Medium Attachment Unit			
Kollisionserkennung			

Router

Welche der folgenden Protokolle werden zwischen Routern verwendet, um interne Routingtabellen abzugleichen: IPX, IP, RIP, OSPF, SNA, Netbios, TCP?

Welcher Ebene des OSI Modells sind diese Routingprotokolle am ehesten zuzuordnen?

Weitere Fragen (ohne Unterlagen):

1. Wofür steht der Begriff MAN?
2. Welcher Ebene im OSI Referenzmodell könnte das TCP Protokoll am ehesten zugeordnet werden?
3. Welches sind die wichtigsten, typischen Aufgaben von Protokollen der Netzwerkschicht?
4. Nenne Sie zwei wichtige Kriterien zur Bewertung von Leitungscodes zur Datenübertragung
5. Was versteht man unter dem Begriff "selbsttaktender Code"?
6. Wie hängt die Transferrate in Ethernet und Token Ring-Netzen von der Paketgröße ab?
7. Wofür steht der Begriff "10 Base 5" und wofür stehen die 10, das Base und die 5?

8. Wodurch unterscheiden sich Ethernet 802.3 und 802.2 Frame Formate?
9. Wie groß (Wieviele Byte) ist ein Token im 802.5 Token Ring?
10. Erklären Sie kurz die Funktionsweise der Datenübertragung im 802.5 Token Ring
11. Erklären Sie den Begriff "Early Token Release"
12. Durch welches Protokoll wird bei transparenten Brücken verhindert, daß im Netz redundante Wege entstehen
13. Welches der genannten Protokolle ist nicht routbar: IP, IPX, SNA, DECNET
14. Wodurch unterscheiden sich in der Funktionsweise Transparente und Source-Routing Brücken?
15. Wie kann die Ausfallsicherheit von Servern im Hinblick auf Festplattenausfall verbessert werden? Nennen Sie 3 Möglichkeiten
16. Was ist ein Alias in Directory Services
17. Das Standardmanagement Modell der ISO sieht fünf funktionale Bereiche für Netzwerkmanagementaufgaben vor. Nennen Sie drei davon.
18. Wodurch unterscheidet sich im wesentlichen die Netzwerkadressierung in IPX bzw. TCP/IP Netzen.
19. Wie wird in IP und IPX Netzen verhindert, daß sich Pakete endlos im Netz befinden können
20. Wodurch unterscheidet sich die Kommunikation in SNA Netzen grundlegend von der in PC-Netzen

18.4 Literatur

/1/ Larry L. Peterson, Bruce S. Davie: Computernetze, dpunkt.verlag, 2000

/2/ Anatol Badach, Erwin Hoffmann: Technik der IP-Netze, Hanser Verlag, 2001

/3/ Andrew S. Tanenbaum: Computernetzwerke, Prentice Hall, 1997