

# Rechnernetze 2006

Prof. Dr. Konrad Froitzheim

## • Geschichtliches

schon lange Semaphore

1835 Morse-Telegraph

1861 Telephonie (Reis), 1874 Bell

1865 1. Transatlantikkabel

1891 Automatischer Drehwähler (A. B. Strowger)

1895 Funk (Marconi)

1923 Rundfunk als erster Broadcast-Dienst

1929 Koaxialkabel

1930 Multiplextechnik: Mehrfachnutzung von Leitungen

1936 Fernsehen

1948 Puls Code Modulation

1964 Nachrichtensatelliten

1964 AT&T's PicturePhone, Markteinführung 1969

1966 Glasfaserkabel (Idee C.V. Boys, 1881!)

1980 Internet

1994 WWW

## Thema

- Technische Kommunikation
  - Rechnerkommunikation
  - Datenaustausch
  - technische Kommunikation zwischen Menschen
  - Sprache, Video, ... => Multimediakommunikation
- Stichworte
  - Telefon, Telefax, ...
  - Fernsehen
  - Mobilfunk, Satelliten
  - Lokale Netzwerke
  - Filetransfer, E-Mail
  - Internet
  - Protokolle

## Inhaltsübersicht

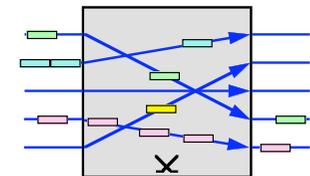
### 1. Überblick und Taxonomie

- 1.1 Informationsbegriff (WH)
- 1.2 Modelle der Kommunikation
- 1.3 Dienstebegriff
- 1.4 Verkehrsbeschreibung
- 1.5 Verbindungen und Topologie
- 1.6 Namen und Adressen
- 1.7 Standards



### 2. Vermittlungsdienste

- 2.1 Bit- und Bytevermittlung (WH)
- 2.2 Zellvermittlung (WH)
- 2.3 Paketvermittlung

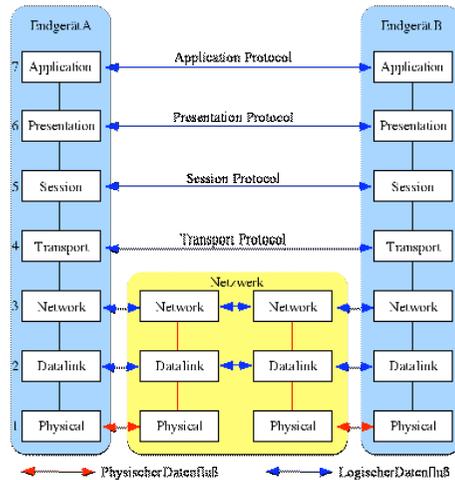
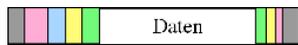




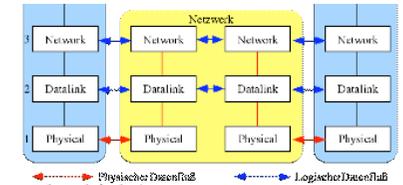
## 1.2 Modelle der Kommunikation

### 1.2.1 OSI

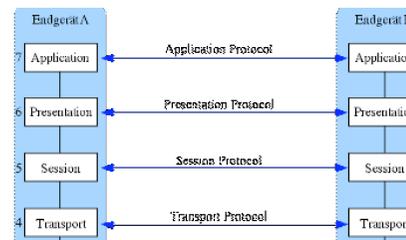
- Open Systems
- definierte Schnittstellen
  - Service Access Point (SAP)
  - Service Data Units
  - Request
  - Response
  - Confirmation
  - Indication
- Peer Protocols
  - Protocol Data Units
  - Verpackung



- Netzwerkabhängige Schichten
- Übertragung (physical layer)
  - ungesicherte Übertragung der Signale über eine Leitung
  - Kanalkodierung, Modulation
  - Bsp: Leitungskodierung
- Verbindungsschicht (data link, Sicherungsschicht)
  - Rahmenbildung (frames)
  - Prüfsummen zur Fehlerentdeckung
  - evtl. Fehlerbehebung
  - Flußkontrolle
  - Verstopfungskontrolle / Lastabwehr
  - Bsp: HDLC
- Vermittlungsschicht (network layer, Netzwerkschicht)
  - Übermittlung zum Kommunikationspartner
  - Routing (Weglenkung)
  - Flußkontrolle
  - Verstopfungskontrolle / Lastabwehr
  - IP, X.25



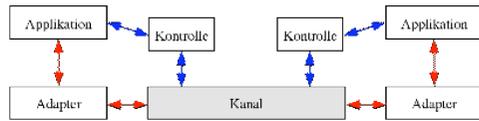
- Netzwerkunabhängige Schichten
- Transportschicht (transport layer)
  - gesicherte Nachrichtenübertragung
  - Adressierung, Numerierung, Prüfung
  - Flußkontrolle
  - Ende-Ende Fehlerbehebung
  - Bsp: TCP, XTP, TP4
- Verbindungssteuerung (session layer)
  - Datenstrom
  - Verbindungsaufbau, Dialogsteuerung
  - Synchronisation und Wiederaufsetzpunkte
  - Bsp: http, ftp, smtp, ...
- Darstellungsschicht (presentation layer)
  - Daten als Informationsobjekte
  - Informationsdarstellung, Kompression und Verschlüsselung
  - Transfersyntax zur Datenbeschreibung
  - Bsp: html, sgml, ASN.1, ...
- Verarbeitungsschicht (application layer)
  - benutzende Prozesse



- Leistungsprobleme
  - Schnittstelleninflation
  - Marshalling-Overhead und Prozeduraufruf
  - Prozesswechsel
  - schichtinterne Protokolle
  - Verpackungsmüll
- konzeptuelle Probleme
  - Einteilung in Schichten fraglich
  - Funktionsduplizierung
  - Konvertierung der Parameter evtl. verlustbehaftet
  - Semantikverlust im Stack
  - Schichtentimeouts
- Beispiel MAP/TOP, ALF
- Sinnvoll evtl. für Entwurf und Verifikation

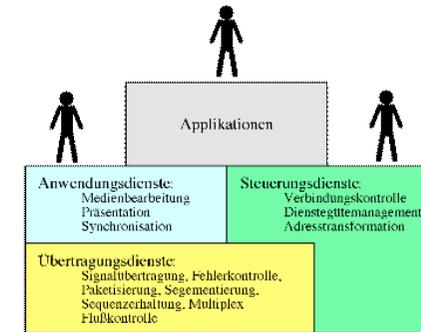
## 1.2.2 Komponentenmodell

- Komponenten der Kommunikation
  - frei nach Shannon



- Potenzmenge bilden:
  - 16 Mengen
  - Bsp: {Kanal, Kontrolle, Adapter, Applikation}
- sinnvolle Mengen identifizieren. Drei bleiben übrig:
  - {Kanal, Kontrolle, Adapter} - Übertragungsdienste
  - {Kanal, Kontrolle, Adapter, Applikation} - Anwendungsdienste
  - {Kontrolle, Adapter, Applikation} - Steuerungsdienste
- Weitere Unterteilung
  - Übertragungsdienste = Übertragung und Vermittlung

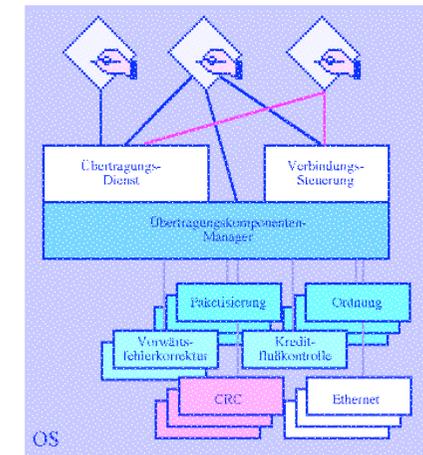
- Anordnung der Bereiche



- Übertragungsdienste
  - Signalübertragung
  - Vermittlung
  - Protokolle

- Steuerungsdienste
  - Verbindungsaufbau und -abbau, Features
  - Netzzugang (Autorisierung, ...)
  - Netzwerkmanagement?
- Anwendungsdienste
  - kennen Medien
  - End-zu-End Protokolle (http, ftp, ...)
  - konsumptive und kooperative Dienste (WWW, Telefon)
  - Anwendungselemente (RPCs, RTP, ...)
  - Verzeichnisdienste (DNS, X.500)
  - Endgeräte (Funktelefon)
- Applikationen
  - kennen Semantik
  - verwenden Anwendungselemente und Anwendungsdienste
  - Textprogramme, CSCW, ...

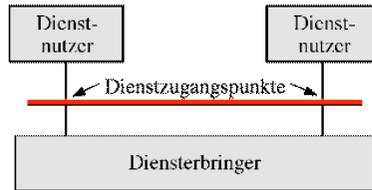
- Baukastenprinzip
    - mehrere Abstraktionsebenen
    - Mechanismenklassen
  - Klassen
    - Protokollfunktionen
    - Fehlerbehandlung
    - Flußkontrolle
    - Lastkontrolle
    - Traffic-Shaping
- => konfigurierbare Dienste



### 1.3 Dienstebegriff

Dienst ist eine Menge von Funktionen, die einem *Benutzer* von einem *Erbringer* zur Verfügung gestellt werden

- Funktionen unterschiedlicher Abstraktion je nach Dienst
  - Dateiübertragung, Datentransport, Read, Write, ...
  - Verbindungsaufbau, - abbau, ...
  - SendPDU(packType, parameter)
- Service Access Points SAP



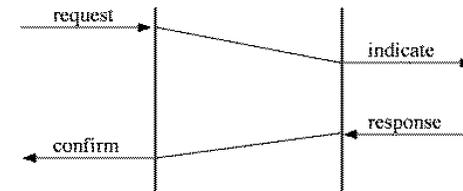
- Dienstschnittstelle, Dienstzugangspunkt
- im OSI-Modell für jede Schicht (X-SAP)

- Dienstegüte (Quality of Service, QoS)
  - Geschwindigkeit, Bandbreite, Fehler
  - Wahrscheinlichkeit des Verbindungsaufbaus
  - Anbieter verspricht bestimmte Dienstigenschaften
- Dienstenutzer erwarten bestimmte Eigenschaften
  - QoS an der Benutzungsschnittstelle => Akzeptanz des Dienstes
  - Medien und Interaktivität bestimmen Anforderungen

Die Dienstegüte ist eine Menge quantitativer Kenngrößen, die die Eigenschaften eines Dienstes beschreibt.

- Abbildung an Schnittstellen
  - Programmabsturz, Rauschen -> Bitfehler
  - Knacken, Krachen -> Paketverlust
  - Bildrate -> Durchsatz, Ruckeln -> Jitter

- Dienstprimitive nach OSI



- Bezeichnung: Schicht-Objekt.Primitiv
- Bsp: T-DATA.request  
N-CONNECT-indication

- Verbindungsorientiert

- X-CONNECT.request      X-CONNECT.confirm
- X-DATA.request          X-DATA.confirm
- X-DISCONNECT.request   X-DISCONNECT.confirm

- Verbindungslos

- X-UNITDATA.request(Quell-, Zieladresse, Dienstegüte, Daten)
- X-UNITDATA.indication(Quell-, Zieladresse, Dienstegüte, Daten)

### 1.3.1 Dienstegüteparameter

#### 1.3.1.1 Verkehrscharakteristik

- Übertragungsgeschwindigkeit, Bitrate, Bandbreite?
- Schwankungen der Bitrate
- Problem mit Bitrate: nicht alles bitseriell
- Router, Puffer, Protokolle etc.
- Paketrate, Paketgröße, Paketoverhead
- Allgemein:

Durchsatz ist das Verhältnis der Größe einer SDU zu der Zeit, bis die nächste SDU übertragen werden muß (kann).

- Beispiele

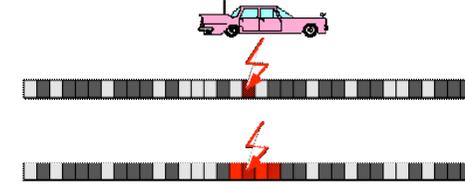
- $1 \text{ bit} / (1 \text{ s} / 64.000) = 64.000 \text{ bit/s} = 1 \text{ byte} / (1 \text{ s} / 8.000)$
- $1500 \text{ byte} / (1 \text{ s} / 100) = 150.000 \text{ byte/s} = 1.2 \text{ Mbit/s}$

- statistische Angaben problematisch
  - Zeitraum der Messung unbekannt/unterschiedlich
  - => Leaky Bucket etc.
- Transit-Verzögerung (Delay)
  - Zeit zum Durchlaufen des Netzes und der Übertragungsprotokolle
  - pro SDU
- Verzögerungsschwankung (Delay Jitter)
  - Schwankungen der Durchlaufzeit
- Besser:
  - Mittelwert (Erwartungswert) = Delay
  - Schwankung (Varianz) = Delay-Jitter
- Oder:
  - minimale und maximale Verzögerung einer SDU
- Anzahl SDUs pro Zeiteinheit schwankt
  - Kompression
  - Burstiness

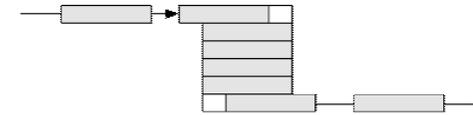
- Fehlererkennung
  - Paketverlust, Paketduplizierung oder Datenkorrumpierung
  - Prüfsummen auf mehreren Schichten
  - Sequenznummern
- Zuverlässigkeitsinformation vom Kanaldekoder
  - Soft-Output (Viterbi)
  - Wahrscheinlichkeit pro bit:  $P(0)$ ,  $P(1)$
  - Auswertung durch Mediendekoder bzw. höhere Schichten
- Fehlerbehebung durch Wiederübertragung
  - Retransmission
  - selective Retransmit
  - Bestätigungen
    - Automatisch vom Empfänger (positiv, negativ)
    - auf Anfrage vom Sender
  - Zeitüberwachung (timeout)

### 1.3.1.2 Zuverlässigkeit

- Fehlerarten
  - Bitfehler einzeln oder als Burst



- Paketverlust (Prüfsumme falsch, Pufferüberlauf)



- Fehlerwahrscheinlichkeit
  - Bitfehler  $P(F_B)$  aus Signal-Rauschverhältnis
  - $P(F_{PDU}) = P(F_B) * \text{Länge}_{PDU} + P(\text{Pufferüberlauf})$

- Problem Retransmissions bei großen Multicastgruppen

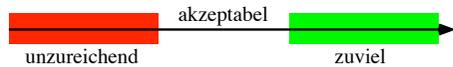
Eine Verbindung zu  $n$  Empfängern ist **k-zuverlässig** ( $0 \leq k \leq n$ ), falls für jede Dateneinheit gilt, daß sie bei mindestens  $k$  Empfängern angekommen ist.

- Unbekannte Gruppen?
- k-deterministisch-zuverlässig
  - bei  $k$  spezifizierten Mitglieder angekommen
- Ablieferungszeitpunkt in Mehrpunkttopologien
- Voraus-Fehlerkorrektur
  - Fehlerkorrigierende Codes
  - Blockcodes: Reed-Solomon
  - Faltungscodes
- Wechselwirkung mit Verkehrscharakteristik
  - Burst bei Retransmission
  - höhere angebotene Last durch Voraus-Fehlerkorrektur

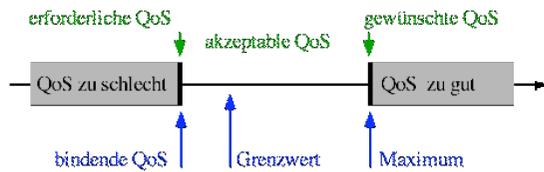
### 1.3.1.3 Transaktionsbeschreibung ...

### 1.3.2 Dienstgütesemantik

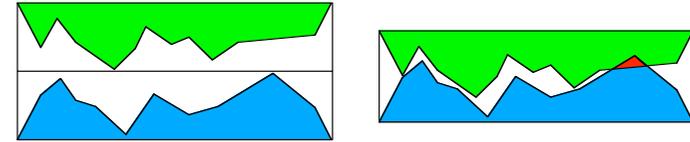
- Best Effort
  - LANs, Internet
- QoS an der Benutzungsschnittstelle => Akzeptanz des Dienstes
- Medien und Interaktivität bestimmen minimale Anforderungen
- Aufwand (Preis) bestimmt maximale Anforderung



- Aus der Sicht der Kontrahenden



- Garantierte Dienstgüte
  - Pessimistische Annahmen Systemverhalten
  - Sehr zuverlässig
  - Überreservierung => schlechte Ressourcennutzung
  - Bsp. Sprachkanal im Telefonsystem



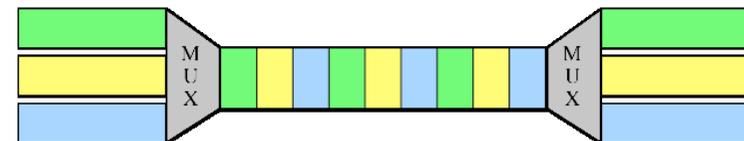
- Statistische Dienstgüte
  - Stochastische Ausnutzung der angeforderten Dienstgüte
  - keine Garantie ...
  - Bsp. Verbindungsmanagement im Telefonsystem
- Dienstgütemanagement im Netz schwierig
  - Überwachung
  - weitere Zustände
  - 'Rerouting'

### 1.3.3 QoS-Anforderungen

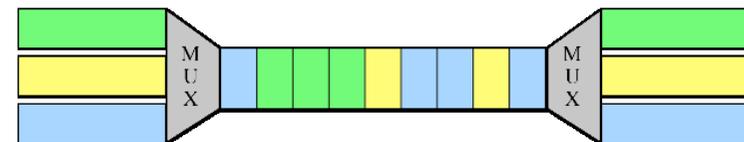
Medium	QoS-Parameter	Werte	Qualitätsvergleich
Audio	Verzögerung	≤ 200 ms	Satellitenverbindung
	Fehler	≤ 10 <sup>-2</sup> Pakete	Internet 10 <sup>-1</sup>
	Durchsatz	16 kbit/s	Telefon-ADPCM
Video		192 kbit/s	CD
	Verzögerung	≤ 200 ms	
	Fehler	≤ 10 <sup>-2</sup> Pakete	unkomprimiert
		≤ 10 <sup>-11</sup>	MPEG-komprimiert
Text	Durchsatz	1,5 Mbit/s	Videorekorder komprimiert
		140 Mbit/s	TV unkomprimiert
	Fehler	≤ 10 <sup>-3</sup>	
Grafik	Durchsatz	14 kbit/s	interaktiv
	Fehler	≤ 10 <sup>-3</sup>	unkomprimiert
Bild	Durchsatz	28 kbit/s	interaktiv
	Fehler	≤ 10 <sup>-2</sup>	unkomprimiert
		≤ 10 <sup>-11</sup>	JPEG-komprimiert
	Durchsatz	64 kbit/s	JPEG, interaktiv

### 1.4 Verkehrsabschätzung

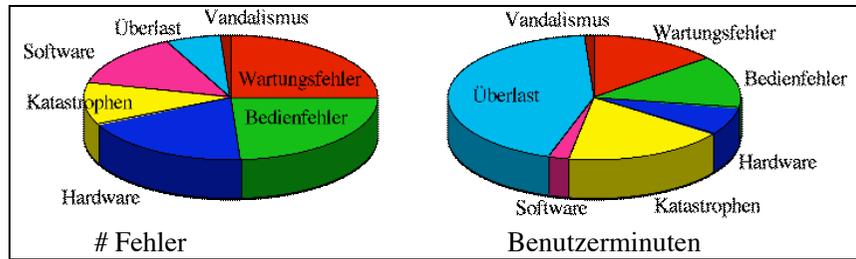
- gemeinsam genutzte Ressourcen: Leitungen, Vermittlungen
  - => Multiplex
- feste Multiplex-Schemata
  - Verbindungen im Telefonsystem, ISDN-Teilnehmeranschluß
  - Ressourcenverschwendung



- statistischer Multiplex
  - Ethernet, 'call attempts' im Telefonsystem, Internet



• Probleme im US-Telefonsystem 4/92 - 3/94

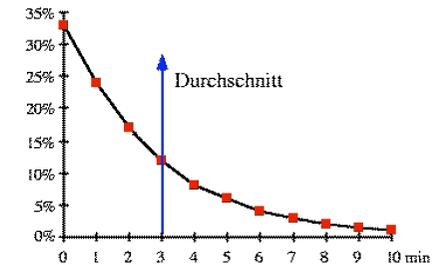


=> traffic engineering

- Abschätzung von Multiplex-Effekten
  - Eigenschaften der Einzelströme
  - zeitliche Kombination
  - Auslegung der Ressourcen (Durchsatz, Puffer, Prozessor, ...)
- Warteschlangen -Theorie (queueing-theory)

• Annahme: wichtige Verteilungen mit Poissonverteilung

- negative Exponentialverteilung
- Nachrichtenlänge
- Ankunftsrate



• Mittelwerte

- $\lambda$  für mittlere Ankunftsrate
- $\mu$  für mittlere Länge/Bearbeitungsdauer

- Angebotene Last = Ankunftsrate\*Dauer [Erlang]
  - 100 [Benutzer] \* 3 Anrufe/Stunde \* 0,05 Stunden pro Anruf
  - = 15 Erlang

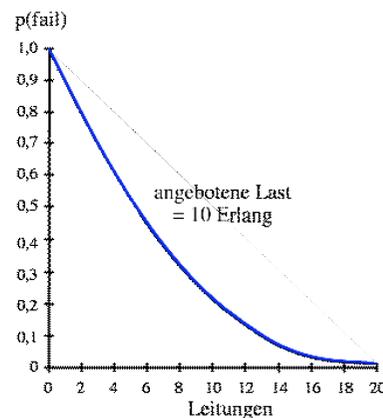
• A.K. Erlang 1918

- Anzahl der Anrufe poisson-verteilt
- Dauer der Anrufe poisson-verteilt

=> Blockierwahrscheinlichkeit

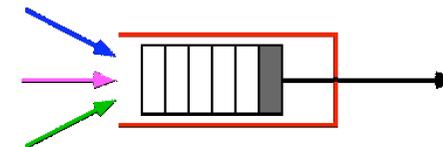
$$B(a, n) = \frac{a^n}{n! \sum_{i=0}^n \frac{a^i}{i!}} = \frac{a * B(a, n-1)}{n + a * B(a, n-1)}$$

- a angebotene Last in Erlang
- n Leitungen
- iterieren bis B(a,n) klein genug



• Warteschlangen

- ankommende Nachrichten kommen in Queue
- werden einzeln abgearbeitet



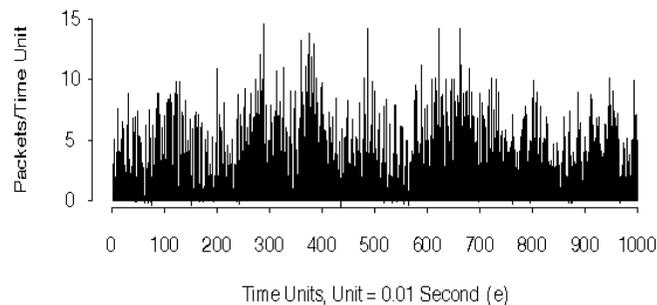
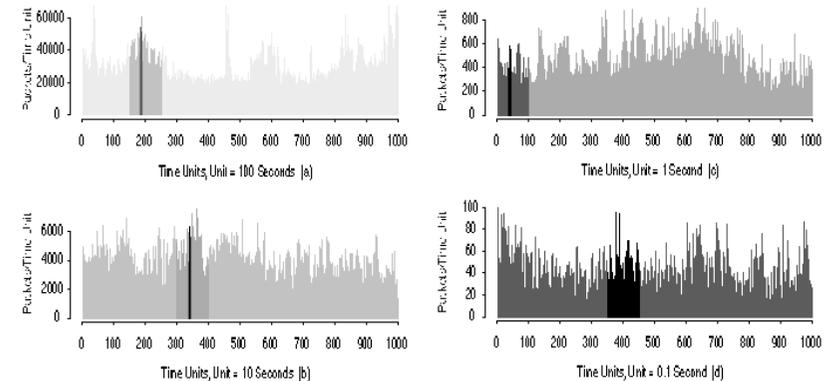
• Anzahl Schlangen vs. Anzahl Verarbeitungseinheiten

• Warteschlangenmodelle

- Ankunftsmodell/Dienstdauermodell/# Server - Länge Warteschlange
- M = gedächtnisloses Modell (memoryless, Poissonverteilung)
- D = deterministisches Modell
- G = allgemeines (general) Modell
- M/M/1 abgeforscht
- G/G/m-k?

- Sind Ankunftsintervalle poissonverteilt?
  - Poisson: große Anzahl unabhängige Benutzer
  - Videokompression, MPEG
  - TCP's Vegas-Algorithmus reagiert auf Verstopfung
  - Netscape-FTP und neue 'Pre-Releases'
- Ist Servicedauer poissonverteilt?
  - FTP-Server für wenige Dateien
  - Telefonauskunft
- Sind Verkehrsquellen unabhängig?
  - Protokolle erzeugen Abhängigkeit (Request-Response)
  - Überlast, Verlust und Paketwiederholungen
  - besondere Ereignisse (Fußballspiel, Sylvester)
- Fraktales Stromverhalten beobachtet
  - MPEG-Video ohne VBV [Garrett, Willinger, 1994]
  - Verteilung in verschieden langen Zeiträumen ähnlich

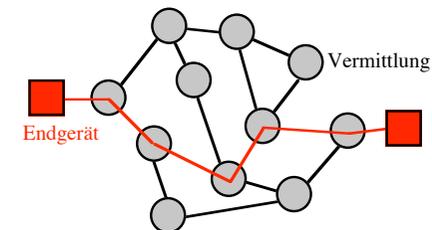
- On the Self-Similar Nature of Ethernet Traffic
  - Messungen über mehrere Jahre in echten Netzen
  - Bellcore Morristown Research Centre
  - [Leland, Taqqu, Willinger, Wilson, 1994]



- Selbstähnlich
  - gleichgültig wie stark vergrößert: Bilder ähnlich
  - burst, flacheres Gebiet, burst
  - Poisson-Modell unzutreffend für Ethernet
- Folgerungen/Maßnahmen
  - selbstähnliche Quellen konstruieren
  - Meßmethoden für 'Burstiness' entwickeln
  - Schluß vom Gesamtverkehr auf Quellen möglich

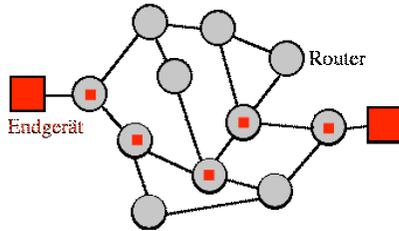
### 1.5 Verbindungen und Topologie

- Verbindung: Abstraktion der Shannon'schen Kanäle
  - Kontext der Informationsübertragung
  - fest
  - statisch geschaltet
  - dynamisch geschaltet
  - virtuell



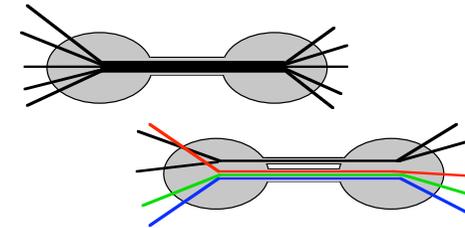
- Verbindungslose Dienste?
  - Dateneinheiten enthalten komplette Zielbeschreibung
  - Erreichen des Zieles ohne Kontext
  - implementierungsbezogene Definition

- Technik verbindungsloser Dienste
  - Weglenkung bei jeder SDU neu
  - ATM wird aber auch als verbindungsorientiert bezeichnet
  - Router verwenden Adress-Cache zur Beschleunigung
  - => Eintrag in caches entlang des Pfades = Kontext



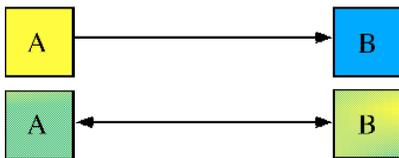
- Benutzungsemantik
  - Warten auf Bestätigung durch Benutzer
  - Kontext um Antworten einzuordnen (Listen von Requests)
  - Protokolle auf höheren Ebenen

- QoS in verbindungslosen Diensten?
  - Identifikation von Paketen als Teil eines 'flows'
  - Reservierung für Flows
  - Reservierungsprotokoll

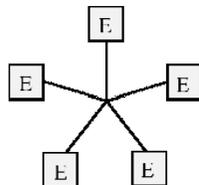


- Verkehrsklassen (Internet, ATM, ...) brauchen Kontext
  - konstante Bitrate
  - variable Bitrate
  - verbleibender Verkehr
- Leitungsvermittlung und Paketvermittlung
- hard state und soft state

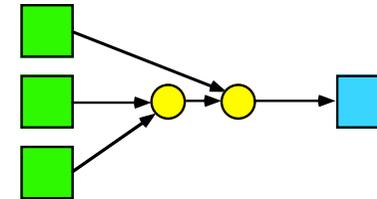
- Verbindungstopologie
  - gerichtet (Funk, Fax, ftp, Telefon, ...)
  - bidirektional (ISDN, V.24, telnet, ...)
  - Richtungsumschaltung möglich



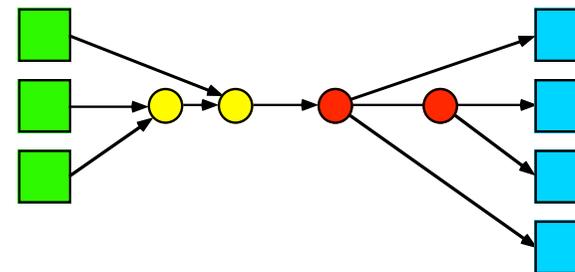
- asymmetrische Verbindung
  - unterschiedliche Dienstegüte je nach Richtung
- Zweipunkt und Mehrpunkt



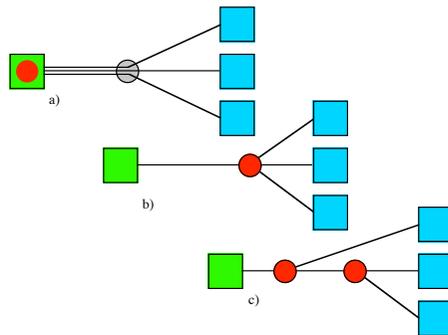
- Unicast n:1



- Multicast n:m
  - n-facher Mehrfachzugriff auf Verbindung

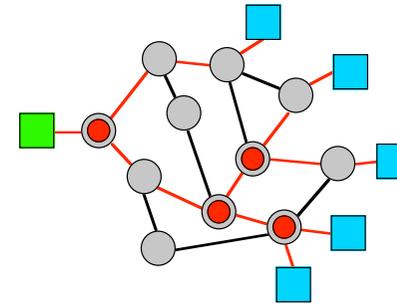


- Nachrichtenreplikation
  - in der Quelle
  - im Netz
  - explizit oder implizit

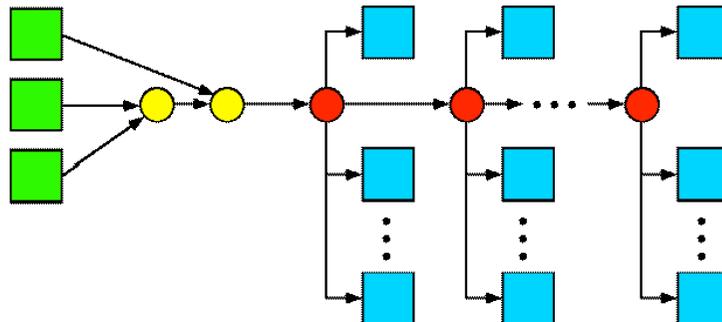


- Gruppenverwaltung
  - statisch/dynamisch
  - deterministisch = vollständiges Topologiewissen in einem Gerät
  - Zugangsregeln: offen oder geschlossen

- Mehrpunkttopologie im Netz
  - normal in LAN's (Ethernet, Token-Ring, PhoneNet, ...)
  - Mehrfachadressen
  - explizite Replikation in vermaschten Punkt-zu-Punkt-Netzen (z.B. Internet)
  - Mehrfachversand



- Broadcast
  - AppleTalk NBP, Ethernet ARP
  - verteiltes Rechnen



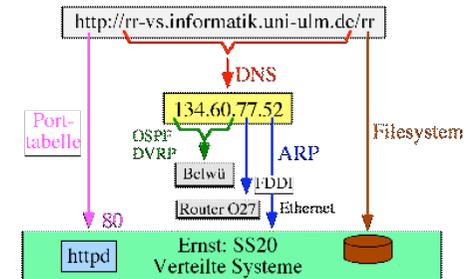
- Skalierbarkeit
  - Mehrstufigkeit
  - Begrenzung der Reichweite
  - Mbone, Usenet-News

- Multiplexen und Mischen
  - Sequentialisierung (Multiplexen)
  - Mischen
- Mehrfachzugriff verursacht Überlagerung
  - Sequentialisierung (Multiplexen)
  - Mischen
- Multiplexen
  - feste Kanalzuordnung (Telefonsystem, SONET, ...)
  - Verfahren mit dynamischer Zuordnung
  - Bsp. Unterhaltung, Konferenz (soziale Protokolle)
- Beispiele dynamischer Verfahren
  - Reservierung mit Anmeldung
  - Token
  - Aloha
  - CSMA mit Varianten
- Mischen = Konferenzschaltung

## 1.6 Namen und Adressen

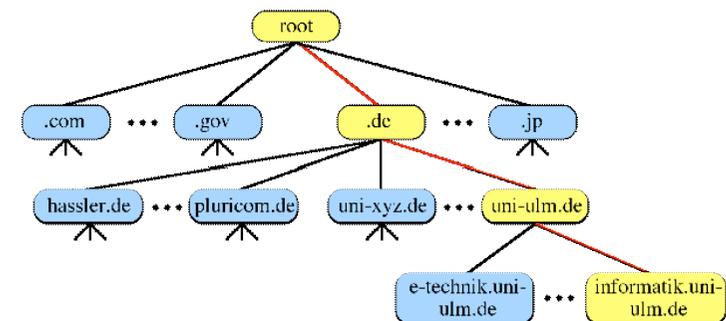
- Endpunktbezeichnung
  - Personen, Dienste, Geräte
  - Postanschrift ist Tupel
  - (Name, Bestimmungsort, Leitangaben, Zustellangaben)
  - menschliche Bezeichnung vs. Weglenkungsinformation
- flache und hierarchische Adressen
  - Bsp. Pass-Nummer vs. DNS-Name
- Besser: 3 Abstraktionsebenen
  - Kommunikationsadressen
  - Netzwerkadressen
  - Anschlußpunktadressen
- Kommunikationsadressen
  - von Menschen benutzbar
  - konzeptuelle Beschreibung
  - (Dienst/Person, Beschreibung)
  - beschreiben Person oder Konzept
  - Bsp. URL

- Netzwerkadressen
  - technische Beschreibung des Kommunikationszieles
  - (Länderziffer, Ortsnetz, lokale Nummer)
  - Bsp.: Telefonnummer, IP-Adresse
- Anschlußpunktadressen (Service Point Access)
  - eindeutig in einem Kontext
  - Hardware-Beschreibung
  - (Gerätenummer, Dienstidentifikation)
- Adreßbindung = Abbildung auf untere Klasse



- **Kommunikationsadressen**
  - (Dienstname, Kontextname, Gerätename)
  - konzeptuelle Beschreibung
  - verständlich für Menschen
- URL Universal Resource Locators/Identifiers
  - Dienst://Rechnername[/Pfad][/Datei]
  - RFC1738
  - http://rr-vs.informatik.uni-ulm.de/rr.html
  - ftp://frodo.informatik.uni-ulm.de/~frz/directrain/delta.zip
  - Ausnahme: mailto:wolf@informatik.uni-ulm.de
- Dienst: ftp, http, ...
- Rechnernamen sind DNS-Namen
- Pfad und Dateiname entsprechend OS-Konvention
- Oft Dienst auch im Namen wiederholt
  - http://www.joeboxer.com
  - ftp://ftp1.apple.com/
  - besser: <dienst>://info.itu.ch

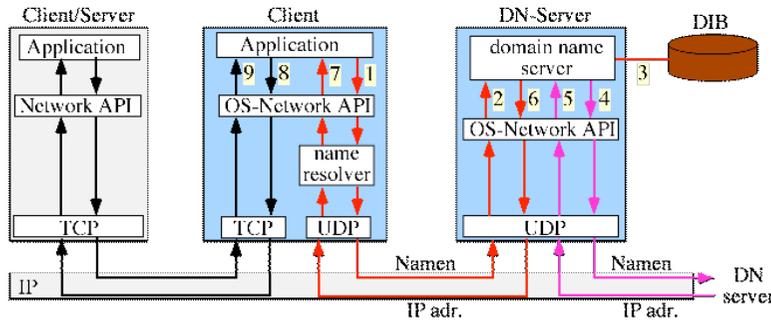
- Domain Name System (DNS) Namen
  - Bsp: info.itu.ch, altavista.digital.de
  - Namenssyntax RFC1034
  - hierarchisches System



- Länderkennzeichen oder .com, .org., .net, .gov, .mil, .edu, .int
- Dauerstreit um Domainnamen
  - McDonalds.com, bau.de
- Neue Domains: .pers, .store, ...

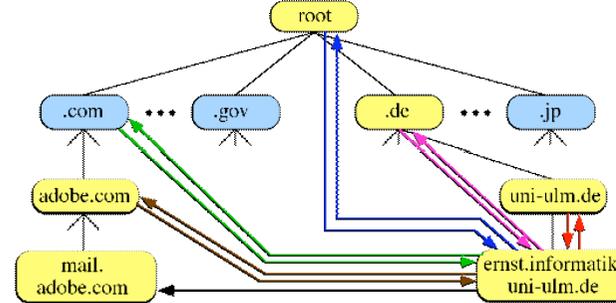
- DNS-Adressbindung

- Anfrage an wohlbekannten Nameserver (IP-Nummer)
- dynamisch: erst beim Verbindungsaufbau
- Auskunft vom DNS aus den Servern statisch
- statische Registrierung
- Domain Information Base (DIB)

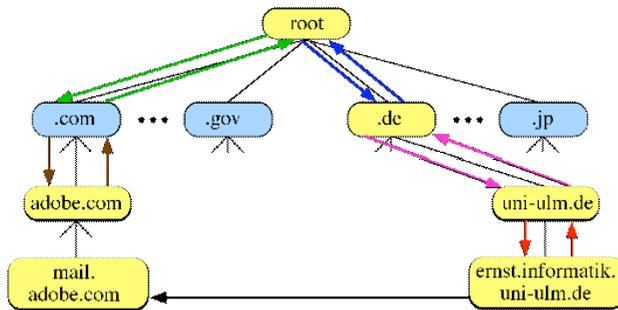


- name resolver

- Software im Klientenrechner
- verwendet TCP/IP
- Verteilter Algorithmus in RFC1035
  - Adressbindung DNS-IP
  - Lokalität der Namensbindung
  - nichtlokale Anfragen: referral
  - iterativ (Nameserver antwortet mit anderer Serveradresse)



- rekursiv (Server fragt andere DNS-Server)



- name cache

- ITU X.500

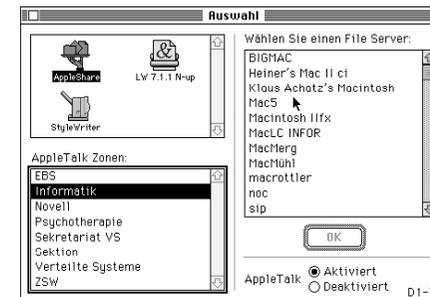
- Objekt hat Attribute (Attributtyp, Wert)
- Directory Information Tree
- Search Port (yellow pages), Read Port (white pages)
- Welt, Nation, Organisation, Unterorganisation, Personen

- AppleTalk Namen

- <object>.<type>.<zone>
- object: frei wählbare Zeichenkette für Knoten
- type: Dienst, z.B. LaserWriter, AFS-Server
- zone: logisches Netzwerksegment

- Name Bindung Protocol NBP

- Resource-Lookup
- Abbildung direkt auf Anschlußpunktadresse
- =,LaserWriter,\* findet alle lokalen PAP-Drucker



• **Netzwerkadressen**

- technische Beschreibung des Kommunikationszieles

• **Nummern-Tupel**

- Dienst (oft implizit durch Geräteauswahl)
- Kontext (Subnetz oft impliziert)
- Subnetz {Subnetz}
- lokals Gerät/Leitung

• **Telefonsystem**

- [<Art>] [<Land>] [<Vermittlung>] [<Teilnehmer>] [<Nebenstelle>]
- Art = {externalFernverbindunginternationalIDienst}
- lokal unterschiedlich
- Anpassung bei mobilen Geräten oft schwierig (siehe GSM)

• **Beispiel Telefonnummer**

-				4146	in der Universität
lokal			502	4146	in Ulm
Fernverbindung		731	502	4146	in Deutschland
international	49	731	502	4146	im Ausland

• statische Aufteilung in USA und Kanada: **415-653-9516**

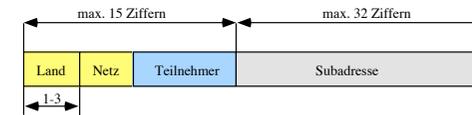
- **area code** bzw. besondere Dienste (800, 900) dreistellig
- **Vermittlungscode** dreistellig
- **Teilnehmer** vierstellig
- PBX (Centrex, virtuelle Nebenstellenanlage) in Nummernplan integriert

• **Deutschland**

- Vermittlungsbereich 2-5-stellig (89, 731, 7336, 34491)
- Rufnummer

• **ITU Norm E.164**

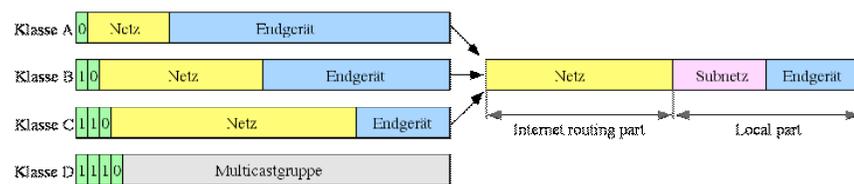
- abgeleitet von Telefonnummern
- Länge durch Wahlbewertungshardware eingeschränkt
- Netzkennziffer (Bsp: 161, 171, 172, 177, ...)



- Subadresse
- Endgeräteauswahl im ISDN
- Teilnehmer in Nebenstellenanlagen

• **Internet Protocol: IP-Adressen**

- IP Version 4 Header: 32 Bit-Adressen
- (netid, hostid)
- (netid, subnetid, hostid): 134.60.77.7 (863C4D07)



• **Adressklassen**

- Klasse identifiziert Semantik und Format
- Punktnotation zur Lesbarkeit
- Klasse F reserviert

• **Probleme**

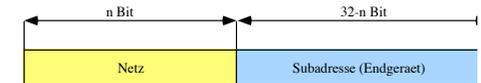
- zu wenig **verwendbare** Adressen
- Größe der Routingtabellen begrenzt
- schnelle Cache-Invalidierung durch große benutzte Adressmenge
- hierarchische Weglenkung steigert Effizienz

• **CIDR: Classless Inter-Domain Routing [RFC1519]**

- flexiblere Längeneinteilung der IPv4 Adressen -> mehr Netze
- Zusammenfassung von Netzbereichen in Routern (Hierarchie)

• **Schreibweise: A.B.C.D/n**

- Präfix: n Bits (von links)
- 139.20.0.0/16 : tu-freiberg.de
- 139.20.16.0/24: informatik.tu-freiberg.de



• **Flexible Netzwerkgrößen**

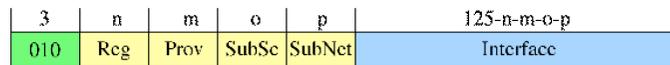
- kleine Netze möglich: z.B. /27
- oder mehrere Class-C Netzwerke (/24) zusammenfassen: /22 etc.

• **Aggregation**

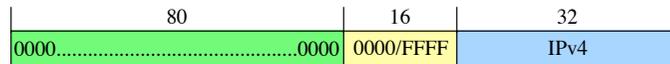
- Router fassen Netzwerke mit gleichem Präfix zusammen
- Tabellen kleiner im Speicher und in Paketen
- longest match algorithm

• **VLSM: Variable Length Subnet Mask**

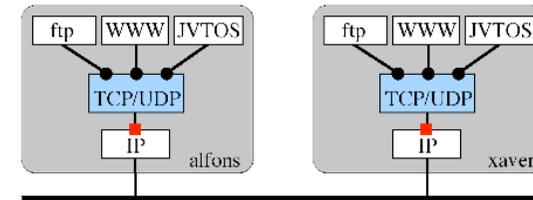
- IPv6: neuer Header
  - Quell- und Ziel-Adresse je 128 bit
  - weitere Felder (flow-label, etc) siehe Kapitel 3.5
  - 1500 bis  $3 \cdot 10^{17}$  Adressen / m<sup>2</sup>
- Präfix 3 - 10 Bit unterteilt in Klassen
  - Unicast, Multicast (1/256)
  - Local-Use
  - IPX, NSAP-Adressen für ISO CLNP
  - 85% future use
- Unicast-Adressen
  - 12,5% des Adressraumes
  - 4 Stufen



- auch als Anycast-Adressen verwendet
- gekapselte IPv4-Adressen



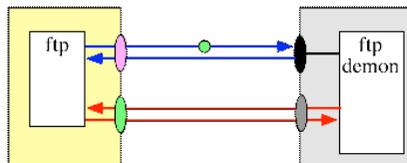
- Portkonzept
  - mehrere Dienste in einem Rechner
  - mehrere Verbindungen zwischen zwei Rechnern
  - Identifikation der Prozesse



- Well-known-ports für wichtige Internet-Dienste

21	ftp control port
23	Telnet
80	httpd
111	Sun Remote Procedure Call

- Mehrfacher Service auf einem Port
  - Kontaktaufnahme über 'Anrufport'



- Datenverbindung auf dynamisch zugewiesenem Port
- 'passive ftp'
- Wie finden andere Computer den richtigen Port?
  - Anfrage bei zentraler Instanz: port-mapper
  - port-mapper vermittelt Dienst und Anfrage
  - unterhält Verzeichnis (Dienst, Programm)
  - sucht freien Port
  - teilt Quelle und Ziel den Port mit
  - hat feste Portadresse

- Anschlußpunktadressen
  - ohne Bindung von der Hardware verwendet
  - (Dienstadresse, Kontextadresse, Geräteadresse)
  - (Telefonnummer), Pointer, I/O-Portadressen, ...

- IEEE 802.2
  - 16 bit oder 48 bit
  - 48 bit in LANs
  - Multicast: 01 00 5E + 24 bit
- z.B. Ethernet
  - 22 bit Organisationskennung (Hersteller)
  - 24 bit Seriennummer
  - im Ethernet-Chip oder ROM
  - Bsp: 08:00:07:1E:5F:8B
- dynamische Festlegung (z.B. LocalTalk)
  - beim Einschalten des Gerätes
  - zufällig Adresse auswählen
  - Paket an diese Adresse schicken
  - falls Bestätigung kommt: Adresse bereits vergeben
  - falls Timeout: Adresse verwenden

- Address Resolution Protocol ARP (RFC826)
  - welche Ethernet-Adresse gehört zu 134.60.77.7?
  - Senden im lokalen Netzwerk **nicht** über Gateway
  - Abbildung muß auch ohne Gateway funktionieren
- Cache mit Tupeln (IP,802.2)

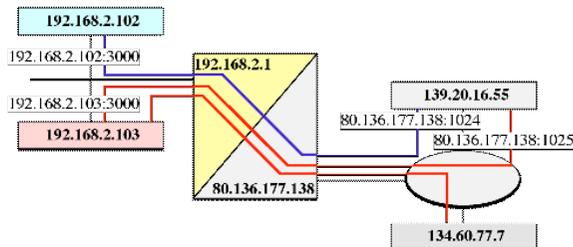
IP-Adresse	802.2-Adresse
134.60.77.99	08:00:08:14:11:2B
134.60.77.77	08:00:07:1E:5F:10
134.60.77.56	08:00:07:1E:5E:AB
134.60.77.10	08:00:07:1E:AF:FE

- Falls IP-Nr nicht im Cache
  - Ethernet Broadcast (FF:FF:FF:FF:FF:FF)
  - ARP-Request(gesuchte IP-Adr, eigene 802.2-Adr, eigene IP-Adr)
  - ARP-Reply(IP-Adr, 802.2-Adr)
  - Eintragen in lokale Tabelle
  - Timeout: Gateway-Adr eintragen

## • Effizienz von Adress-Systemen

- Technische Einschränkungen
  - Hierarchische Weglenkung
  - Nummernbereiche identifizieren Vermittlungseinheiten
  - 0731 502 XXXX
  - kann durch komplexere Adressbindung verbessert werden
  - 1996: 9,5 M Hosts,  $2^{32}$  Adressen  $\Rightarrow$  0,16%
- Administrative Einschränkungen
  - Zuordnung von Gruppen an Verwaltungseinheiten
  - Wachstum der Bedarfsgruppe ungewiß
  - 134.60.nnn.mmm blockiert  $2^{16}$  Adressen
  - Herstellerid im Ethernet
- H-Ratio [Huitema]:  $H = \frac{\log(\text{Anzahl Objekte})}{\text{Adressbits}}$ 
  - Maximum 0,30103
  - Sättigung zwischen 0,14 und 0,26
  - Telefonsysteme: Frankreich: 0,26, USA: 0,24
  - 200 Millionen Ethernetadapter: 0,18
  - IPv4 1996: 0,22 bei 9,5 Millionen Hosts

- NAT: Network Address Translation
  - auch IP-Masquerading
  - Gateway hat eine öffentliche IP-Nummer
  - viele nicht-öffentliche IP-Nummern
  - nur lokale Signifikanz
- Multiplex vieler n-IP-Nummern auf einer ö-IP-Nummer
  - Gateway vermittelt



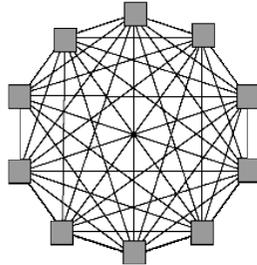
- Portnummern als Adresserweiterung
- evtl. Konflikte mit Portnummernvergabe
- wohlbekannt Ports im privaten Netz?

- Eigenschaften der Adreßbindung
  - Zeitpunkt
  - Algorithmus und Antwortzeit
  - Flexibilität und Fehlertoleranz
  - Wartbarkeit
  - Aktualität und Verfügbarkeit
  - Hochleistungs-Datenbank
- statische Adressbindung
  - fester Zeitpunkt
  - hoher Aufwand für die Aktualisierung der Datenbasis
  - Voraussetzung: hierarchisches System
  - Telefonnummern
  - ungeeignet bei mobilen Teilnehmern
  - DNS: Eintrag in Nameserver
- dynamische Adressbindung
  - beim Verbindungsaufbau
  - fortlaufende Adressbindung
  - suchende Adressbindung



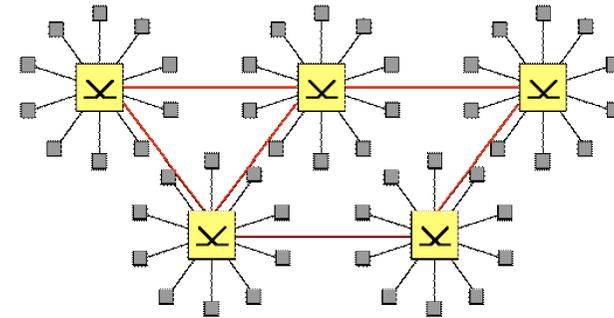
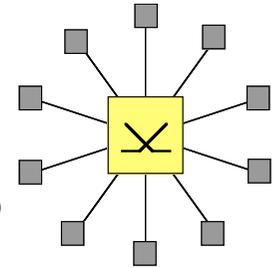
## 2. Vermittlungsdienste, insbesondere Paketvermittlung

- Welches Problem wird gelöst?
  - N Teilnehmer
  - paarweise temporäre Kommunikationsbeziehungen
  - 'Verbindungen' = Kontext der Kommunikation



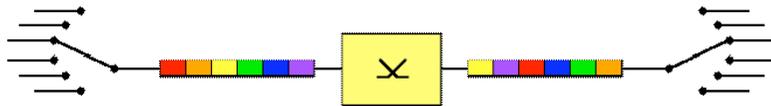
- voll verbundenes Netz:  $N*(N-1)$  Leitungen
- Historische Lösungen
  - Kurier, Post
  - Versammlung, Kino
  - Telefon, TV

- Lösungsidee: Sternstruktur
  - N Leitungen
  - 1 Vermittlungsknoten
  - evtl. Multiplex
- Skalierung: Vermaschtes Netz von Sternen
  - M Vermittlungen,  $M \ll N$
  - Leitungen zwischen Vermittlungen  $\leq M*(M-1)$
  - $N \ll N*(n-1)$  Anschlußleitungen

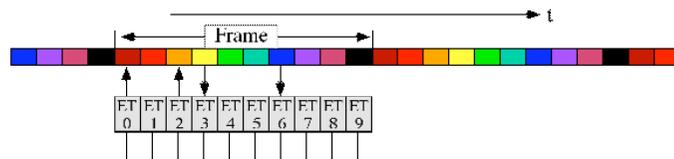


### 2.1 Bit- und Bytevermittlung

- Zeitlagewechsel

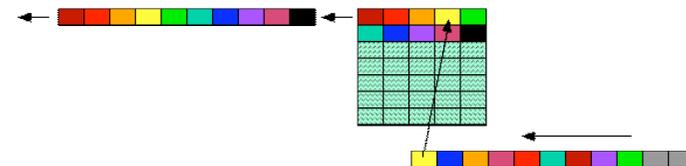


- Zeitkoppelfeld
  - Bus mit hoher Übertragungsrage
  - Zeitschlitztechnik



- Vermittlung einfach
  - $\text{Frame}[j] := \text{ET}[q]$ ;  $\text{ET}[p] := \text{Frame}[j]$ ;
  - Verbindungssteuerung konfiguriert ETs
- Multicast einfach: m Einheiten empfangen

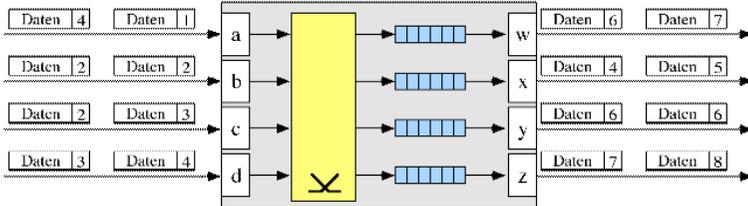
- Sonderfall Ethernet: Pakete statt Bytes
- Speichervermittlung: Memory Time Switch
  - Bytes kommen im Multiplexstrom an
  - Eingabe: Bytes werden einzeln adressiert und in RAM geschrieben
  - Ausgabe: Bytestrom wird sequentiell aus RAM ausgelesen
  - Ähnlich Video-RAM (VRAM)



- Verbindungssteuerung
  - Adresstabelle verwalten
  - Tupel (Zeitschlitznummer, Speicheradresse)
- Multicast = ein Byte in mehrere Speicherzellen schreiben

## 2.2 Zellvermittlung

- Adressbasierte Vermittlung
  - (Eingang, Kennzeichen<sub>n-1</sub>) -> (Ausgang, Kennzeichen<sub>n</sub>)
  - Dynamische Steuerung der Durchschaltung
  - Eingangspuffer und Ausgangspuffer

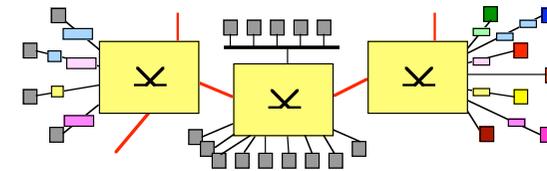
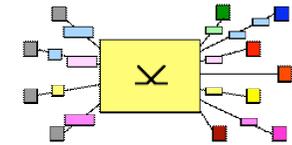


- Adressaustausch

Port in	Header	Port out	Header
a	1	x	5
a	4	w	6
b	2	y	6
b	4	y	7
c	3	z	8
c	2	x	4
d	4	w	7
d	3	z	7

## 2.3 Paketvermittlung

- Paket als Einheit der Vermittlung
  - variable Länge
  - Verbindungskennzeichen im Paket
- Vermittlung
  - m Eingangsleitungen
  - n Ausgangsleitungen
- Verbindungskennzeichen
  - bestimmt Ausgangsleitung
  - => Routing, Weglenkung
- 'Verbindungslose' Paketvermittlung
  - Adressabbildung bei jedem Paket



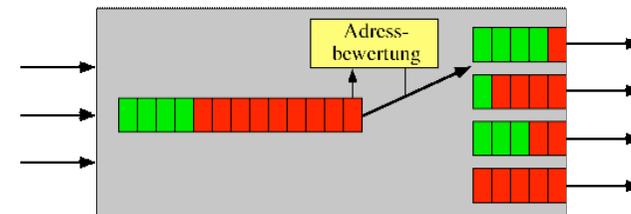
- Internet Protocol (IP)
  - Header mit vielen Feldern
  - Adressfelder siehe 1.7

Version	HeaderLen	Type of Service
Length		
Identification		
Fragment Offset		
Time-to-live	Protocol	
Header Checksum		
Source IP Address		
Destination IP Address		
Data ≤ 65.536 Byte		

- Protokollfunktionen
  - Fragmentierung (Tokenring <4500, Ethernet <1500, ...)
  - 1 PDU => n MTUs
  - Größe kann sich auf dem Weg ändern
  - Assemblieren (Reihenfolge!)
  - Kreisen der Pakete verhindern (TTL)

### 2.3.1 Routertechnik

- Store&Forward
  - Eingangsschlange sammelt Pakete
  - Ausgangsschlangen vor jeder ausgehenden Leitung
  - Adressauswertung ordnet Pakete Queues zu



- Header Processing + DMA
  - Input-Queue enthält Pakete
  - Output-Queues enthalten Zeiger auf Pakete
  - Adressbewertung erzeugt Output-Queue-Elemente
  - Direct-Memory-Access von den I/O-Interfaces

• Beispiel Internet Weglenkung

- von uhu.telematik.tu-freiberg.de nach www.adobe.com
- Programm traceroute

Find route to: www.adobe.com. (192.150.12.102), Max 30 hops, 40 byte packets

```

1 139.20.17.254 (139.20.17.254): 7 ms 9 ms
2 vivid.hrz.tu-freiberg.de. (139.20.64.254): 7 ms 7 ms 7 ms
3 cisco.hrz.tu-freiberg.de. (139.20.64.4): 2 ms 3 ms 2 ms
4 tu-dresden1.win-ip.dfn.de. (188.1.7.113): 7 ms 7 ms 15 ms
5 zr-leipzig1.win-ip.dfn.de. (188.1.7.37): 10 ms 11 ms 9 ms
6 zr-berlin1.win-ip.dfn.de. (188.1.144.6): 12 ms 12 ms 20 ms
7 zr-frankfurt1.win-ip.dfn.de. (188.1.144.106): 19 ms 18 ms 18 ms
8 ir-perryman1.win-ip.dfn.de. (188.1.144.78): 117 ms 126 ms 122 ms
9 bordercore3-hssi0-0-0.washington.cw.net. (166.48.41.249): 190 ms 143 ms 141 ms
10 core9.washington.cw.net. (204.70.9.61): 171 ms 163 ms 136 ms
11 * * *
12 112.atm3-0.xr1.tco1.alter.net. (146.188.160.90): 155 ms 138 ms 137 ms
13 293.atm2-0.tr1.dca1.alter.net. (146.188.161.170): 139 ms * 157 ms
14 101.atm6-0.tr1.scl1.alter.net. (146.188.136.222): 242 ms 233 ms
15 199.atm7-0.xr1.scl1.alter.net. (146.188.146.5): 232 ms 233 ms 201 ms
16 195.atm11-0-0.gw2.scl1.alter.net. (146.188.145.137): 226 ms 212 ms 200 ms
17 192.150.13.1 (192.150.13.1): 217 ms 203 ms *
18 www.adobe.com. (192.150.12.102): 225 ms * 211 ms

```

•Trace completed•

• Strategie des Routing

- Statische Verfahren mit vorher bestimmten Tabellen
- Dynamische, verteilte Verfahren
- DVRP: Distance Vector Routing Protocol
- Metrik: Hops, Verzögerung, Auslastung, ...
- Link State OSPF: Link State Open Shortest Path First

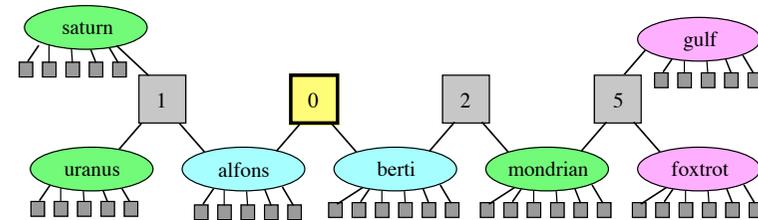
• Cut-Through-Routing

- Adressauswertung sehr schnell
- Paketelemente (Bytes) sofort in den Ausgang bewegen
- Problemfälle konventionell verarbeiten

• Multicastrouting

- Pakete im Router replizieren
- wohin schicken?
- überall ist einfach
- selektiv: Multicastbaum
- evtl. unbekannt

• Adressabbildung



- Tabelle (Netz, Kosten, Gateway)
- Tabelle beginnt mit lokaler Information

Netz	Kosten	Gateway
alfons	0	0
bert	0	0
saturn	1	1
uranus	1	1
mondrian	1	2
foxtrot	2	2
gulf	2	2

- Mitteilungen von anderen verbessern Tabelle (DVA)
- im Prinzip vollständiges Wissen nötig

MBONE: Multicast Backbone

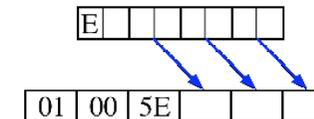
• IP-Multicast

- Steve Deering
- RFC 1112

• spezieller Bereich von IP-Nummern

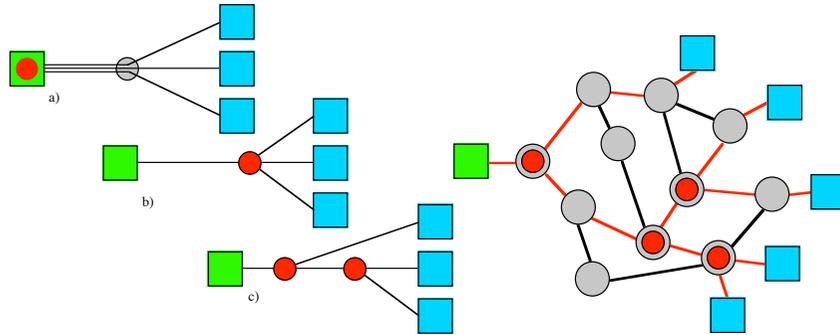
- Class-D Adressen: En.nn.nn.nn
- 224.0.0.0 to 239.255.255.255
- SendIP normal
- JoinHostGroup ( group-address, interface )
- LeaveHostGroup ( group-address, interface )
- Besondere Adressen zum Management

• z.B. Ethernet Multicast

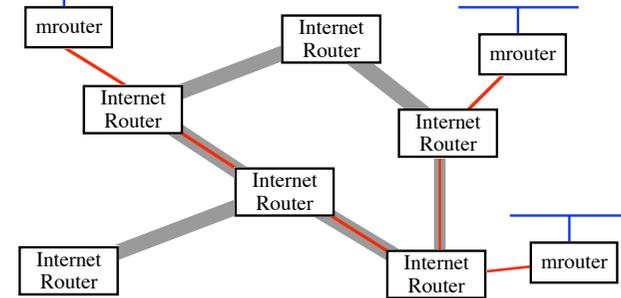


- Ethernet Interface entsprechend konfigurieren

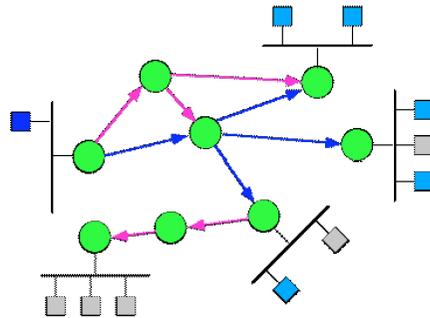
- Multicast-Router
  - Nachrichten-Replikation



- Mbone
  - mrouter in einer Workstations im LAN
  - Tunnel zum Transport durch das normale Internet
  - zum nächsten mrouter
  - Multicast-Pakete in normale IP-Pakete kapseln
  - bilden Overlay Netz (virtuelles Netz)

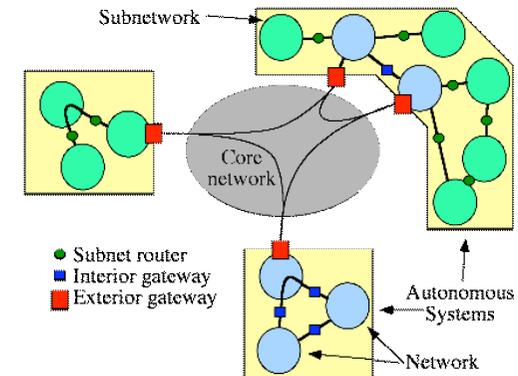


- DVMRP - Distance Vector Multicast Routing Protocol ('dense mode')
  - 'flood and prune'
  - 'grow back' nach einer gewissen Zeit

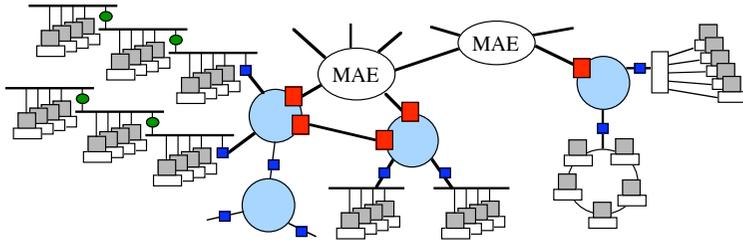


- Sparse Mode
  - Mehr Wissen über Multicast-Baum
- PIM Protocol Independent Multicast ('sparse mode')
  - Beitrittsnachricht an bekannte Rendezvous-Punkte
- MBONE tools (vat, nevt, ivs, nv, vic, sd, wb, rat, ...)

- Internet mit mehreren Netzwerken
  - verschiedene Routing-Algorithmen in Teilnetzen
  - Subnetwork (subnetid, z.B. 134.60.77.80)
  - Network (netid, z.B. 134.60.77.80)
  - Interior Gateways
  - Exterior Gateways



- Das Internet
  - heterogene Netzwerke
  - Router
  - homogenes Internet-Protocol (IP)



- Wer bezahlt?
  - 'Peering' Abkommen zwischen Netzwerken: Volumenverhältnis
  - Durchleitung zu Subnetzwerken: Pakete, Volumen
  - Abrechnung mit Endkunden: pauschal, Zeit, Volumen, Pakete
  - Peers gleichen Kosten aus
  - Abhängige: decken Kosten und erzeugen Ergebnis

## 2.3.2 Routingalgorithmen und Protokolle

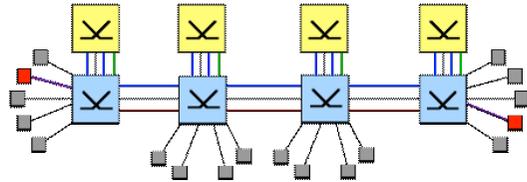
- Distance-Vector-Algorithm [Bellman-Ford]
  - Entfernung als Metrik
  - Delay, Anzahl Hops, ...
- Ablauf
  0. jeder Router kennt nur seine Netzwerke + Kosten dorthin
  1. jeder versendet Tabelle an die Nachbar-Router
  2. Tabellen-update: for (alle nets in den angekommenen Tabellen)
    - 2a.  $D_{net} := D(\text{zum Nachbar}) + D(\text{vom Nachbar})$
    - 2b. if RTable[net] leer: RTable[net].dist :=  $D_{net}$ ;
    - 2c. else RTable[net].dist :=  $\min(\text{RTable[net].dist}, D_{net})$ ;
  3. weiter mit 1.
  4. Tabelle hat vollständiges Wissen über Netzwerk
- Routing Information Protocol: RIP
  - Interior gateway protocol (IGP)
  - hello-messages alle 30 Sekunden
  - timer sorgen für Ausfallsicherheit
  - count-to-infinity-Problem
  - Tabellen evtl. inkonsistent

- Border Gateway Protocol zwischen autonomen Systemen
  - Border Gateway hat Routing-Tabelle für eigenes Netz
  - Border Gateway kennt all Router im Core-Network
  - Transit-System oder Stub-System
  - Tabellenaustausch per TCP-Verbindung
  - Routing-policy!
- RIP bis 1979 im Internet
  - Queue-Länge als Metrik
  - Durchsatz immer gleich (56 kbit/s)
  - Schlechte Nachrichten verbreiten sich langsam
  - Tabellennachrichten verkehrintensiv
- Link State Verfahren
  - ab 1979
  - 1. Nachbarn und deren Netzadressen entdecken (HELLO-Paket)
  - 2. Metrik-Parameter zu Nachbarn messen (ECHO-Paket)
  - 3. Link-State-Paket zusammenstellen
  - 3. Link-State-Paket an alle Router schicken (Flooding)
  - 4. kürzesten Pfad mit besonderem Algorithmus berechnen

- Dijkstra-Algorithmus
  - berechnet günstigsten Pfad
  - basiert auf Kosteninformation über Leitungen
  - Graphenalgorithmus
- Link State Open Shortest Path First (OSPF)
  - Standard im Internet als Interior Gateway Protocol (IGP)
  - berücksichtigt verschiedene Metriken
  - hierarchische Systeme
  - Lastausgleich
  - Link-State-Update nur an logische Nachbarn
- Internet Control Message Protocol ICMP
  - Wartungsprotokoll
  - Destination unreachable, time exceeded, ...
  - Echo Request und Reply

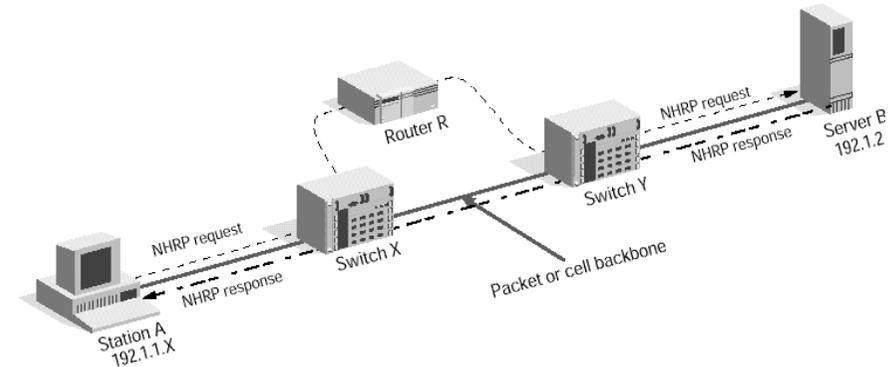
### 2.3.3 Layer-3 Switching

- Routing ist relativ teuer
  - IP-Adresse => physikalische Adresse
  - zeitkritische Datenströme: viele Pakete mit gleicher Adresse
- Ethernet-Vermittlungen
  - EN-Zieladresse als Vermittlungskriterium
  - Hardware zur Adresserkennung



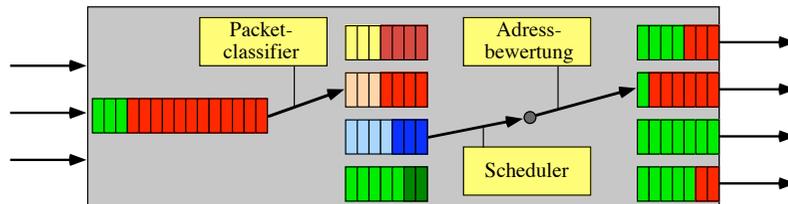
- Route first - then switch
  - Routing in Software (Tabellengröße!)
  - Router finden Ethernetadresse des Zieles heraus
  - Ethernet-Switches konfigurieren
  - im ersten EN-Switch Zieladresse von Gateway auf Ziel ändern

- Fast IP (3COM)
  - Zunächst ganz normal Pakete schicken
  - Verbindungsaufbau mit NHRP-Request (Next Hop Routing P)
  - Router setzt Verbindung auf falls Switched-IP existiert
  - Endgerät erhält Ziel-Ethernet-Adresse



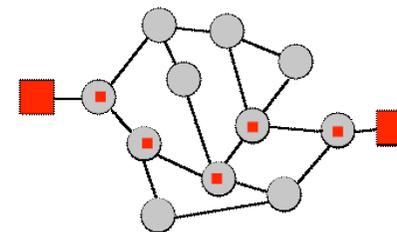
### 2.3.4 QoS im Internet

- Einfacher Ansatz: DiffServ (differentiated services)
  - mehrere Eingangsqueues
  - verschiedene Paketklassen
  - Klassifizierung mit Type\_of\_Service-Feld
- Priority-Queues
  - pro Klasse eine Queue



- Deja Vu
  - Prioritäten gab es schon oft
  - Welche Applikation benutzt niedrige Priorität freiwillig?
  - Hohe Priorität = hohe Kosten?

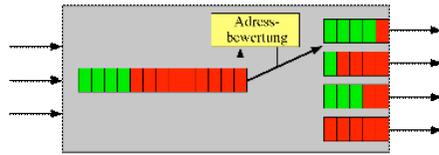
- Internet Integrated Services Architecture
  - IISA, IntServ
  - alle Einheiten im Pfad müssen QoS unterstützen
  - Reservierungsprotokoll
  - Zulassung zum Dienst
  - Überwachung
- QoS im Internet
  - erfordert Verbindungsparadigma
  - Reservierung von Ressourcen entlang des Pfades



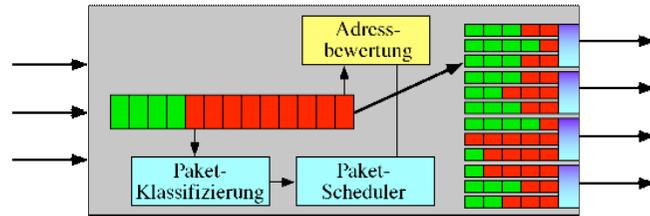
- eindeutiger Pfad?
- Zulassungsbeschränkungen

- Router: Store&Forward

- Eingangsschlange sammelt Pakete
- Ausgangsschlangen vor jeder ausgehenden Leitung
- Adressauswertung ordnet Pakete Queues zu



- Paket-Scheduling



- Paketklassifizierung?

- Definition: Flow (~ Verbindung)

- Identifikation von IP-Paketen
- IPv6: flowlabel
- IPv4?

Version	HeaderLen	Type of Service
Length		
Identification		
Fragment Offset		
Time-to-live		Protocol
Header Checksum		
Source IP Address		
Destination IP Address		
Data ≤ 65.536 Byte		

- Filterspec

- identifizieren Untermengen im Paketstrom
- Verwerfen von Paketklassen
- getrennt von der Reservierung

- Session + Sender\_Template

- Bezeichnung des Stromes durch den Sender
- z.B. IP-Nummern, Port
- referenziert beim Reservieren

- Sender\_TSpec

- Obergrenze des Verkehrs vom Sender

- AdSpec

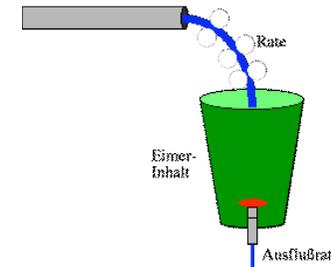
- QoS-Schätzungen vom Netzwerk
- Verzögerung, Bandbreite
- wird auf dem Weg durch das Netzwerk aufgesammelt

- FlowSpec, Flow-Specification

- rudimentäre QoS-Spezifikation
- controlled delay
- predictive QoS, guaranteed QoS
- TSpec: Traffic-Specification: Peak Rate, Packet Size, Token-Bucket rate und size
- RSpec: Receiver-Specification: Rate

- Token Bucket Modell

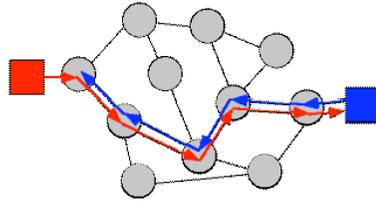
- Verkehr modellieren
- Verkehr 'formen' (traffic shaping)



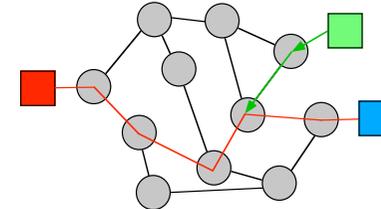
- Eimerinhalt entspricht Warteschlangenlänge
- Einfluß mit Bursts
- Bucket-Parameter beschreiben den Verkehrsfluß

## RSVP: ReSerVation Protocol

- Pfad-Nachrichten vom Sender
  - enthalten Flowspec
  - erzeugen 'Pfad' im Netz (path state Einträge in Routern)
  - enthalten Sender\_Template, Session, Sender\_TSpec
  - werden von Routern periodisch erneut gesendet (flooding)
- Reservation-Request
  - von den Blättern zur Wurzel
  - rückwärts entlang des Pfades
    1. Admission control
    2. Reservation: flowspec, filterspec
    3. Zusammenfassung von Res. Requests eines Multicast-Baums
  - eventuelles Scheitern upstream -> Freigabe

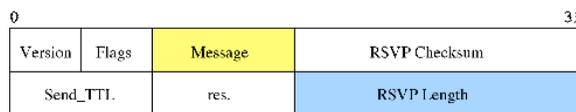


- IP-Multicast und Mbone
  - Multicast-Baum
  - flood-and-prune
  - Mbone: Multicast-Router und Tunnel
- 'Einklinken' weiterer Empfänger

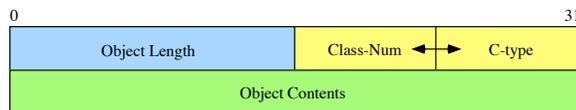


- Reservierung freigeben
  - upstream: ResvTear(session, style, filterspec)
  - downstream: PathTear(session, sender-template, ...)
  - von Sender oder Empfänger
  - von Routern nach path-state timeout

- RSVP-Pakete
  - Path, Resv, Resv\_Confirm
  - PathTear, ResvTear
  - PathErr, ResvErr

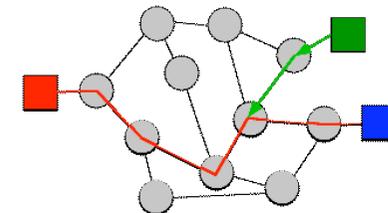


- RSVP-Objekte
  - session, sender\_template
  - style, flowspec, filterspec
  - sender\_tspec



- Objekt-Klasse und Objekte

- Soft-State
  - Gruppen groß und sehr dynamisch (TV-Modell)
  - Fehlersituationen in Routern managen?
  - Reservierungen verfallen nach kurzer Zeit (path state timeout)
  - Quellen senden periodisch 'Path'-Nachrichten
  - Empfänger senden periodisch 'Reservation'-Nachrichten
  - Paketrage-Kontrolle
- Charging fehlt



### 3. Protokolle

- Ende-zu-Ende Datenübertragung
  - Abstraktion benutzter Dienste
  - Standardisierung von Operationen

#### 3.1 Aufgaben und Mechanismen

- Ende-zu-Ende Verbindungen
- Partner-Prozess finden
  - Portkonzept (siehe 1.7)
  - Object Request Broker (CORBA, siehe Vorlesung Verteilte Systeme)
- Verbindungsaufbau
  - Ressourcen anlegen
  - Dienstegüte verhandeln
- Datentransport
  - Pipeline-Charakter
  - Aufteilen und Verpacken, Fehlerkontrolle
- Verbindungsabbau (Aufräumen)
  - Ressourcen im Netz freigeben
  - Kontexte in Endgeräten löschen

### 3.1.1 Fehler kontrollieren

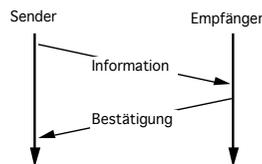
- Fehlerarten
  - Verlust der Meldung (Router, Puffer im Empfänger, ...)
  - Verfälschung der Meldung (Bitfehler, Zelle verloren)
  - Reihenfolge, Duplikate
- Fehler erkennen
  - Sequenznummer
  - Prüfinformation (CRC - Cyclic Redundancy Check)
- Nachrichten bestätigen
  - neue Protokollelemente
  - Bestätigungen (ACK)
  - Rückfragen (NACK)
  - separate Pakete oder piggyback
- Fehlerkontrolle und Medien
  - Audio und Video brauchen keine Wiederholungen
  - Text und Bild tolerant gegen Bitfehler
  - verschiedene Empfindlichkeit auch in den Daten
  - Fehlerkontrolle sollte kontrollierbar sein!

#### 3.1.1.1 a-posteriori Fehlerbehebung

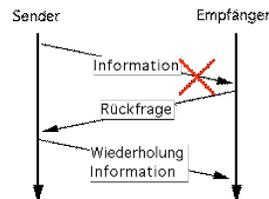
- Erneute Übertragung: Retransmission
  - fehlerhaftes/fehlendes Paket wird nochmal übertragen
  - go-back-n
  - selektive Wiedervertragung

- Aktive Fehlerkontrolle

Normale Übertragung



Fehlerhafte Übertragung

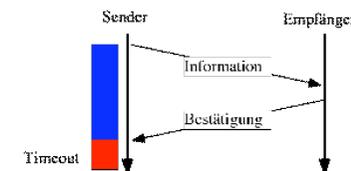


- Probleme

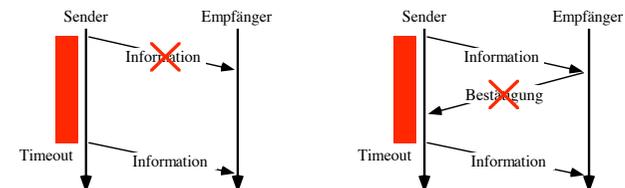
- was passiert wenn ACK/NACK beschädigt bzw. verloren?
- Endlosschleife bei fehlerhafter Rückfrage
- verlorene Datenpakete => timeout beim Empfänger?

- Passive Fehlerkontrolle

- Empfänger passiv
- Zeitüberwachung: Timeout im Sender



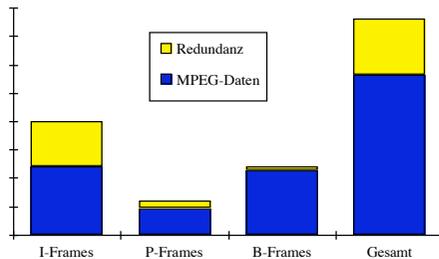
- Nachricht bei Ausbleiben der Bestätigung wiederholen



- Wiederholt auch verlorene Meldungen
- Round-trip-delay: Information-Bestätigung ...

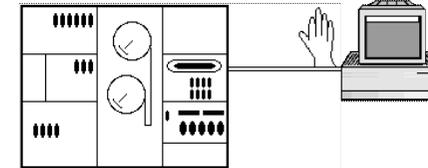
### 3.1.1.2 a-priori Fehlerbehebung

- Forward Error Correction
  - Nachrichten mehrmals senden
  - Nachrichten in verschiedenen Kodierungen senden
- Fehlerkorrekturinformation
  - Reed-Solomon-Codes, Faltungscodes (GSM)
- PET: Priority Encoded Transmission [Albanese, et.al]
  - selektive Redundanz für wichtige Komponenten



### 3.1.2 Kapazitäten managen

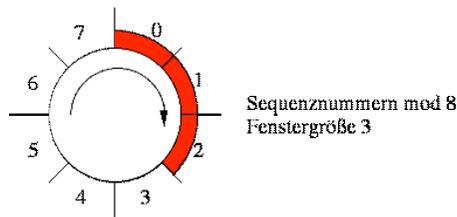
- Flußkontrolle
  - Anhalten des Datenflusses infolge Überlastung im Empfänger
  - keine Puffer frei
  - Anwenderprogramm verarbeitet zu langsam
  - Empfänger kann nicht schneller schreiben
  - Papier, Bildspeicher, Platte, zum nächsten Knoten ...



- Explizite Flußkontrollbefehle (RR, RNR, X-ON, X-OFF, ...).
- Implizite Flußkontrolle
  - Anzahl ausstehender Bestätigungen begrenzt
  - Empfänger hält Bestätigung zurück
  - Bestätigung nicht zu lange zurückhalten (time-out)

### • Fenstertechnik

- Übertragungsfenster = Anzahl unbestätigter Pakete
- Fenster ausgeschöpft => nicht senden

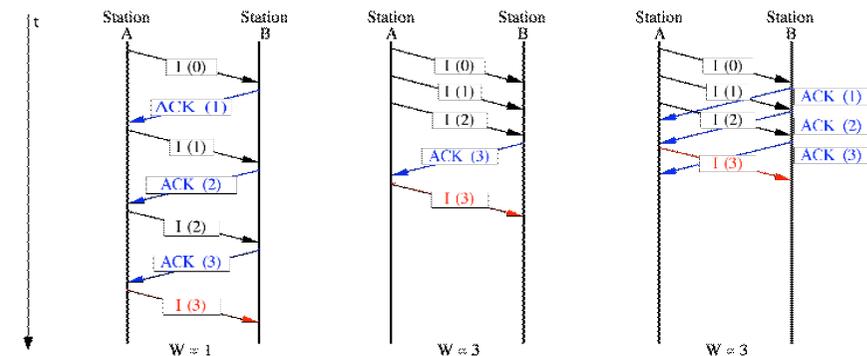


```

IF packetIn THEN
CASE inPacket.type OF
  Ack : window:=window+1;
  Nack: Retransmit(inPacket.expectedNumber);
  ...
END {CASE};
IF packet[bufile].ready AND (window>0) THEN BEGIN
  window:=window-1;
  SendPacket(packet[bufile]);
  bufile := (bufile+1) MOD cXmitBufferSize
END {then};
    
```

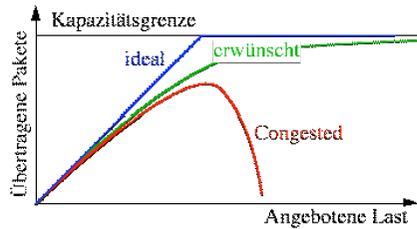
### • Beispiele für unterschiedliche Fenstergrößen

- ACK enthält Sequenznummer des erwarteten Paketes/Bytes



- weitersenden nach ACK bis Fenster erschöpft
- oft vermischt mit Fehlerkontrolle
- sofortiges Senden individueller ACKs verbessert Durchsatz
- $W > 1$  reduziert round-trip-delay

- Verstopfungskontrolle
  - typischer Durchsatzverlauf

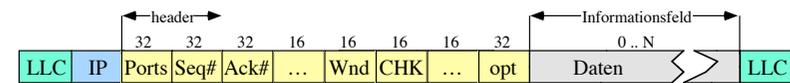


- Puffermangel im Router
- Leitungsempässe
- Unterschied zur Flusskontrolle
  - ein Phänomen im Netz
  - keine Verstopfung im Endknoten
  - Kommunikation Netz-Endgerät
- Netz verwirft SDUs
  - Reaktion des Endgerätes?
  - Beispiel IP und TCP

### 3.2 Beispiele

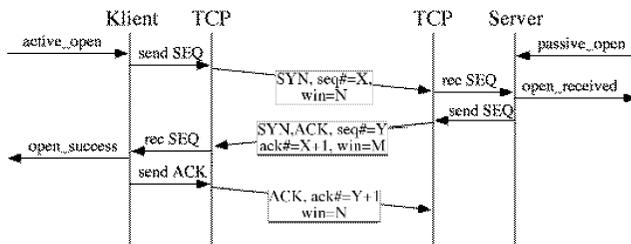
#### 3.2.1 Transmission Control Protocol TCP

- RFC 793 und viele weitere
- Stromorientierter Dienst (Sockets)
  - Bytestrom ohne weitere Struktur
  - voll duplex
  - gepuffert
- Sockets als Benutzungsmodell (service access point)
  - Server: `passive_open`, `open_received`
  - Klienten: `active_open` -> `open_success`
  - `send(daten)` -> `deliver`
  - `close` -> `closing` -> `terminate`
- Adresse = IP-Nummer + Port
- Paketformat



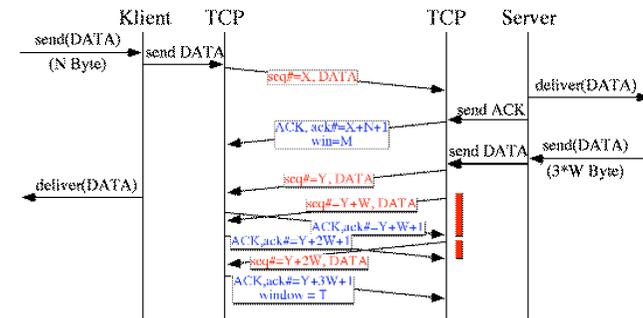
- ...: 6 Bit zur Signalisierung: URG, ACK, PuSH, ReSeT, SYN, FIN

- Verbindungsaufbau
  - Active Open: Verbindungsaufbauwunsch
  - Passive Open: ankommende Verbindung wird akzeptiert
  - three-way handshake
  - SEQ, ACK sind Bits im Headerfeld Codebits

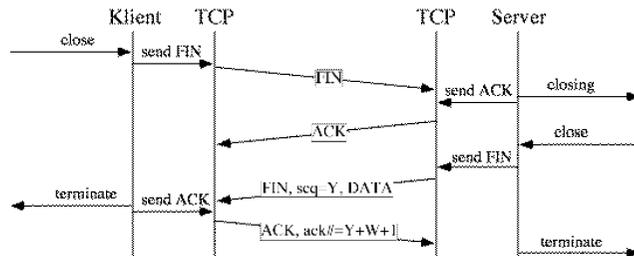


- Sequenznummern werden synchronisiert
- Auch andere Verbindungsaufnahme möglich
  - active\_open auf beiden Seiten
  - auch das erste Paket kann Daten enthalten

- Datenaustausch
  - sliding window, Byte-basiert
  - window advertisement im ACK
  - Sequenznummern 32 bit
  - adaptive Timeouts basierend auf RTT (round trip time)
  - window:= Min(advertisement, congestion\_window)

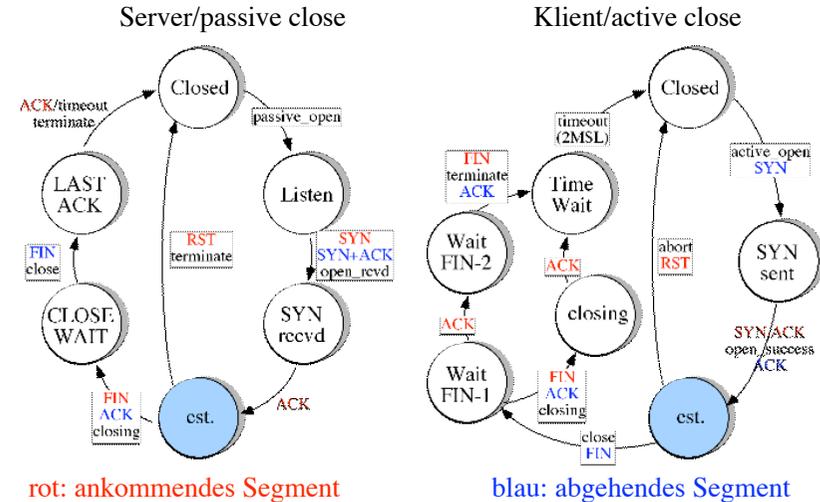


- Verbindungsabbau
  - Handshake zum Abbau
  - Daten können noch gesendet werden
  - warten auf FIN vom anderen Ende

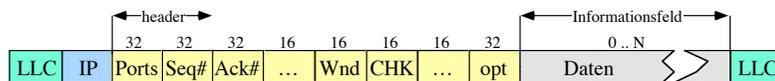


- FIN-Wait
  - Verbindungsende
  - A hat alles gesendet => schickt Paket mit FIN-flag
  - B antwortet mit ACK, hat aber noch Daten
  - B schickt Daten
  - B schickt FIN mit den letzten Daten

- Die TCP-State-Machine (siehe [W.R. Stevens: TCP ...])

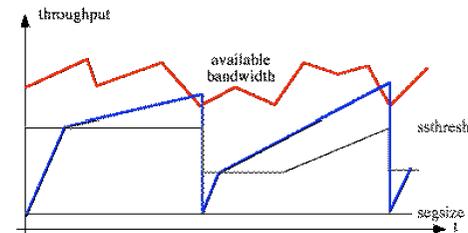


- eine kurze Kritik von TCP
  - einfache Protokollmaschine
  - Prüfsumme am Anfang
  - => erst Paket vollständig zusammenbauen dann Prüfsumme eintragen danach versenden



- Trick: const = trueCHK + fudge
- Fehlerkorrektur nicht abschaltbar
  - Flußkontrolle und Fehlerkontrolle vermischt
  - Go-back-n Fehlerkorrektur
  - Selective acknowledgement (SACK) vorgeschlagen
  - bei jedem Paketempfang ACK(höchste# 'in-order')

- Lastabwehr
  - Verlust eines IP-Datagrammes => Netzwerk überlastet
  - => Congestion-Window verkleinern
  - 'self-clocking': ACK=> inc(cwin)
  - slow\_start [Jacobsen, 1988]

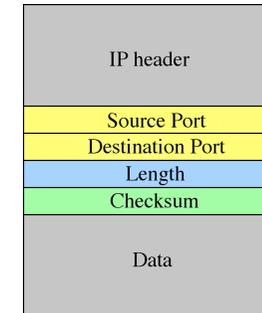


- Timeout: cwin = seg\_size; ssthresh = ssthresh/2;
- Slowstart: Paket erfolgreich übertragen: cwin = cwin+seg\_size;
- bis cwin>ssthresh => congestion avoidance (cwin=cwin+seg\_size\*seg\_size/cwin)
- am Übertragungsanfang zufällige Werte
- Sonderbehandlung für 2\*ACK und 3\*ACK

- Verbesserung des Fehler- und Verstopfungsverhaltens
  - doppelte Acks
  - Tahoe: fast Retransmit
  - Reno: slow start
  - schnell wieder an ordentliche Fenstergröße herantasten
  - exponentiell bis ssthresh, dann linear
  - Vegas
  - trotzdem: Paketverlust  $\neq$  Verstopfung
- Timed Wait Problem
  - Verbindungsende
  - A hat alles gesendet => schickt Paket mit FIN-flag
  - B antwortet mit ACK, hat aber noch Daten
  - B schickt Daten
  - A-Socket zu + neuer A-Socket offen => falsche Daten
  - => A muß Socket noch offenlassen
  - Mindestens 2 round-trips
  - Absturz an der anderen Seite?
  - => Server mit vielen, kurzen Transaktionen

### 3.2.2 User Datagram Protocol

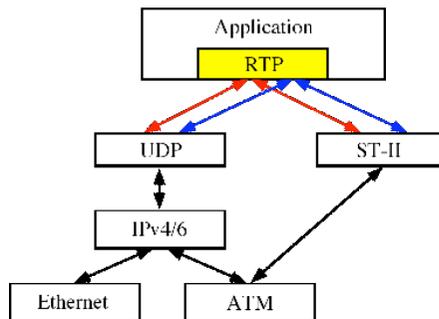
- Datagrammdienst für die Applikationen
  - IP für jeden
  - Ports kommen dazu
  - Prüfsumme (kann auch abgeschaltet werden)
  - Längenfeld
- Einfaches Paketformat



- Keine Fehlerbehandlung

### 3.2.3 RTP: Real-Time Transport Protocol

- Schulzrinne, Casner, Jacobson
- Multicast-Unterstützung

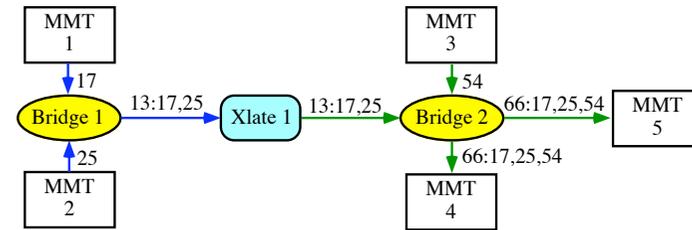


- RTP Data Transfer Protocol

- RTP Control Protocol: RTCP
  - Session-Kontrolle
  - Datenverteilung messen
- Problem (N)ACK-Impllosion
  - Multicast
  - n Empfänger -> n Receiver-Reports
- Lösung: Reports auf Multicastverbindung
  - Verwaltungsverkehr < 5%
  - Empfänger überwachen Verwaltungsverkehr
  - Bestätigungen auswerten
- Mechanismen
  - Receiver Reports: Paketverluste, Jitter, timestamp
  - Sender Report: Session-Größe, Uhrsynchronisation
  - Source Description
- Skalierbarkeit der Session (Multicast)
  - Multicast von RTCP-Paketen
  - Sendezeit 'zufällig' wählen
  - Gesamtdurchsatz limitieren

- Sender Report (SR)
  - Zeitstempel
  - Gesamtzahl Pakete und Bytes gesendet
- Receiver Report (RR)
  - Gesamtzahl Pakete empfangen
  - Gesamtzahl Pakete erwartet
  - Jitter
  - Letzter SR
- Source Description (SDES)
  - Source-ID
  - Canonical Endpoint Identifier: <User>@<DNS-name>
  - Name, E-Mail Adresse, Ort und Beschreibung (Prosa)
- Payload type mapping (FMT)
  - Source-ID
  - Typ 8 bit
  - Theoretischer Sample-Takt
  - IANA-ID (Internet Assigned Numbers Authority)
- Endpaket: BYE
  - Quelle beendet Sendung

- Bridge
  - Mixer



- Translator
- Synchronization Source, Content Source
- Einfaches Paketformat

0	9	16	31
Flags	d-type	Seq#	
Timestamp			
Sync Source			
1..n Content Sources			
Daten			

### 3.2.4 eXpress Transfer Protocol: XTP

(Literatur: T. Strayer, B. Dempsey, A. Weaver: XTP)

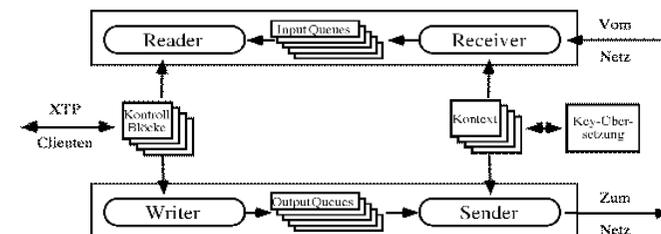
#### 3.2.4.1 Ideen

- Transfer Layer

Application	Application	Application
Presentation	Leer	
Session		
Transport	XTP	TCP und UDP
Network		IP
Data Link	L.L.C	L.L.C
Physical	MAC	MAC   PPP
	Medium	Medium

- Transport-Schicht + Netzwerk-Schicht
- Gründe: Leistungssteigerung
  - Integriertes Puffermanagement und Lastabwehr
  - Verminderte Duplizierung von Mechanismen (siehe X.25!)

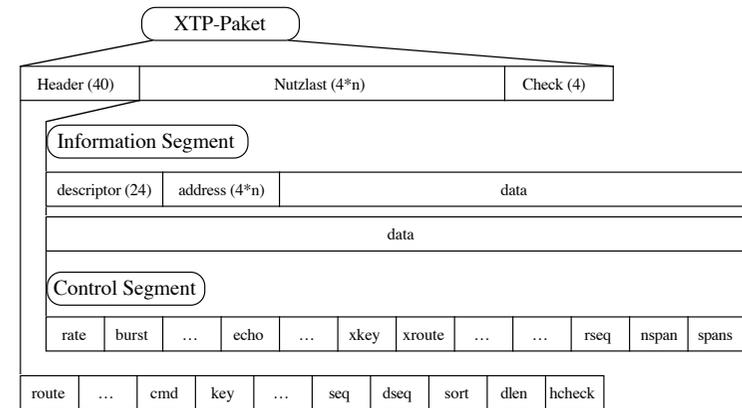
- Mechanisms vs. Policy
  - flexible Prozeduren
  - Verwendung bestimmt Protokoll
- Architektur



- Assoziationen
  - Sammlung von Statusinformationen für aktiven Datenaustausch
  - Im Sender und im Empfänger
  - In Zwischenknoten
  - Kontext-Record
- XTP - Vermittlungen (Router)
- Multicast
  - ungesichert (blast)
  - semi-zuverlässig
  - Synchronizing Handshake mit StatusRequest und -Report
  - Verschiedene Heuristiken zur Fehlerkontrolle
- Hardwareimplementierung als Entwurfsziel
- Prozeduren
  - Assoziations- und Pfadmanagement
  - Datenübertragung
  - Raten- und Flußkontrolle
  - Fehlerkontrolle

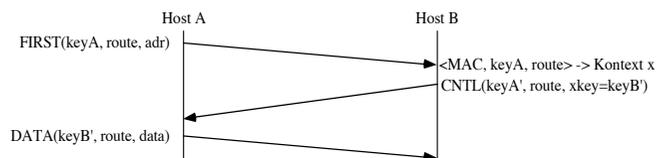
### 3.2.4.2 Paketformat

- Langwortaligniert, es gibt aber Bitfelder



### 3.2.4.3 Verbindungsaufbau

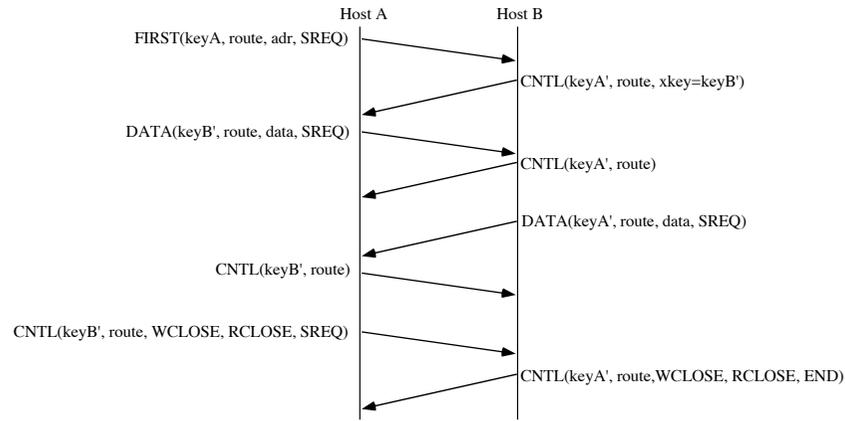
- FIRST-Paket
  - enthält Zieladresse
  - route und key Felder wichtig
  - rate-Feld
- In den Vermittlungsknoten
  - Pfad durch das Netz finden
  - Tabelle mit Tupeln (inroute, Eingangsport, Ausgangsport, outroute)
- Beim Empfänger
  - Muß Kontext im 'listen' Zustand haben
  - Abbildung des Tupels (key, route, MAC-Adresse) auf Kontext
  - Rückwärtspakete mit route+MSB und key+MSB (= returnkey)
  - key-Austausch zur einfacheren Abbildung auf Kontext



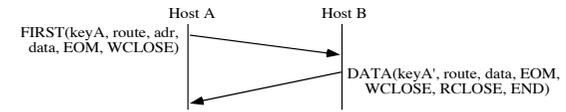
### 3.2.4.4 Datenübertragung

- In DATA und FIRST-Paketen
- Flußkontrolle mit Krediten
- Fehlerkontrolle
  - Bytesequenznummern geben Adresse im Empfangspuffer an ein langer Puffer genügt
  - Sender fordert Bestätigung an
    - SREQ- oder DREQ-Bit (DREQ nach 'delivery' beim Klienten)
    - Empfänger antwortet mit CNTL(rseq) oder CNTL(rseq,span(s))
    - rseq allein ermöglicht go-back-n
    - span(s) ermöglichen selective retransmission
  - Beispiel (Daten 1..34 bereits gesendet):
    - > DATA(seq=35, len =5, data=(35..39), SREQ)
    - <- CNTL(rseq=15, nspan=2, span1=(20,25), span2=(30,35))
    - > DATA(seq=15, len=5, data=(15..19))
    - > DATA(seq=25, len=5, data=(25..29))
    - > DATA(seq=35, len=5, data=(35..39))

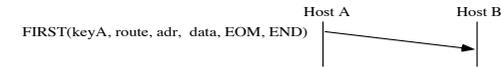
- Verschiedene Verbindungs-Paradigmen auf XTP abbilden
  - verbindungsorientiert



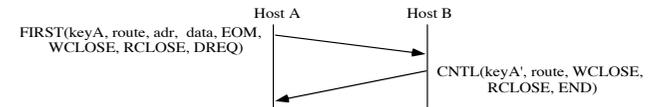
- Transaktion



- Datagramm

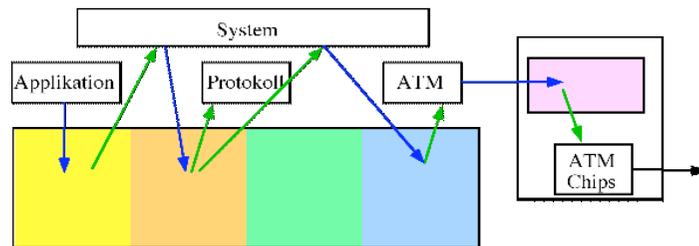


- zuverlässiges Datagramm



### 3.3 The demise of protocols

- Schichten bremsen den Datenfluß



- Ausführungsfluß im Schichtenmodell

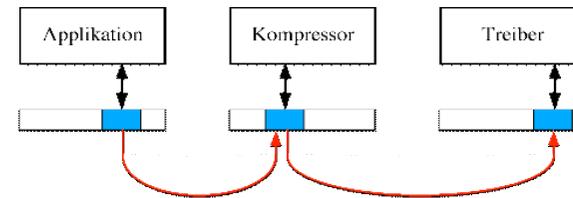
- Applikation
- Systemdienst: Socket-Layer
- Protokoll
- Systemdienst: BlockCopy/Gather
- ATM/AAL (Segmentation)
- System: Scheduler

- Warum?

- Kontextwechsel (Taskswitch) kostet Zeit, Cache ungültig
- Adressräume müssen umgeschaltet werden
- => Datenkopieren oder shared memory

- Datenkopieren vermeiden

- CPU-Last, Bus-Last
- Cache-Effizienz



- Beispiel Standard C - Ein/Ausgabe

- getc, putc jeweils mit einem Zeichen

- Integrated Layer Processing
  - Beschleunigung von Protokoll-Stacks
  - kein Kontextwechsel
  - eventuell Threads zur Abstraktion
  - nur ein Adressrum
  - keine Duplizierung von Funktionen

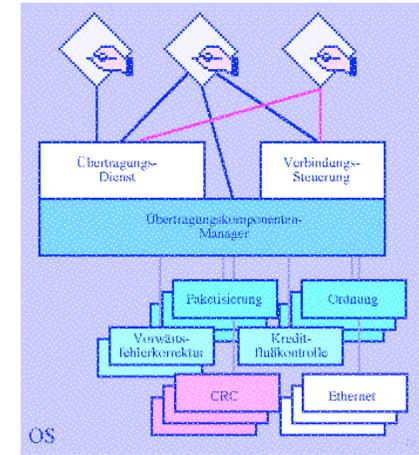
- Dynamische Dienstgüte
  - Quellkodierer kennt Datenstruktur
  - Beispiel Fehlerkorrektur:



- Zusammenarbeit Quellkodierer, Protokoll und Kanalkodierer
- Application Layer Framing
  - Abbildung (Anforderungen -> QoS -> Netzwerk) komplex
  - Dienstgüteverhandlung
  - flexible Fehlerkontrolle

- Optical Switching und WDM
  - End-to-End Verbindungen
  - superschnell
  - skalierbar mit schmalen Bändern (präzisen Lasern)
  - hochgradig zuverlässig
  - => viele Protokoll-Funktionen werden nicht mehr gebraucht

- Baukastenprinzip
  - mehrere Abstraktionsebenen
  - Mechanismenklassen
- Klassen
  - Protokollfunktionen
  - Fehlerbehandlung
  - Flußkontrolle
  - Lastkontrolle
  - Traffic-Shaping
  - => konfigurierbare Dienste



## 4 Steuerungsdienste

### 4.1 ISDN-Signalisierung: Q.931

- Paketformat
  - Protocol Discriminator  
Q.931 (5E, NT): 8;  
ITR6: 64, 65; ...
  - Connection Reference
  - Nachricht  
Setup, Setup-Ack, Call Sent, Alert, Connect, Disconnect, Disconn-Ack
  - Informations-Elemente  
Länge 1: Codesatzumschaltung  
1 < Länge < 256:  
Called Number, Calling Number, ...  
Cause, Bearer Capability, Service Indicator ...

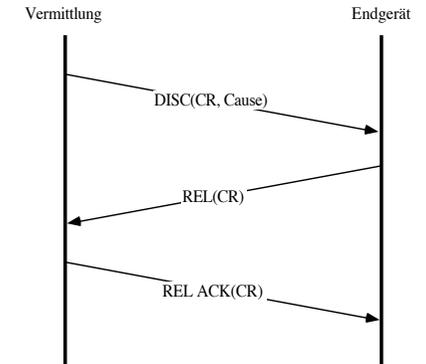
	Protocol
	Länge
C; R;	Call Ref.
	Message
	Information
	Element
I;	Informationstyp
0;	Informationstyp
0;	Länge
0;	Inhalt
1;	Inhalt

- Beispielpaket
  - Verbindungsaufbau
  - Telefonnummer
  - evtl. weitere Beschreibung: bearer capabilities

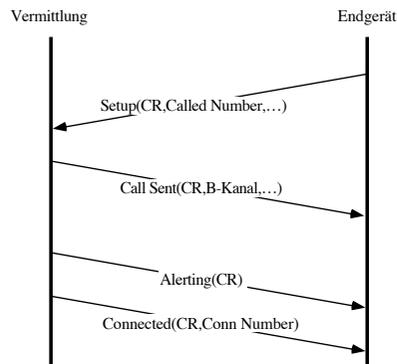
Inhalt	Paket	Feld
leitungsvermittelt	0 0 0 0 1 0 0 0	Protokoll-Diskriminator
1	0 0 0 0 0 0 0 1	} Referenz-Nummer
19	0 0 0 1 0 0 1 1	
CALL SETUP	0 0 0 0 0 1 0 1	Nachrichten-Typ
Zieleadresse	0 ... 1 1 1 0 0 0 0	Element-Typ (Adresstyp)
12	0 0 0 0 1 1 1 0	Länge des Adressfeldes (Rufnummer)
nationale ISDN-Nummer	1 ... 0 1 0 ... 0 0 0 1	Typ der Rufnummer, Nummernplan
0	0 0 1 1 0 0 0 0	} Rufnummer, IA5-Zeichen (ASCII)
8	0 0 1 1 1 0 0 0	
1	0 0 1 1 0 0 0 1	

- Prozeduren
  - Command / Response für Ebene 3
  - implizite Festlegung der Anwendungs-Schicht
- Basic Calling
  - Verbindungsaufbau, -abbau
  - Status, Information, Progress
  - Pakete: Setup, Disc, Alert, ...
  - nicht besonders komplex
- Verbindungsbezogene Leistungsmerkmale
  - Facility Paket
  - huckepack in anderen Paketen
  - entsprechende Connection reference
- Verbindungslose LM (anschlussbezogen)
  - Facility Register, -Status, -Indication,
  - besondere Connection Reference
- Informationselemente für LM
  - Functional: Facility Information Element
  - Stimulus: Feature Activator/Indicator.

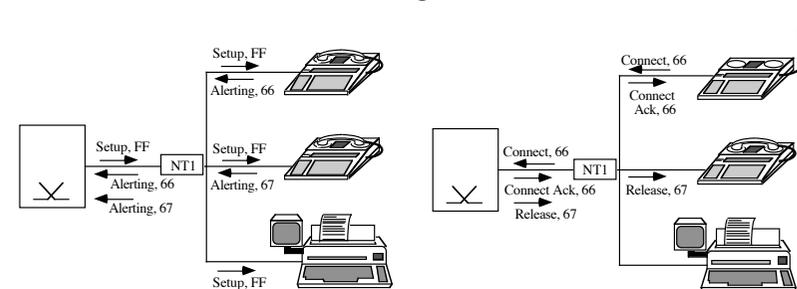
- Funktionale Signalisierung
  - Remote Procedure Call
  - besondere Pakete oder huckepack
  - Prozedur (Facility(X)):
    - Leistungsmerkmal X ausführen
    - ← Leistungsmerkmal X (nicht)ausgeführt
- 1TR6; DKZ-N1; NT BCS 29; 5E6 (; VN2 )
- Einfaches Beispiel:
  - Verbindungsabbau durch Netz



- Mittleres Beispiel:
  - Verbindungsaufbau durch Endgerät

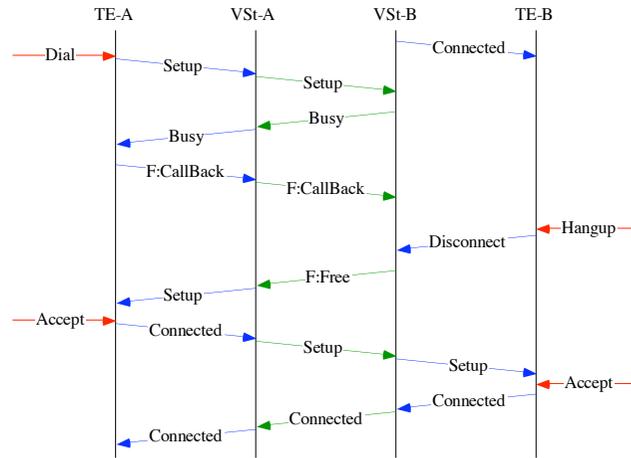


- Komplexes Beispiel: Verbindungsaufbau vom Netz
  - mehrere Geräte am Bus
  - Anbieten der 'kommenden Belegung'
  - Prüfen des Dienstes in den Endgeräten

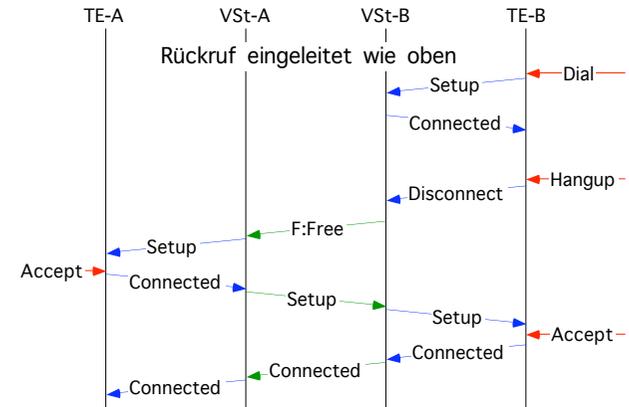


- Annahme des Rufes
- Rücknahme des Angebotes

• Feature Rückruf bei besetzt



• Feature: Rückruf bei frei



• Stimulus Signalisierung

- Terminalbetrieb
- Digital Centrex
- AT&T, NT

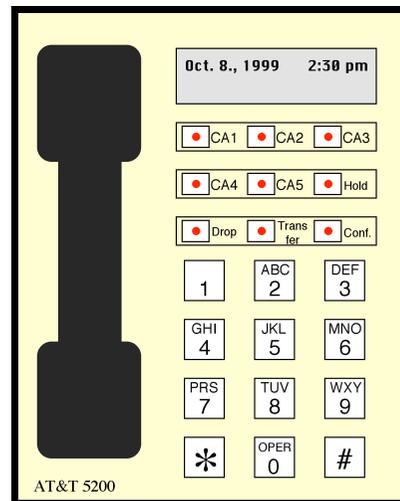
• Prozedur für LM

- INFO-Pakete oder huckepack.
  - > Knopf gedrückt (Info FA=Button 12)
  - <- Lampe einschalten (Info FI=LED 12)

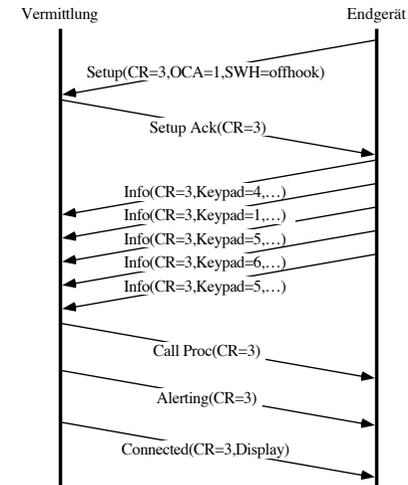
• Terminal Management

- Call Appearance
  - BNx := CAy aktivieren
  - LEDx := CAy aktiv
- Feature Activator und Feature Indicator
  - FAz := Transfer
  - Flz := Transfer aktiv

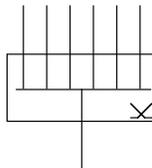
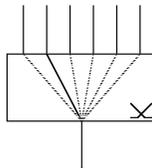
- Bsp: Telefon mit 9 Knöpfen
  - 5 Verbindungen (CAs)
  - 4 Features



• z.B. Verbindungsaufbau:



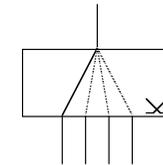
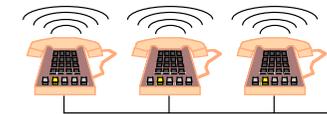
- Verbindungsbezogene LM
- Verbindungsaufbauphase
  - Rückruf
  - Ring Again
  - Anruferidentifizierung (ICLID)
- Asynchronous Multipoint ("Dreierverbindung")
  - Flexible Call Offering:
    - Anklopfen (call waiting)
    - bevorrechtigte Anrufe (Chefruf)
  - Halten und Wiederaufnehmen
    - Rückfrage
    - Makeln
  - Transfer
- Synchronous Multipoint
  - Konferenz
  - LAN



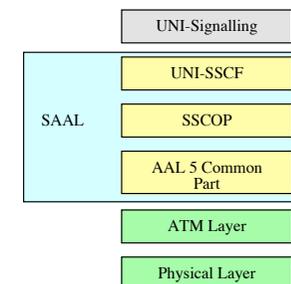
## 4.2 ATM-Signalisierung Q.2931 und ATM-Forum UNI

- PVC - Permanent Virtual Circuits
  - konfiguriert im Cross-Connect
  - mit Attributen cell-rate etc.
  - Änderung i.A. manuell
- SVC - Switched Virtual Circuits
  - geschaltet auf Anforderung
  - automatisch
  - Aufbau  $\ll$  1 sec
- Attribute des Verbindungsaufbaus
  - Adressen (called party, calling party)
  - Verkehrscharakteristik, QoS
  - VPI/VCI als Antwort
- Standardisierung
  - Network Network Interface (NNI), PNNI
  - User Network Interface (UNI)
  - ITU Q.2931

- Key-System
  - Call Pickup (Rufübernahme)
  - Call Park
  - Bridging (Aufschalten)
  - Rufübergabe (Transfer)
  - Sammelschluß (hunt group)
- Verbindungslose LM
  - Ruhe
  - Make Set Busy
  - Rufumleitung (immer, fallweise, ...)
  - Information (Gebührenanzeige, Zeit)
  - Nachrichten
- Abfrageplatzfunktionen ...

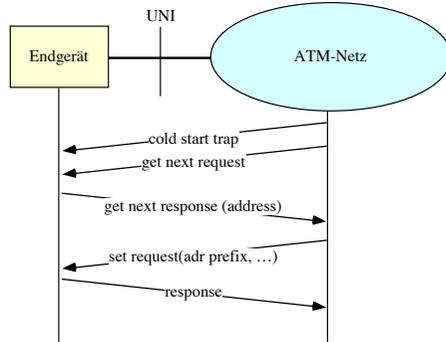


- Fundament Q.931
- Umgebung im ATM
  - SAAL: Signalling ATM Adaptation Layer (AAL 5)
  - Service Specific Coordination Function (SSCF)
  - Service Specific Connection-Oriented Protocol (SSCOP)
  - SSCOP: zuverlässige Übertragung (LAP-D Ersatz, entfernt ähnlich)

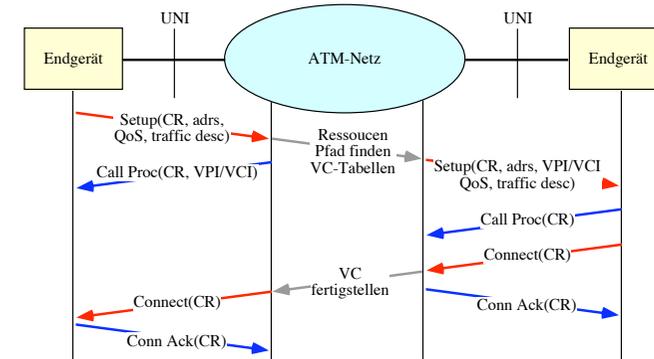


- Signalisierung mit VPI = 0 und VCI = 5

- ATM-Adressierung
  - NSAP-Verpackung
  - Domain und Authority Specification, Domain Specific Part
  - z.B. E.164 als Adresse
- Adressregistrierung basiert auf SNMP
  - UNI Management Information Base (MIB) im Endgerät
  - ILMI: Interim Local Management Interface

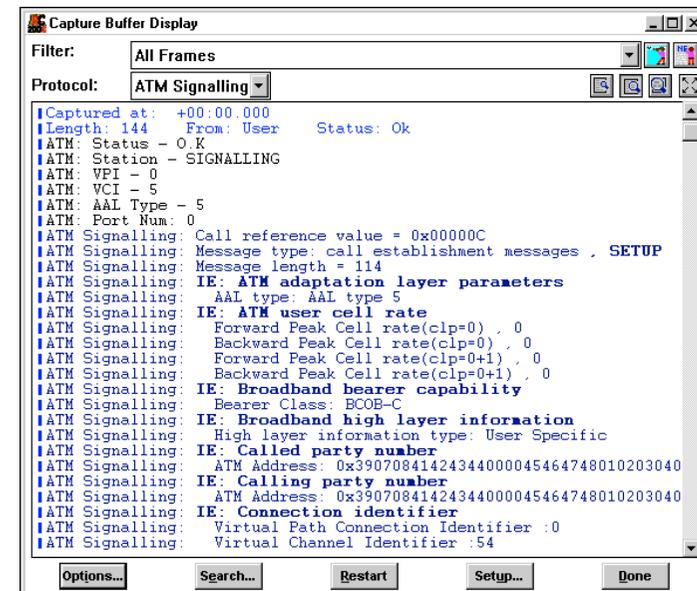
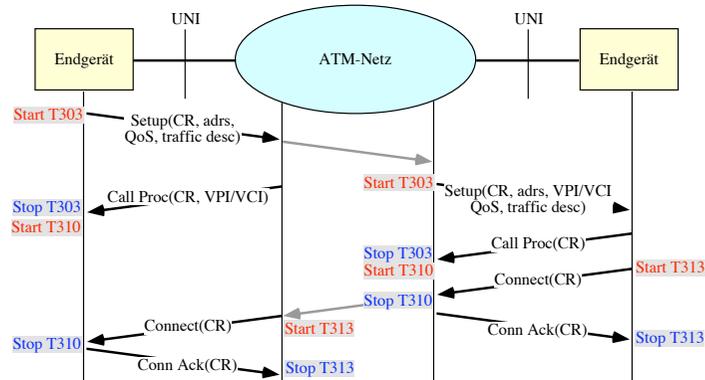


- Setup-Prozedur ähnlich ISDN
  - VPI/VCI statt B-Kanal-Identifikation
  - QoS und traffic descriptor statt Bearer Capabilities

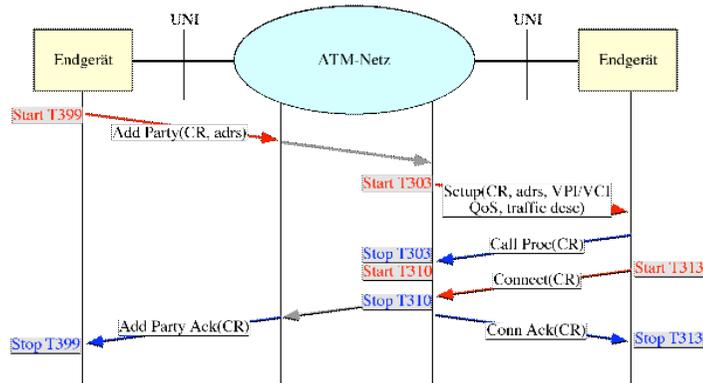


- Verbindungsabbau sehr ähnlich Q.931

- Überwachung durch Timer



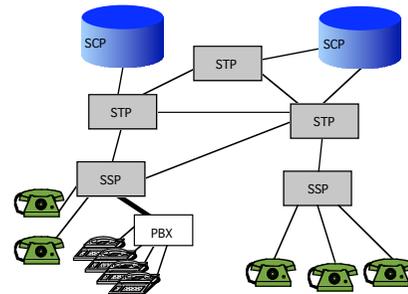
- Mehrpunkttopologien
  - basiert auf Punkt-zu-Punkt Verbindung
  - add party, auch für mehrere Teilnehmer
  - drop party



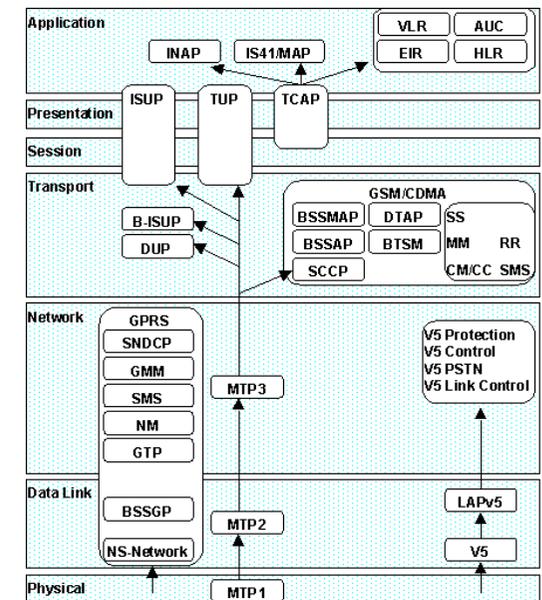
- Modifizierte Informations Elemente (Broadband ...)
  - Bearer capability (delay, CBR, VBR, ...)
  - higher und lower layer compatibility
  - repeat indicator
- AAL-Parameter (neu)
  - Typ: 1, 3/4, 5
  - circuit emulation, high quality audio, video, ...
  - maximale SDU-Größe (AAL 3/4)
  - CBR: 64 kbit/s, DS1, E1, n\*64, ...)
  - AAL 5 mode: message oder streaming
  - clock und error recovery für AAL 1
- ATM user traffic descriptor
  - peak cell rate (forward und backward)
  - sustainable cell rate (forward und backward)
  - maximum burst size (forward und backward)
- QoS-Parameter: CBR, VBR, ABR, unrestricted
- Connection Identifier: VPI/VCI

### 4.3 Signallingssystem 7 (SS#7)

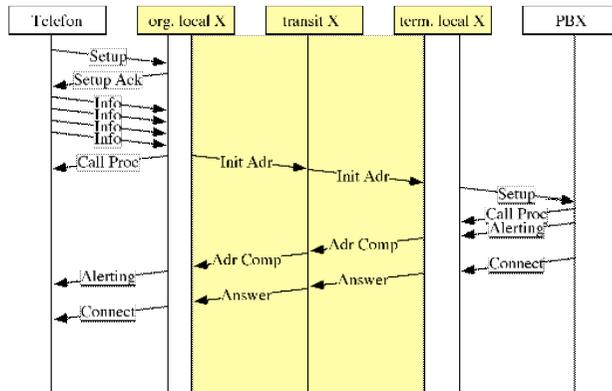
- Common Signalling Channel
  - getrennt vom Medienstrom-Netz
  - 56 oder 64 kbit/s Links
- Verbindungskontrolle im Netzwerk
  - Vermittlungen und Netzwerke
  - GSM-ISDN, GSM-GSM, ...
- Service Switching Point (SSP)
  - Vermittlung
  - Netz-Übergabepunkte
  - verstecken Endgeräte
- Service Control Point (SCP)
  - Rufbewertung
  - Wegfindung
  - Management des Verbindungsaufbaus
  - Prozeduren und Datenbank (VLR, HLR)
- Signalling Transfer Point
  - Router zwischen SCPs und SSPs



- Message Transfer Part
  - MTP2/Q.703
  - entfernt HDLC ähnlich
  - MTP3/Q.704, Routing
- SCCP
  - Signalling Conn. Control
  - => OSI Schicht 3
- User Parts
  - Verbindungssteuerung
  - Pfade finden
  - Leitungen zuordnen
- ISDN User Part
  - MAP
  - IS-41 and GSM
  - MSC-VLR fragt HLR
  - Authentisierung
  - Geräte-Identifikation
  - Roaming



• ISDN User Part



- Call Routing aus Datenbankabfrage
- local exchanges und transit exchanges sind SSPs
- IAM: Nummer bewerten, nächste SSP suchen, reservieren
- ACM: Kanal durchschalten
- ANswer Message

• Initial Address Message

- called und calling party number
- forward call indicator
- nature of connection, user service

• Address Complete Message

- charge indicator
- called party status and category indicators (besetzt ...)
- echo control indicator
- interworking, holding, ...

• ANswer Message

- backward call indicator
- access und network transport
- call reference
- notification indicator

• Siehe: <http://www.pt.com/tutorials/ss7/>

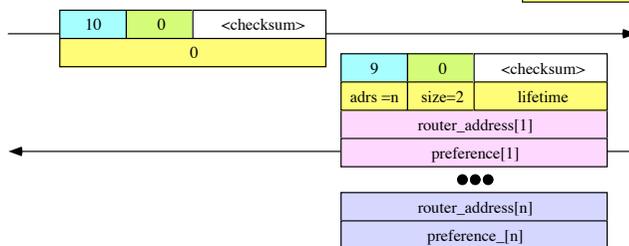
• Heute auch über IP: IP-STP

4.4 ICMP: Internet Control Message Protocol

• Nachrichten der über IP-Datagramme und IP-Router

- echo request/reply für ping
- destination unreachable (code: network, host, protocol, port, ...)
- Verstopfungskontrolle (source quench) veraltet
- Umleitung (redirect),
- 'time exceeded' (TTL oder reassembly-timeout)
- Zeitstempel
- transportiert in der Nutzlast von IP-Paketen

• Router-Bekanntgabe und -Suche



4.5 Netzwerkmanagement SNMP

• Infrastruktur für Netzwerkmanagement

• Management von Netzwerk-Elementen

- Hosts, X-Terminals, Terminal Server
- Drucker
- Router
- alle Netzwerk-verbundenen Geräte (Variablen-Definition ...)

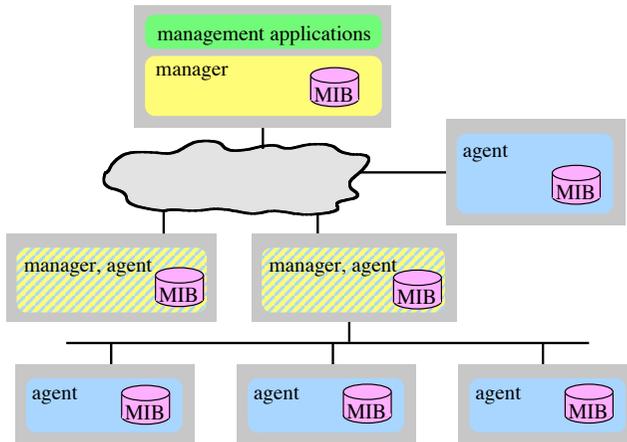
• Funktionen

- Statusabfrage => Object-Variable lesen (get request)
- Funktionen konfigurieren: Object-Variable setzen (set request)
- Alarme (traps)

• Komponenten

- Management Information Base (MIB): RFC 1213
- Structure of Management Information (SMI): RFC 1155
- Simple Network Management Protocol (SNMP): RFC 1157

- Management-Station
  - Protokoll, MIB, Grafisches UI

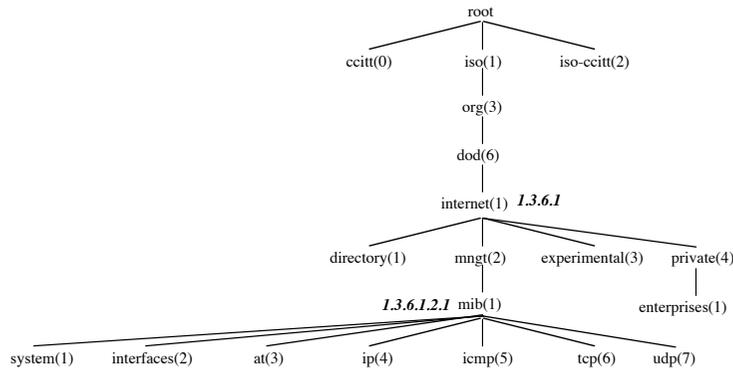


- Agent
  - Protokoll, MIB

- Structure of Management Information (SMI)
  - Datentypen
  - skalare Typen
  - Tabellen: 2-dimensionale Arrays

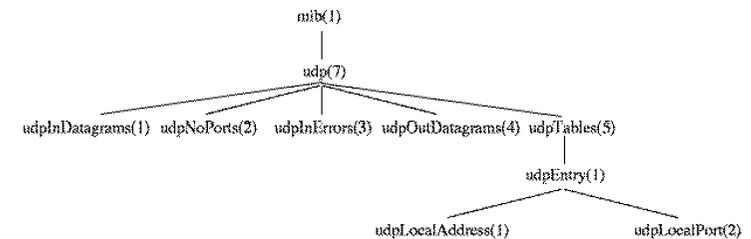
INTEGER	- $2^{31} .. 2^{31}-1$
UInteger32	0 .. $2^{32}-1$
Counter32, Counter64	nur aufwärts zählen, Wrap around
Gauge32	zu- und abnehmen, bleibt bei $2^{32}-1$ stehen
TimeTicks	zählt 10 msec
OCTET STRING	
IpAddress	4 Byte
PhysAddress	z.B. 6 Byte Ethernet Adresse
DisplayString	String, NVT-ASCII
OBJECT IDENTIFIER	
Opaque	Bitfeld
BIT STRING	Bits mit Namen
SEQUENCE	zur Definition von Datenstrukturen

- Namensgebung: Object Identifier
  - Punkt-Notation
  - Baumstruktur



- ASN.1 Makro zur Definition von Objekten (Basic Encoding Rules)
  - SEQUENCE, SEQUENCE OF, OBJECT TYPE
  - Vergabe von Namens-elementen
  - not accessible, read, read/write

- Vordefinierte Objekte (RFC 1213, MIB-II)
  - system: Gerätedaten (displaystring, sysUpTime, sysName...)
  - interface, at (address translation (ARP-cache))
  - ip, icmp, tcp, udp
  - routing tables
  - traps (siehe unten)
- Beispiel udp



- Anzahl Pakete für Ports ohne Listener
- Anzahl sonstiger Fehler (checksum, ...)

- Bezeichnung
  - 1.3.6.1.2.1.7.4.0
  - iso.org.dod.internet.mngt.mib.udp.udpOutDiagrams.0
  - abgekürzt: udpOutDiagrams.0
  - normale Variablen: 0 anhängen
- Tabellen-Objekt-Bezeichnung
  - Tabellen immer aufsteigend sortiert
  - Zugriff Spalte-Zeile
  - 1.3.6.1.2.1.7.5.1.1.0.0.0.0.161 - udpLocalPort.0.0.0.0.161

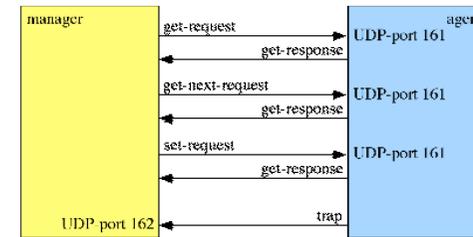
udpLocalAddress	udpLocalPort
0.0.0.0	67
0.0.0.0	161
0.0.0.0	520

- Bsp: snmp

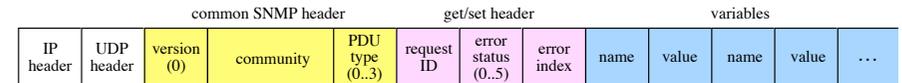
```

c:\iptools> snmp -a eincomputer -c geheim
snmp> get udpInDatagrams.0 udpNoPorts.0
udpInDatagrams.0=2345677
udpNoPorts.0=12
snmp> next
udpInErrors.0=33
snmp> quit
c:\iptools>
  
```

- SNMP Protokollelemente



- SNMP PDU



- trap



trap type: coldStart, warmStart, linkDown, linkUp,...