3. Kommunikationsunterstützung für Multimedia

3.1 Herkömmliche Netztechnik in LANs und WANs

3.2 Dienstgütemerkmale (Quality of Service)

3.3 Multicast

3.4 Medienskalierung und Medienfilter

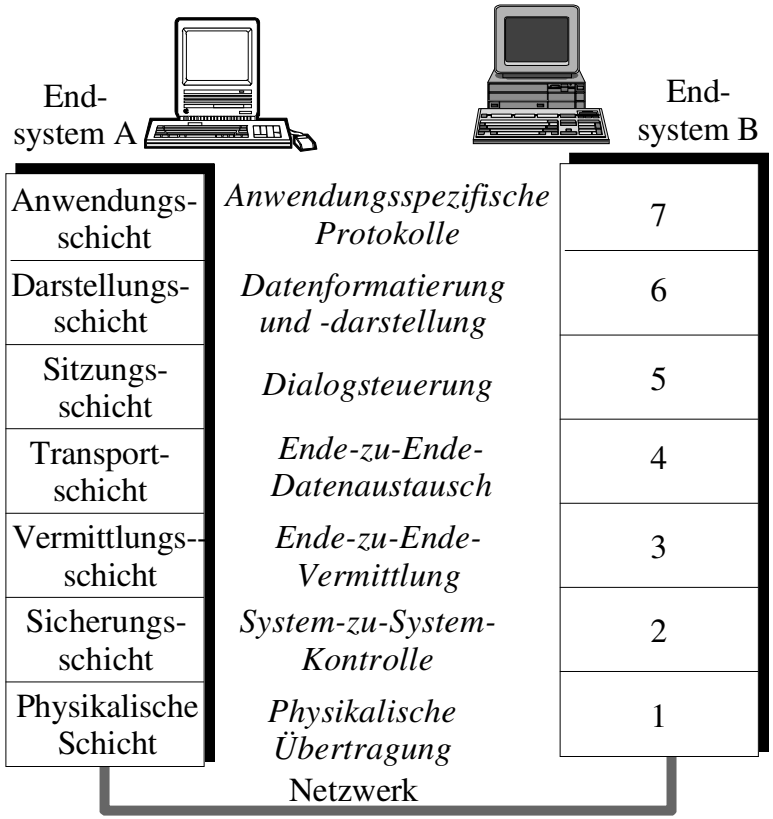
3.5 Neue Algorithmen und Protokolle für Multimedia im Internet

3.1. Herkömmliche Netztechnik in LANs und WANs

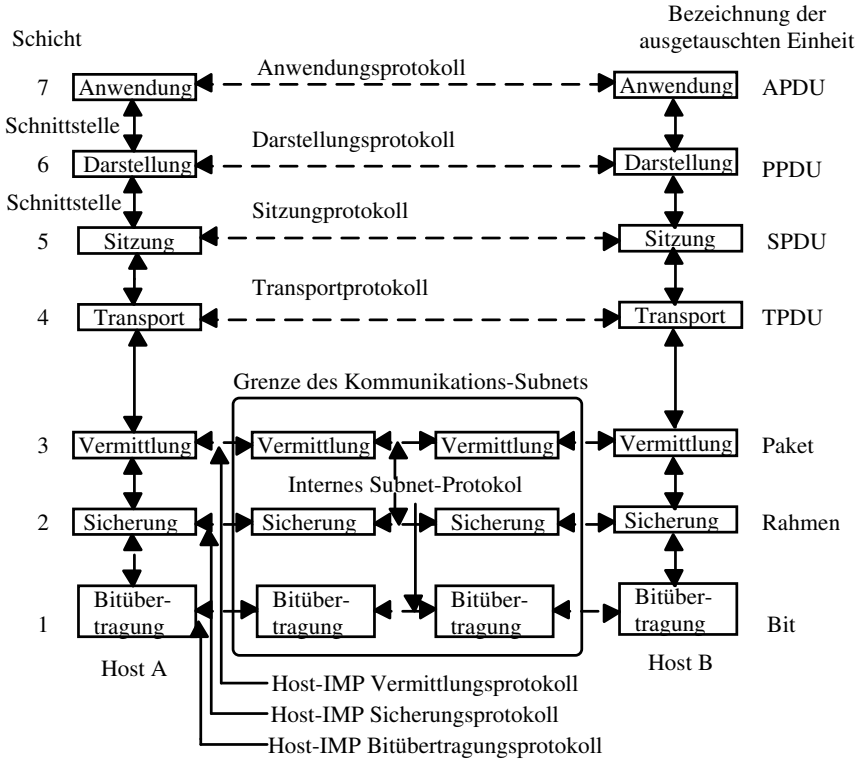
Herkömmliche Rechnernetze wurden für die reine Datenübertragung entworfen, also nur für diskrete Medien. Ihre Unbrauchbarkeit für kontinuierliche Medien wird deutlich, wenn man die Algorithmen und Protokolle in herkömmlichen Rechnernetzen genauer versteht.

3.1.1. Protokollarchitektur in Schichten

Alle herkömmlichen Protokollarchitekturen basieren auf Schichtmodellen.

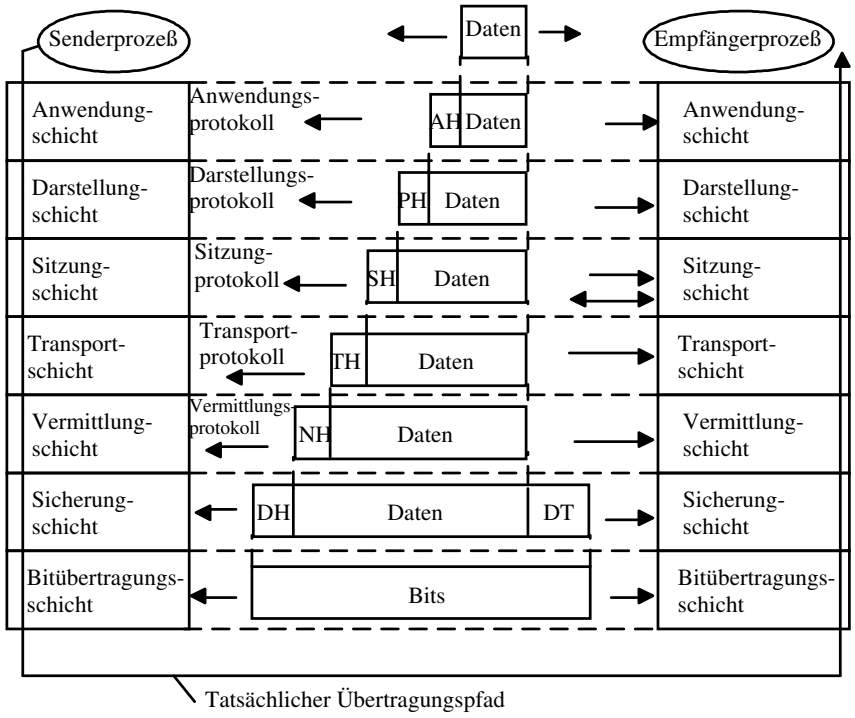


ISO - Referenzmodell für offene Systeme (OSI)



Paket-Header

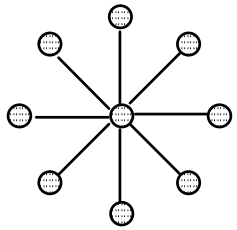
Ein Beispiel dafür, wie sich ein Schichten-Modell auf die Struktur der Datenpakete auswirkt, zeigt die untenstehende Abbildung Jede Schicht fügt einen Header hinzu, Schicht 2 auch einen Trailer.



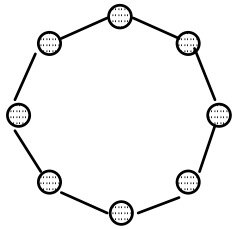
Schichtenmodelle verschiedener Netzarchitekturen

Layer	ISO	Internet	SNA
7	Application	SMTP, FTP, telnet, http	End user
6	Presentation		NAU services
5	Session		Data flow control
4	Transport	TCP	Transmission control
3	Network	IP	Path control
2	Data link control	Data link control	Data link control
1	Physical	Physical	Physical

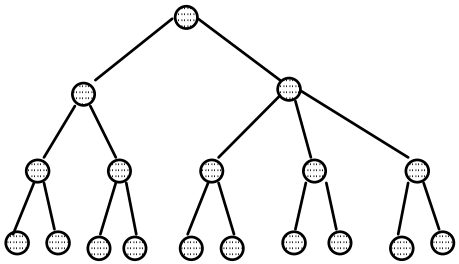
Netztopologien



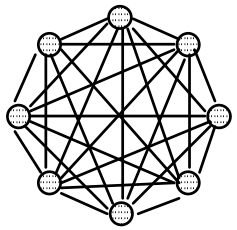
Stern



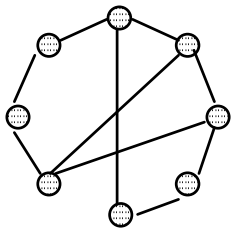
Ring



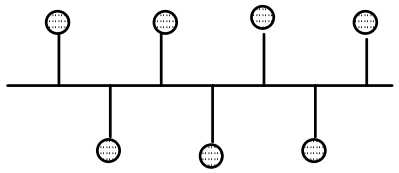
Baum



vollständig vermaschtes Netz



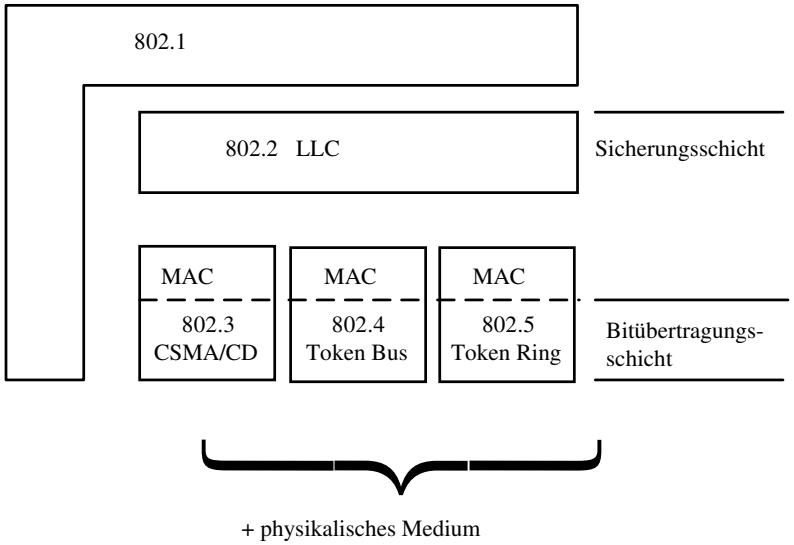
teilweise vermaschtes Netz



Bus

3.1.2. Medienzugangskontrolle und Sicherungsschicht im LAN

- LAN-Charakteristika
  - begrenzte Ausdehnung
  - hohe Übertragungsgeschwindigkeiten
  - niedrige Fehleranfälligkeit und hohe Verfügbarkeit
  - Flexibilität der Rekonfiguration
  - Betrieb in eigener Verantwortung
- Standardisierung durch IEEE und ISO



Punkt-zu-Punkt- vs. Broadcast-Netze

Punkt-zu-Punkt-Netz

- Genau zwei Stationen sind physikalisch verbunden.
- Nachrichten fließen in eine Richtung. Eine explizite Bestätigung ist erforderlich.
- Broadcast erfordert die explizite Vervielfältigung der Nachricht.

Broadcast-Netz

- Mehrere Stationen teilen sich das physikalische Medium.
- Alle Stationen hören alle Nachrichten.
- Falls zwei Stationen gleichzeitig senden, werden beide Nachrichten verfälscht bzw. zerstört.
- Der Sender kann seine eigene Nachricht hören. Falls er genau das hört, was er gesendet hat, kann er annehmen, dass auch der Empfänger die Nachricht korrekt empfangen hat.

Medienzugangskontrolle (MAC)

Problem

- Broadcast-Medium
- Unabhängige Stationen
  => Sendekollisionen

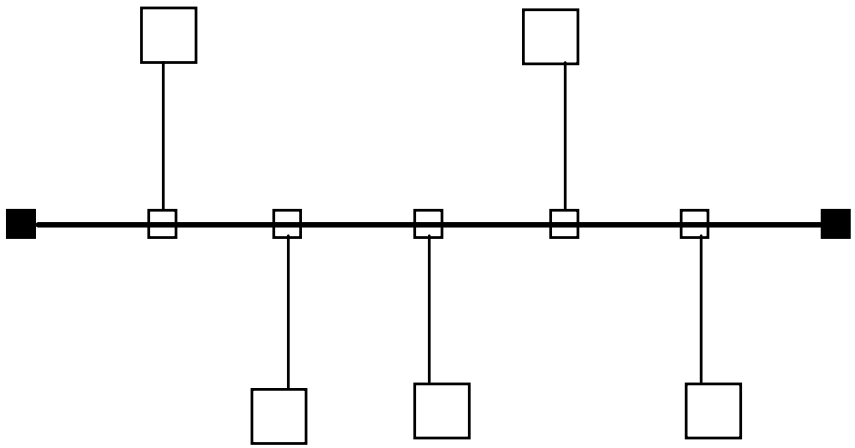
Lösung

- Medienzugangskontrolle
- Zwei Medienbelegungsprinzipien:
  1. Kollisionsentdeckung
     Lasse Kollisionen stattfinden, entdecke sie, wiederhole Übertragung.
  2. Kollisionsverhinderung
     Verwende ein zirkulierendes Token, um den Zugriff auf das Medium zu steuern.

CSMA/CD

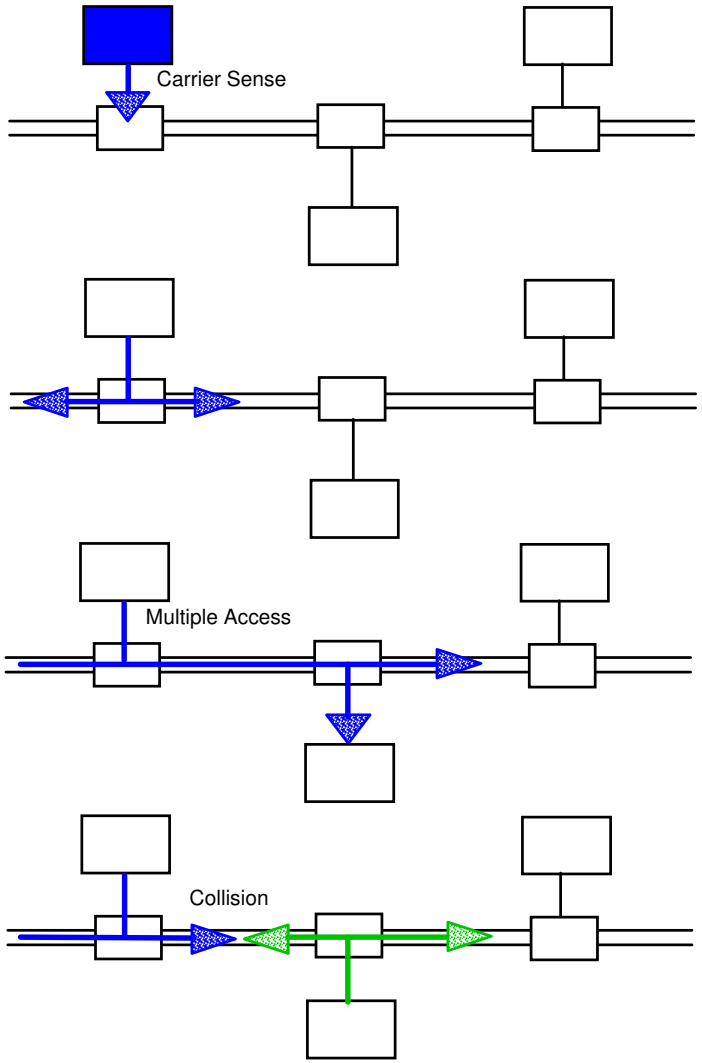
- CSMA/CD = **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection
- ISO IS 8802/3: MAC und Bitübertragungsschicht für CSMA/CD

Topologie: Bus

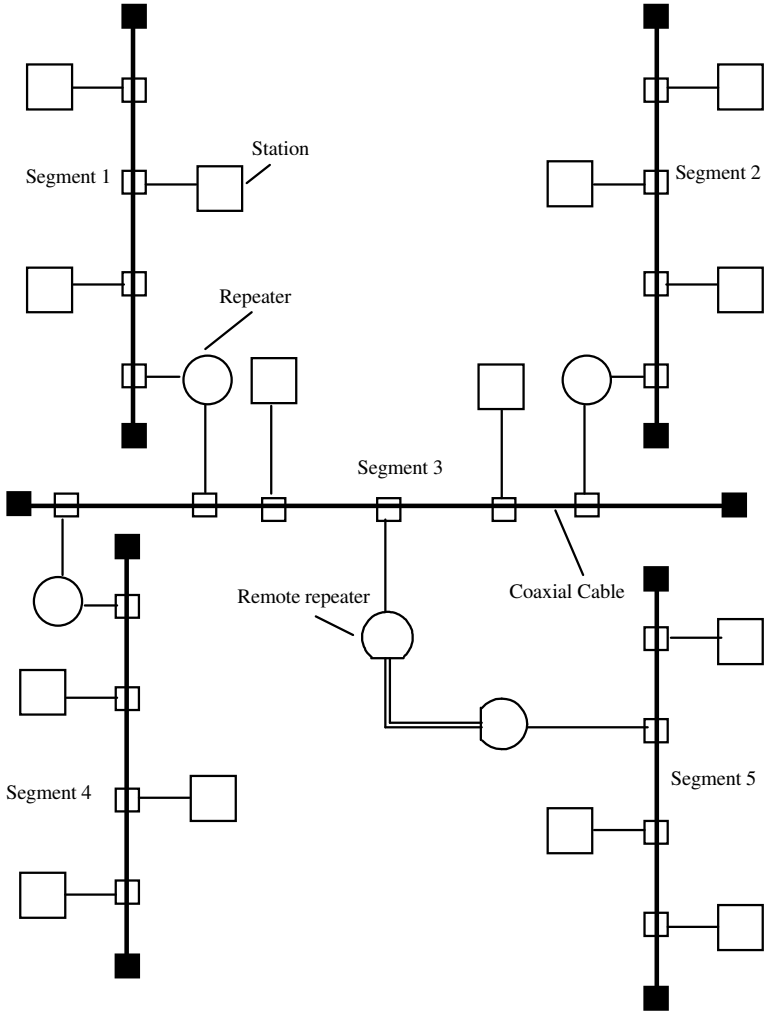


- Bidirektionaler Datenfluss
- Kurze Leitungsführung
- Busunterbrechung = Systemausfall

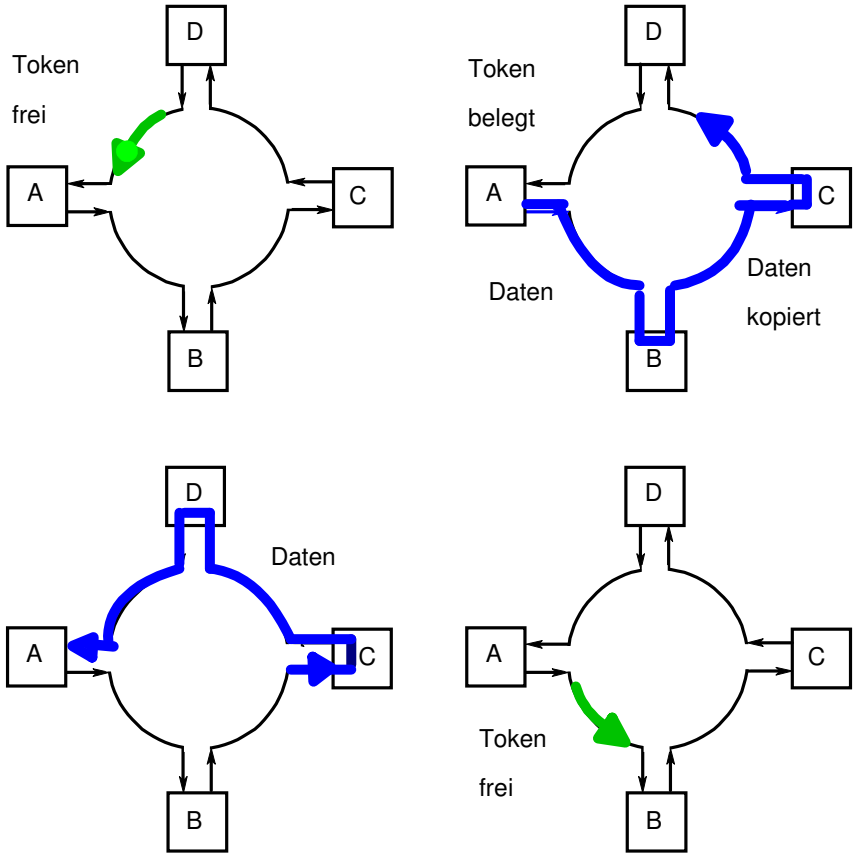
CSMA/CD – Protokoll



Beispiel: Große Konfiguration aus mehreren Segmenten



Token-Ring-Protokoll

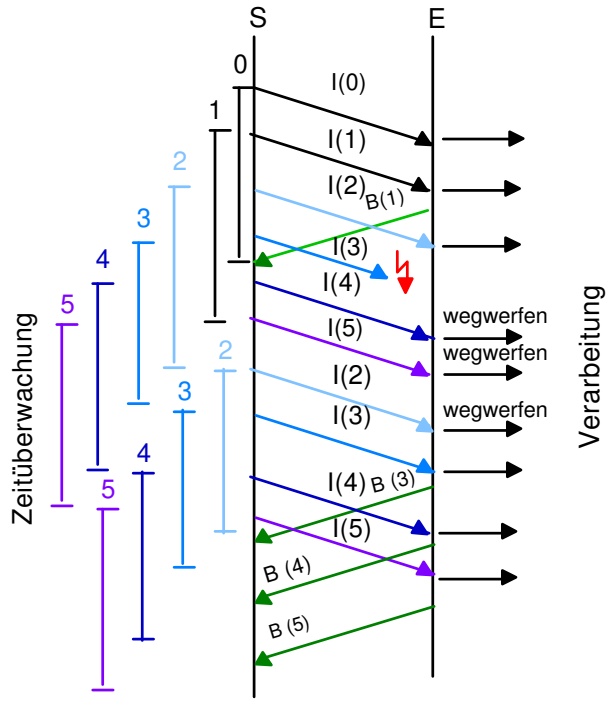


Aufgaben der Sicherungsschicht (Data Link Control)

1. Verdeckung von Übertragungsfehlern zwischen direkt benachbarten Partnern (Erkennung und Behebung)
2. Flusskontrolle

Sequenznummern

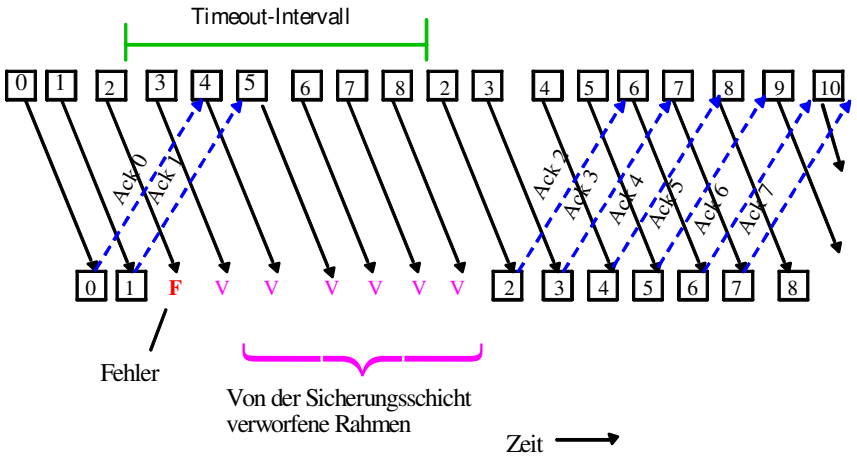
Sequenznummern dienen zur eindeutigen Bezeichnung von Datenpaketen. Sie werden zur Bestätigung durch den Empfänger verwendet. **Mit einer Bestätigung können auch mehrere Informationsblöcke quittiert werden.**



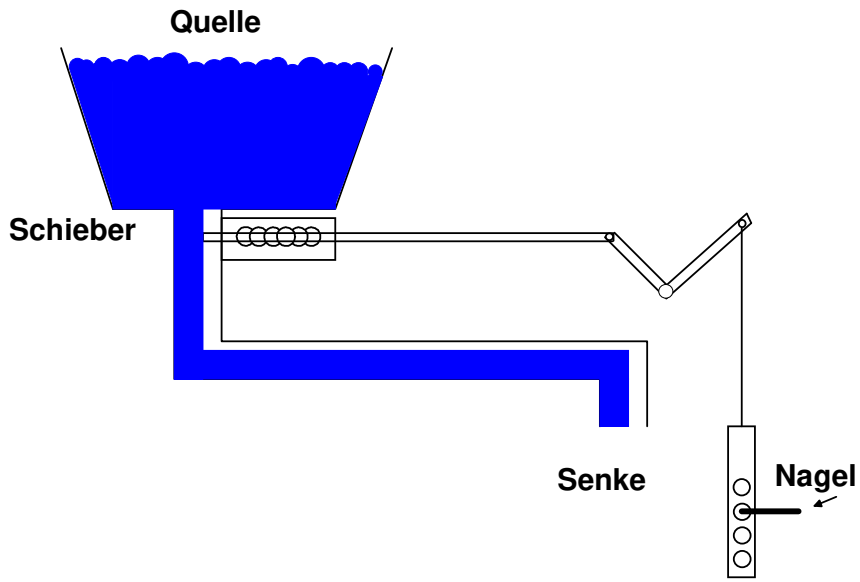
Die Anfangswerte der Sequenznummern werden beim Verbindungsaufbau zwischen Sender und Empfänger vereinbart.

Fehlersicherung durch "Go-back-n"

Im Falle eines Fehlers bleibt das Ack aus. Nach Ablauf des Timers überträgt der Sender **sämtliche** Rahmen ab dem Unbestätigten neu.



Prinzip der Flusskontrolle

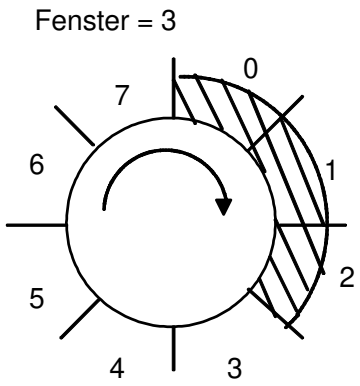


Stellt eine Rückkopplung zur Verfügung, um zu verhindern, dass der Sender den Empfänger überschwemmt.

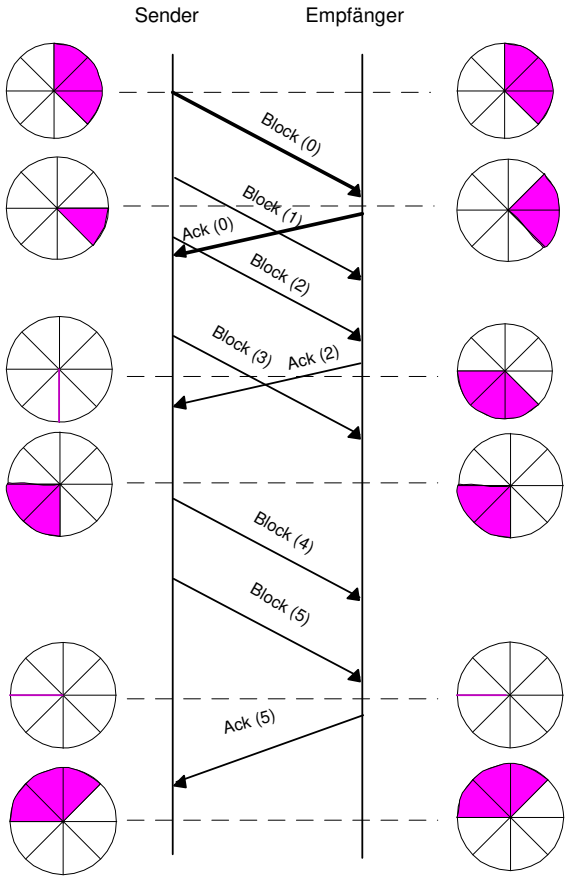
Flusskontrolle mit Schiebefenster

- Nach dem Verbindungsaufbau besitzt der Sender das Recht, so viele Informationsrahmen zu senden, wie durch die Fenstergröße vorgegeben ist.
- Spätestens dann muss vom Empfänger eine Bestätigung eintreffen, ansonsten unterbricht der Sender die Übertragung von Rahmen.
- Der Empfänger kann schon vor dem Erreichen der Fenstergröße Bestätigungen an den Sender übermitteln (Öffnen des Fensters).

Beispiel: Fenstergröße = 3



Schiebefenster



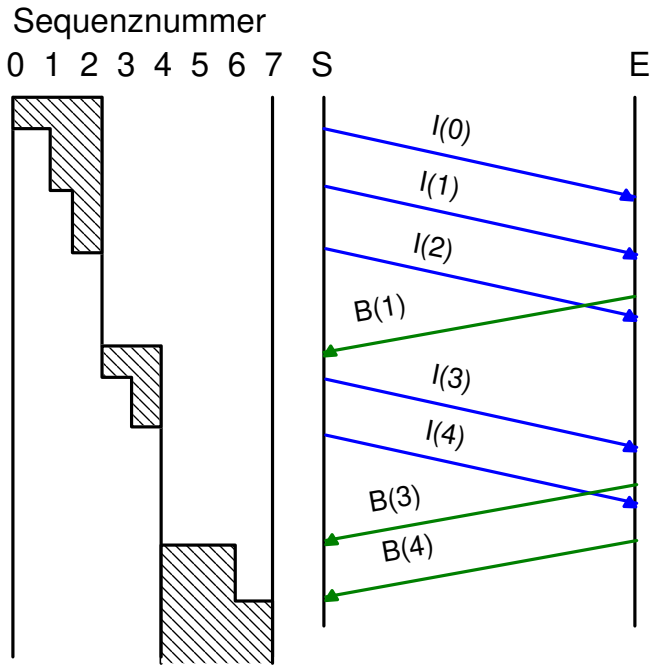
W = 3

Sendeaktionen verschieben den hinteren Zeiger, ACKs den vorderen Zeiger

Eintreffende Blöcke verschieben den hinteren Zeiger, versendete ACKs den vorderen Zeiger

Schiebefenster

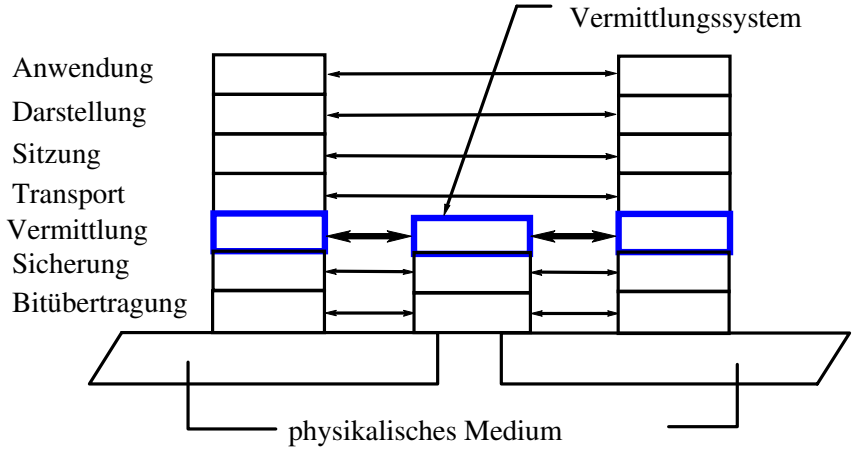
Öffnen und Schließen des Fensters



I = Informationsblock
B = Bestätigung

Dieselben Sequenznummern werden in den meisten Protokollen **sowohl** zur Fehlerkontrolle **als auch** zur Flusssteuerung verwendet!

3.1.3. Paketvermittelte Weitverkehrsnetze



Beispiele für Vermittlungssysteme

- IP-Router im Internet
- X.25–Paketvermittlungsrechner in Datex-P

Virtuelle Verbindung vs. Datagramm

Virtuelle Verbindung

Der Weg durch das Netz wird beim Aufbau der virtuellen Verbindung ausgewählt, d.h. für jede neue virtuelle Verbindung findet in jedem Netzknoten nur einmal eine Wegewahlentscheidung statt. Der ganze über diese virtuelle Verbindung fließende Verkehr nimmt denselben Weg durch das Netz.

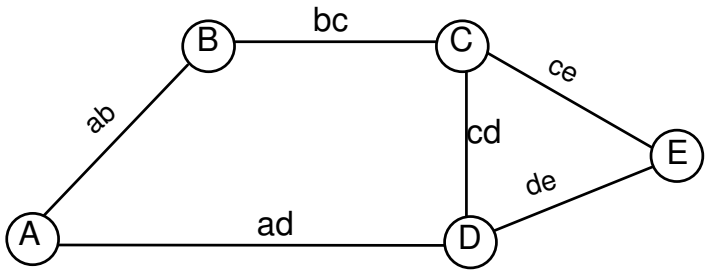
Datagramm

Die Zieladresse bestimmt in jedem Netzknoten auf dem Pfad die ausgehende Leitung. Für jedes Datagramm wird in jedem Knoten erneut eine Wegewahlentscheidung getroffen.

Paketvermittlungsrechner (Router)

In jedem Netzknoten werden Tabellen mit Routing-Informationen verwaltet.

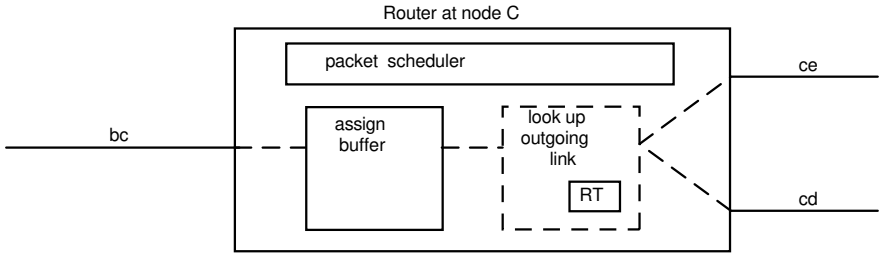
Netztopologie für unsere Beispiele



Funktionsweise eines Routers

RT Routing tabl e at C

From C to	link	cost
A	**bc**	**2**
B	**bc**	**1**
D	**cd**	**1**
E	**ce**	**1**



RT = Routing Table

Leitwegbestimmung (Routing)

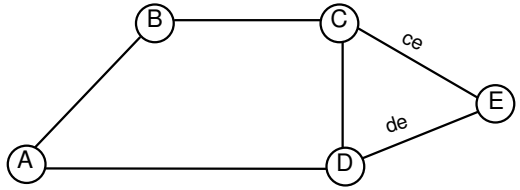
Aufgabe
Leitwegbestimmung für Pakete durch das Netzwerk vom Quellsystem zum Zielsystem

Der **Leitwegbestimmungsalgorithmus** eines Vermittlungsrechners (Knotens), entscheidet, auf welcher Ausgangsleitung ein eingegangenes Paket weitergeleitet wird.

- *Datagramm*: individuelle Entscheidung für jedes Paket
- *Virtuelle Verbindung*: Leitwegbestimmung nur beim Verbindungsaufbau

Idee 1: Lernen durch „Wanderer, kommst du nach Sparta...“

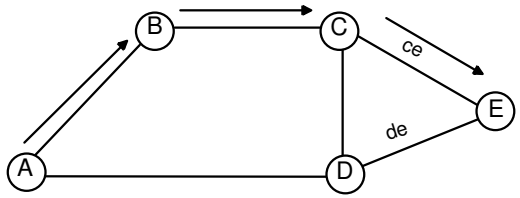
Erlernen der „Hop-Distanzen“ aus eintreffenden Paketen



at E

From E to	link	cost
C	ce	1
D	de	1

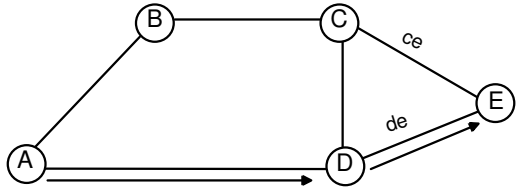
(a) E kennt nur seine direkten Nachbarn



at E

From E to	link	cost
C	ce	1
D	de	1
A	ce	3

(b) Erstes Paket von A ist über ce eingetroffen



at E

From E to	link	cost
C	ce	1
D	de	1
A	de	2

(c) Neues Paket von A ist über de eingetroffen

Idee 2: Jeder kennt die gesamte Topologie

Anwendung des Algorithmus "Kürzeste Pfade" von Dijkstra

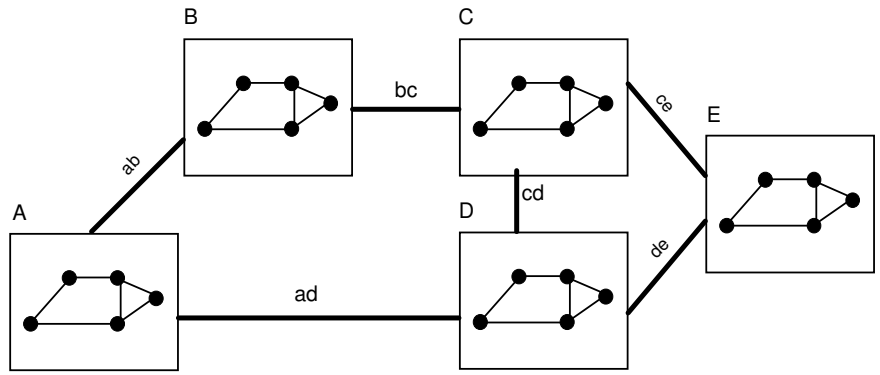
Konstruktion eines Baumes der kürzesten Pfade von einem gegebenen Startknoten A aus.

- Beginne mit dem Ausgangsknoten A.
- Wiederhole, bis alle Knoten verarbeitet sind:
  Füge von allen noch nicht im Baum befindlichen Nachbarknoten denjenigen Knoten hinzu, der zu dem Ausgangsknoten den minimalen Abstand hat.

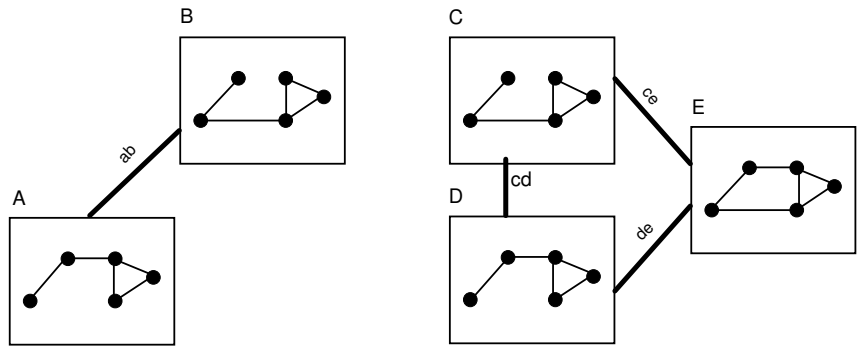
(siehe zum Beispiel: T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen. BI-Wissenschaftsverlag, 1993)

Probleme:
Wie erfährt jeder Knoten stets die aktuelle Gesamt-Topologie des Netzes? Wie kann man Inkonsistenzen in Übergangssituationen verhindern, die eventuell zu Routing-Schleifen führen?

"Full Topology"- Routing (1)

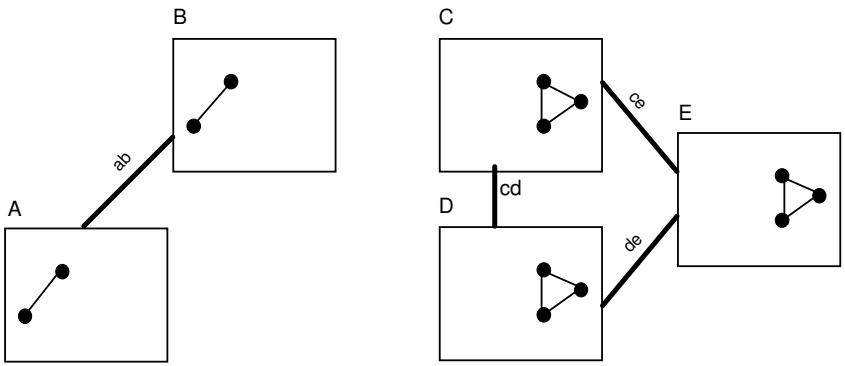


(a) Netz im stabilen Zustand



(b) Links bc und ad sind ausgefallen

“Full Topology“- Routing (2)



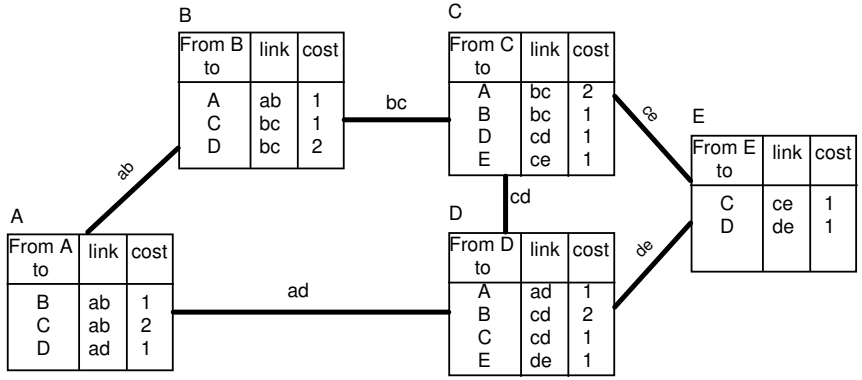
(a) Nach einer weiteren Nachrichtenrunde

“Full Topology”-Routing heißt auch “link state”-Routing. Die Knoten verwalten eine “link state database”, in der sie möglichst aktuelle Informationen über die **gesamte Topologie** des Netzes bereit halten.

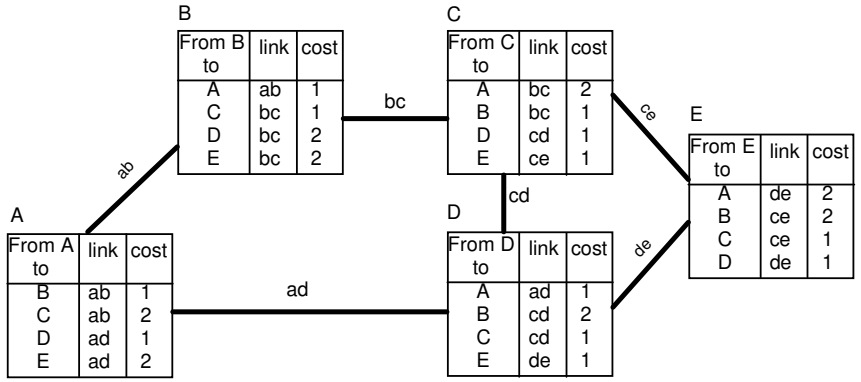
Das erste Routing-Protokoll in der IP-Schicht des Internet war das RIP (Routing Information Protocol), ein Distanzvektorprotokoll. Inzwischen wird fast nur noch OSPF eingesetzt (“Open Shortest Path First”), ein “Full Topology”-Protokoll. Die Implementierung ist zwar wesentlich komplexer, aber Änderungen der Netztopologie werden sehr viel schneller bekannt, was bei großen Netzen besonders wichtig ist.

Idee 3: Routing mit Distanzvektoren

Ein Kompromiss



(a) E kommt als neuer Knoten hinzu

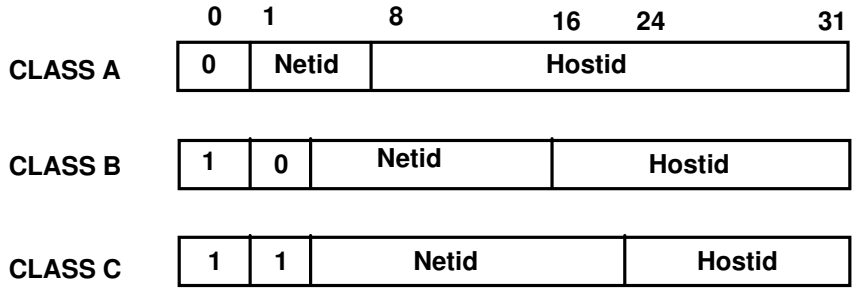


(b) Nach einer weiteren Nachrichtenrunde

Für das Routing gibt es spezielle Nachrichtenpakete (“routing table update messages”).

Internet-Adressierung

- Hierarchische Adresse mit Netz- und Hostidentifikationsnummer (netid bzw. hostid)
- Traditionell gab es drei Formate für Subnetze unterschiedlicher Größe:



- Eingebürgert hat sich eine dezimale Schreibweise mit einer Zahl pro Byte

Beispiele

128.10.0.0 für ein großes Ethernet-LAN
192.5.48.0 für ein kleines Ring-LAN

Beobachtungen im Multimedia-Kontext

- **Die herkömmlichen Algorithmen und Protokolle zerstören den isochronen Fluss der Paketströme!**

  Sie erzeugen erhebliche Varianz in der Verzögerung (delay jitter). Dies gilt für
  - **alle** Netzzugangsprotokolle in LANs
  - die Fehlersicherung durch Übertragungswiederholung (Go-Back-n)
  - die Flusskontrolle mit Schiebefenster (sliding window)
  - und viele weitere Algorithmen!

- **Die herkömmlichen Algorithmen und Protokolle bieten keine Garantien für Dienstgüte.**
  Herkömmliche Netze bezeichnet man deshalb auch als **„best effort“-Netze**.

- **Die herkömmlichen Algorithmen und Protokolle bieten keine Unterstützung für Multicast!**

3.2. Dienstgütemerkmale (Quality of Service)

Erkenntnis

Kontinuierliche Medien erfordern **Dienstgütegarantien** im Netz

Idee: Dienstgüte-Vertrag

- Die **Quelle** spezifiziert den generierten Verkehr und verspricht, sich daran zu halten.
- Das **Netz** verspricht die Übertragung mit garantierten Dienstgütemerkmalen.

Dienstgüte-Parameter (QoS-Parameter)

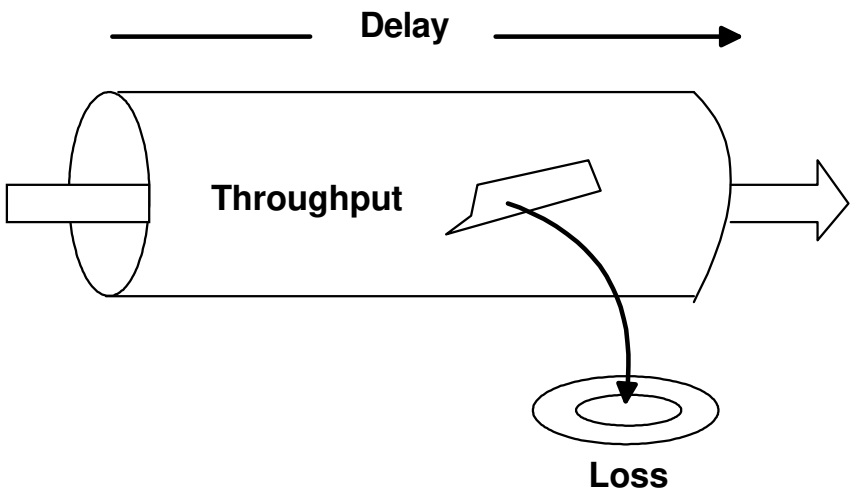
Verkehrsbeschreibung der Quelle

- Verkehrsart: CBR, VBR, UBR, ...
- Stoßweises Verkehrsaufkommen: mittlere Bitrate, maximale Bitrate, Gestalt der Spitzenlast

Dienstgütemerkmale an der Netzschnittstelle

- Verzögerung (delay)
- Varianz der Verzögerung (delay jitter)
- maximale Verlustrate (loss rate)

QoS-Definition: Parameter-Klassen (1)



Throughput (Durchsatz)

- Maximale Langzeitrate = maximale Anzahl von übertragenen Dateneinheiten pro Zeitintervall (z.B. Pakete pro Sekunde oder Bytes pro Sekunde)
- Maximale Burst-Größe
- Maximale Paketgröße

QoS-Definition: Parameter-Klassen (2)

Loss (Verlust)

Sensitivity class. ignore/ indicate/ correct losses
Verlustrate = maximale Anzahl von Verlusten pro Zeitintervall
Verlustgröße = maximale Anzahl von consecutively verlorenen Paketen

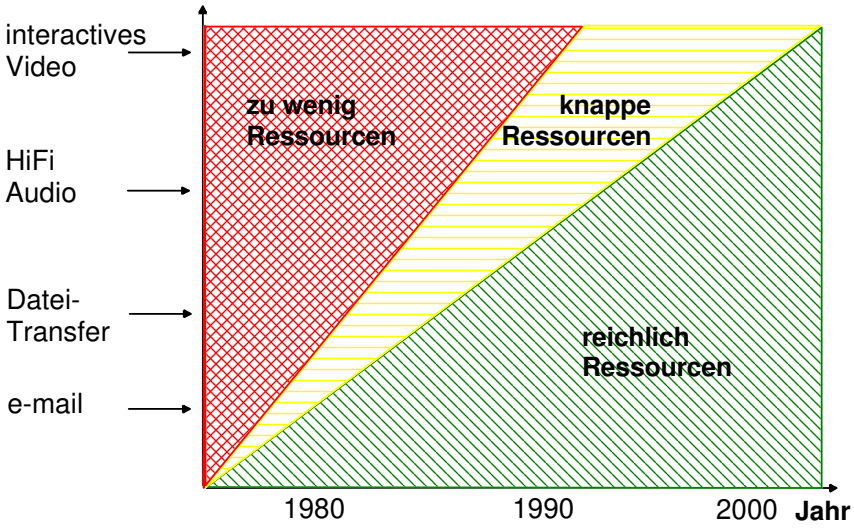
Delay (Verzögerung)

- maximal zulässige Verzögerung zwischen Sender und Empfänger
- maximal zulässige Verzögerungsschwankung (delay jitter) zwischen Sender und Empfänger

QoS und Reservierung von Ressourcen

QoS ist von verfügbaren Ressourcen abhängig.

Ressourcen und Multimedia in der zeitlichen Entwicklung



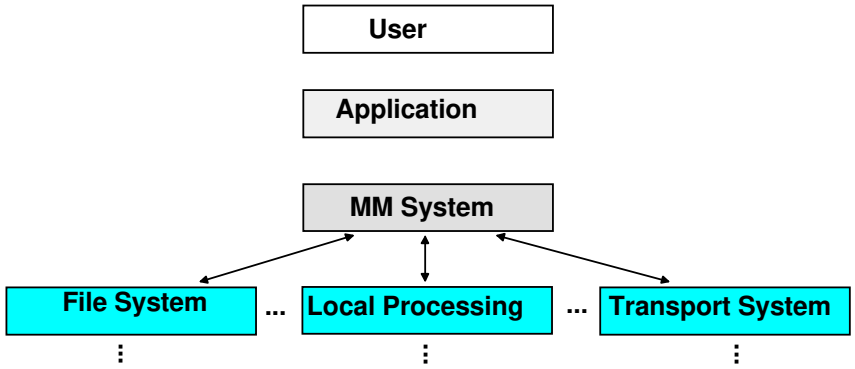
Konkrete Beispiele

Wie die Ressourcen im Netz die QoS-Parameter beeinflussen

- Ein großer Puffer beim Empfänger (playout buffer) erlaubt die Kompensation einer höheren Varianz in der Verzögerung, allerdings auf Kosten einer größeren absoluten Verzögerung
- Bei zunehmender Pufferauslastung in einem Router steigt die mittlere Wartezeit der Pakete an. Neue Verbindungen durch diesen Router sollten nur noch akzeptiert werden, wenn für alle existierenden Verbindungen die Verzögerungsgrenzen eingehalten werden können (Connection Acceptance Control).
- Die CPU-Leistung eines Routers (Verarbeitung der Header) entscheidet über die Maximalzahl und maximale Datenrate der gleichzeitigen Verbindungen.

QoS-Definition auf mehreren Ebenen

Wie bildet man diese aufeinander ab?

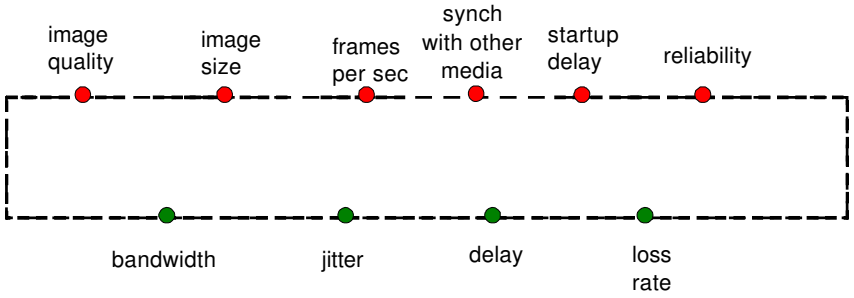


Beispiel:

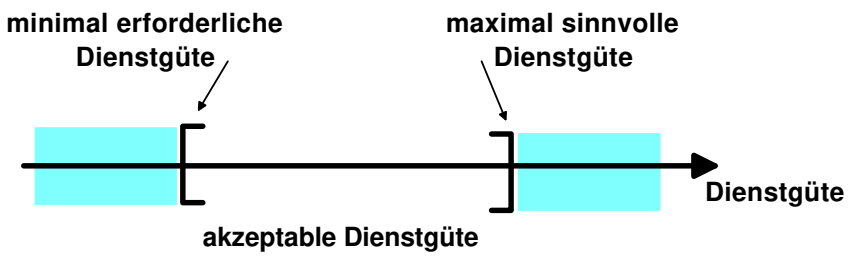
Benutzerebene:	Spiele ein bestimmtes Video
Transportschicht:	Bitrate des Videos, CBR oder VBR, maximale Verzögerung
Netzwerkschicht:	maximale Paketgröße, Paketrate, maximale Verzögerung, maximale Varianz der Verzögerung

Das QoS-Mapping in Netzen ist ein aktuelles Forschungsproblem, bis heute kaum verstanden.

Das Problem der QoS-Abbildung



QoS Definition: minimale und maximale Dienstgüte



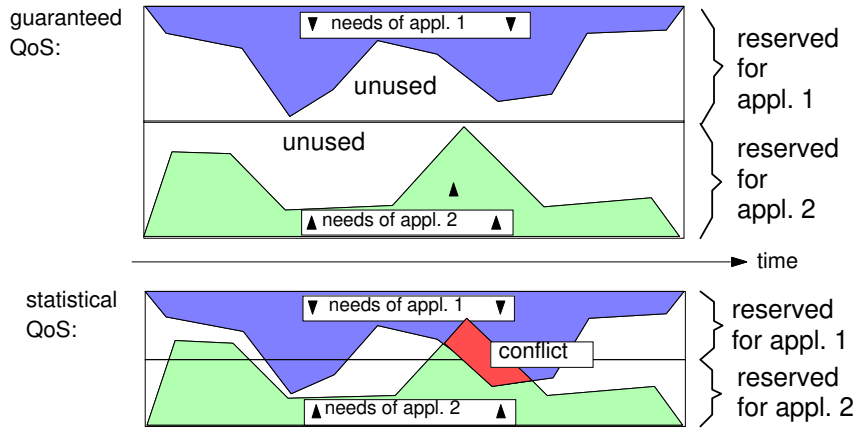
Verhandlung der Applikations-Anforderungen

- Intervall zwischen *erwünschten* und *mindestens erforderlichen* QoS-Werten
- System sollte QoS-Garantie innerhalb des Intervalls zurückgeben (QoS-Vertrag)
- Höhere Dienstgüte muss mehr kosten!

Wie reserviert das Netz Ressourcen?

Problem

Die meisten Anwendungen erzeugen variable Datenraten (z.B. MPEG-Video, Audio mit Schweigeunterdrükkung)

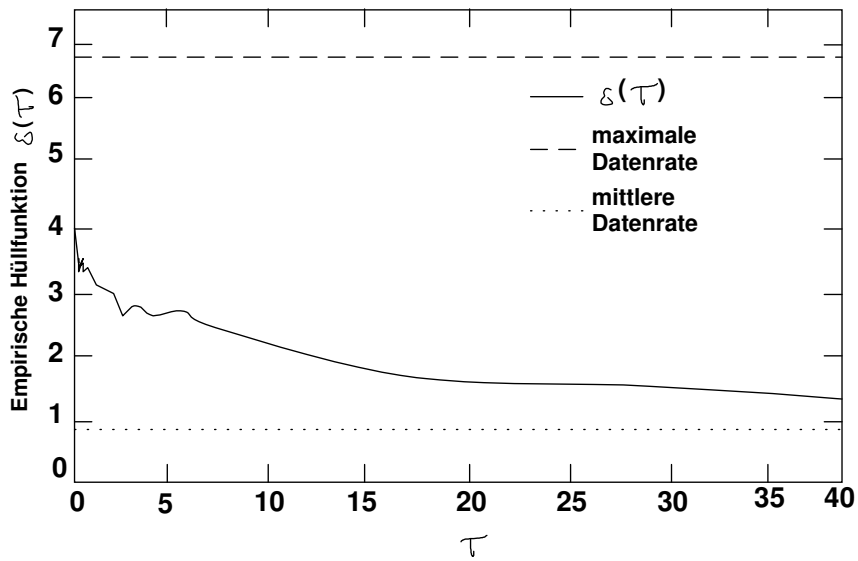


Idee der „einhüllenden Kurve“ (1)

Idee

Beim Abspielen von gespeicherten Videos lässt sich eine **Hüllkurve** für die Datenrate vorausberechnen! Wir beschreiben die tatsächliche Datenrate A eines VBR-Videos durch einen „empirical envelope“ ε in einem Zeitintervall τ:

$$\varepsilon(\tau) = \frac{\max}{t} \cdot A \cdot [\tau, \tau + t]$$



Idee der „einhüllenden Kurve“ (2)

Die Funktion $\varepsilon(\tau)$ ist definiert für alle 0 <= τ <= T, wobei T der Gesamtlaufzeit des Videos entspricht. Für τ = T ist ε (T) die mittlere Datenrate des gesamten Videos.

Wir können $\varepsilon(\tau)$ für ein bekanntes Video verwenden, um damit die Entscheidung über das Akzeptieren einer neuen Verbindung zu treffen: Auch bei voller VCR-Funktionalität wird die (geglättete) Datenrate des Videos $\varepsilon(\tau)/\tau$ nicht überschreiten.

Wir können dann auch im Falle einer Akzeptanz den Playout-Puffer beim Empfänger optimal vorladen: wenn wir sicher sind, dass wir pro Sekunde nicht mehr als τ_0 Frames abspielen werden, können wir einen Puffer der Größe τ_0 beim Empfänger vorladen und sind dann sicher, dass es beim Abspielen niemals Stockungen oder Verluste geben wird.

Deterministisch garantierte Dienstgüte

100%ige Garantie der QoS Werte = „hard bounds“

QoS-Kalkulation basiert auf:

- harten Obergrenzen für den von der Quelle erzeugten Verkehr
- Worst-case-Annahmen bezüglich des Systemverhaltens

Vorteile:

QoS-Garantien auch im schlimmsten Falle erfüllt, hohe Zuverlässigkeit

Nachteile:

- Überreservierung von Ressourcen
- keine Ausnutzung des statistischen Multiplexing-Gewinns im Netz
- unnötige Ablehnung von Reservierungs-Anfragen
- harte Obergrenzen oft nicht wirklich wichtig für die Anwendung (Beispiel: Bildqualität beim Video-Telefon)

Probabilistisch garantierte Dienstgüte

QoS-Werte sind „soft bounds“

QoS-Kalkulation basiert auf:

- Durchschnittswerten/ stochastischen Beschreibungen der Verkehrslast
- probabilistischen Obergrenzen für das Systemverhalten („ in 95% der Fälle soll die Verzögerung < 100 ms sein“)

Vorteile:

- Ressourcen können statistisches Muliplexing ausnutzen;
- mehr Reservierungsanfragen können gleichzeitig berücksichtigt werden

Nachteile:

- QoS kann zeitweise nicht voll erfüllt sein
- sehr schwer implementierbar

QoS Definition: Beispiele

Es gibt keine allgemein anerkannte oder standardisierte QoS-Definition (keinen festen Satz von QoS Parametern)!

Beispiele für QoS-Beschreibungen findet man in:

- Internet (IP): ab Version 6 (IPv6)
- ATM: (Asynchronous Transfer Mode, Q.93b)

QoS und Reservierung

Wichtige Erkenntnis: **ohne Reservierung keine Dienstgütegarantie!**

- Wir brauchen eine lokale Ressourcen-Verwaltung in den Knoten
- Wir brauchen Reservierungsprotokolle
- Wir brauchen eine Überwachung der Quellen („source policing“), ob sie sich bei den eingespeisten Strömen an die Vereinbarung halten

Berechnung von Dienstgütemerkmalen

Wann wird QoS berechnet?

- Neuer Strom benötigt QoS-Garantien für eine spezielle Ressource
- Bestehender Strom möchte seine QoS-Garantien modifizieren

Allgemeine Kalkulationsprozedur:

Falls die neue Anfrage die QoS-Garantien von bestehenden Strömen zerstören würde: Anfrage ablehnen.

Ansonsten die benötigten Ressourcen berechnen und reservieren, entlang des gesamten Pfades, unter Einbeziehung von QoS-Garantien, die bereits an andere Ströme gegeben wurden.

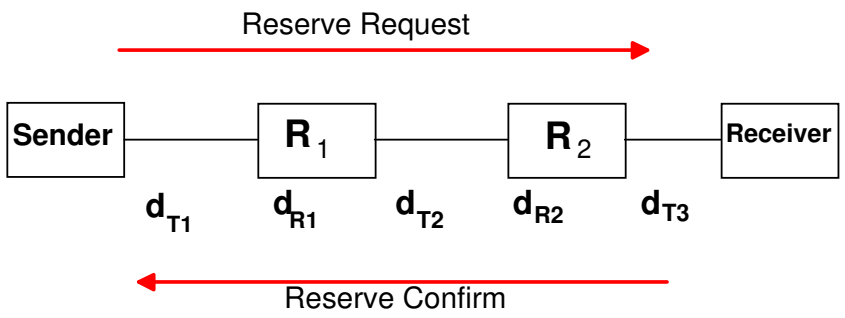
Ein Protokoll zur Reservierung von Ressourcen

Idee

- Verbindungsorientierte Kommunikation
- Reserviere so viel wie möglich auf dem Weg von der Quelle zur Senke
- Ermittle die Gesamtreserve an der Senke
- Verteile die Gesamtreserve über alle Zwischenknoten und relaxiere sie auf dem Rückweg mit der Bestätigungsnachricht

Reservierungsprotokoll: Beispiel

Topologie



QoS-Anforderung: D_{max} = 140 ms

d_{T1} = 5 ms
d_{T2} = 15 ms
d_{T3} = 5 ms
d_{R1} = 10 ms
d_{R2} = 25 ms

Reservierungsanforderung

Reserve Request

$\Sigma \; d_i$ = 60 ms
Reserve: 80 ms

Strategie zum Beispiel: Relaxation gleichmäßig auf die beiden Router R_1 und R_2

Reserve Confirm

d_{maxR1} = 50 ms
d_{maxR2} = 65 ms

Zusammenfassung QoS

Hauptprobleme

- Spezifikation von QoS-Parametern (welche?)
- Abbildung der QoS-Parameter zwischen der Anwendung, den Schichten des Netzes und dem Betriebssystem
- Berechnung des lokalen Ressourcenbedarfs
- QoS-Überwachung während der Dauer der Verbindung