



## 4 Kommunikation von Multimediadaten

### 4.1 Dienstgüte

# Ausgangssituation in Netzwerken



- herkömmliche Netz-Algorithmen und Protokolle zerstören isochronen Fluss der Paketströme
  - Sie erzeugen erhebliche Varianz in der Verzögerung (delay jitter)
- Dies gilt u.a. für
  - Netzzugangsprotokolle in LANs
    - z.B. CSMA/CD, TokenRing
  - Fehlersicherung durch Übertragungswiederholung
  - Flusskontrolle mit Schiebefenster (sliding window)
- „best effort“-Netze
- selten echte Unterstützung für Multicast



# Dienstgüte - Quality of Service



- 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



# QoS-Parameter



- Verkehrsbeschreibung der Quelle
  - Verkehrsart
    - CBR, constant bit rate
    - VBR, variable bit rate
    - UBR, unspecified bit rate
    - ...
  - Stossweises Verkehrsaufkommen (bursty)
    - mittlere Bitrate
    - maximale Bitrate
    - Gestalt der Spitzenlast



# QoS-Parameter



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



# QoS-Definition: Parameter-Klassen



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



# QoS-Definition: Parameter-Klassen



- Loss (Verlust)
  - Sensitivity classes
    - ignore
    - indicate
    - correct losses
  - Verlustrate = maximale Anzahl von Verlusten pro Zeitintervall
  - Verlustgröße = maximale Anzahl von konsekutiv verlorenen Paketen



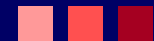
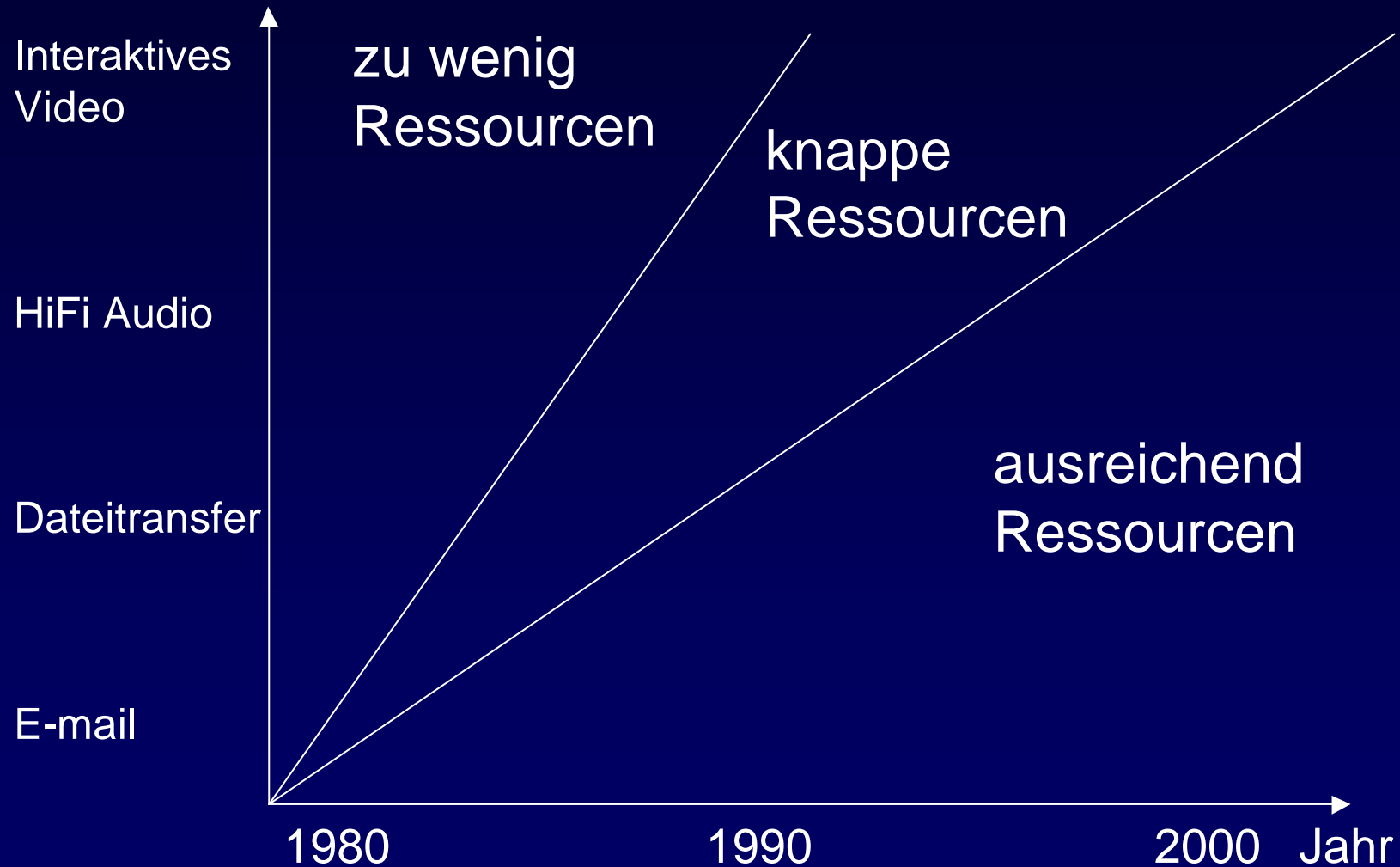
# QoS-Definition: Parameter-Klassen



- Delay (Verzögerung)
  - maximal zulässige Verzögerung zwischen Sender und Empfänger
  - maximal zulässige Verzögerungsschwankung (delay jitter) zwischen Sender und Empfänger
- QoS ist von den verfügbaren Ressourcen abhängig



# Zeitliche Entwicklung



# Netz-Ressourcen und QoS-Parameter



- Ressourcen im Netz beeinflussen QoS-Parameter
- Ein großer Puffer beim Empfänger (playout buffer) erlaubt die Kompensation einer höheren Varianz in der Verzögerung
  - aber auf Kosten einer größeren absoluten Verzögerung



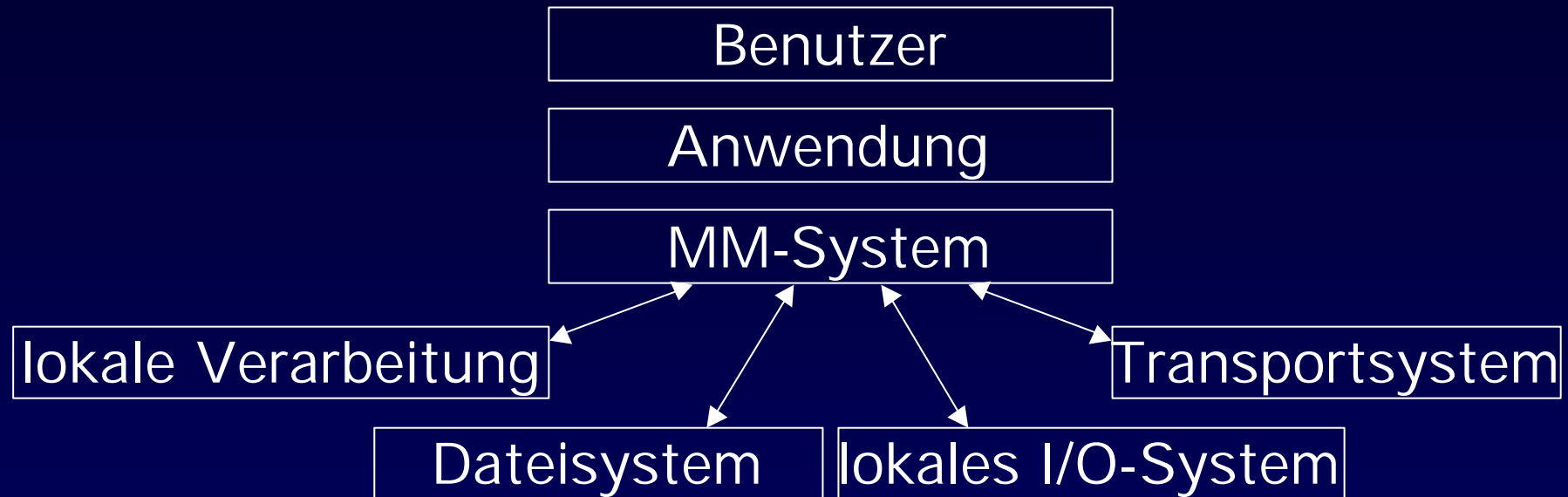
# Netz-Ressourcen und QoS-Parameter



- bei zunehmender Pufferauslastung in einem Router steigt die mittlere Wartezeit der Pakete
  - neue Verbindungen nur dann akzeptieren, wenn für alle existierenden Verbindungen Verzögerungsgrenzen eingehalten werden können (Connection Acceptance Control)
- CPU-Leistung eines Routers entscheidet über die Maximalzahl und maximale Datenrate der gleichzeitigen Verbindungen



# QoS-Ebenen

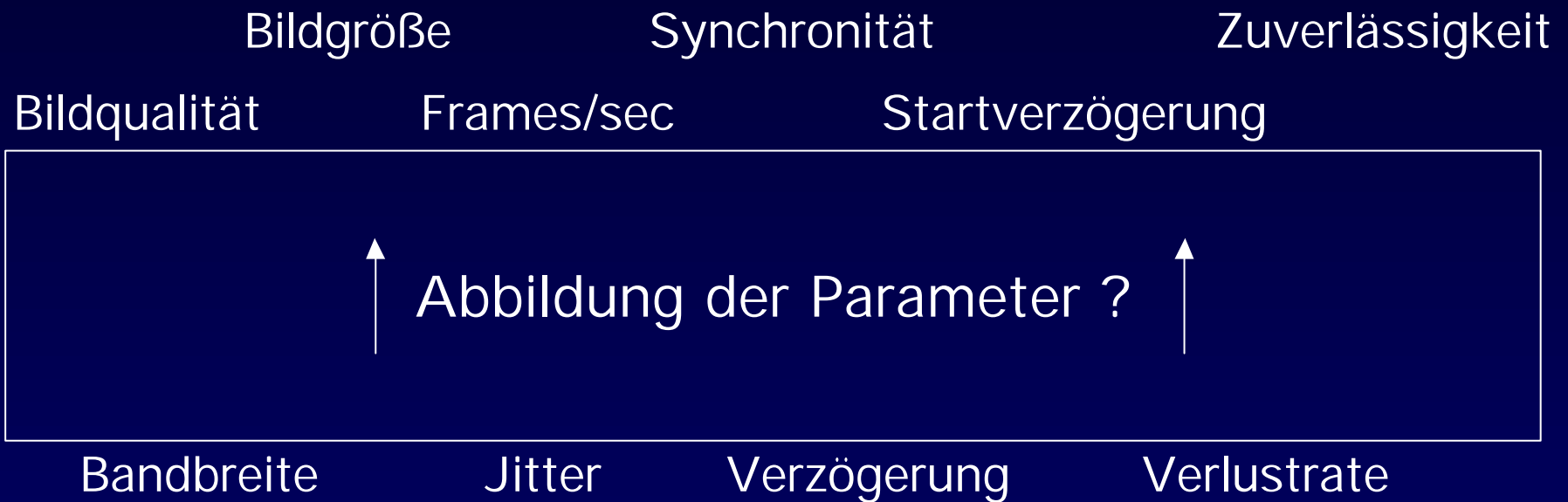


## ■ Beispiele

- Benutzerebene: Spiele ein Video
- Transportschicht: Bitrate, CBR oder VBR, maximale Verzögerung
- Netzwerkschicht: maximale Paketgröße, Paketrage, maximale Verzögerung, maximale Varianz der Verzögerung



# QoS-Abbildung



# minimale und maximale Dienstgüte



- Verhandlung der Anforderungen der Anwendung
  - Intervall zwischen erwünschten und mindestens erforderlichen QoS-Werten
  - System sollte QoS-Garantie innerhalb des Intervalls zurückgeben
    - QoS-Vertrag
  - Höhere Dienstgüte sollte mehr kosten



# Ressourcenreservierung



## ■ Problem

- viele Anwendungen erzeugen variable Datenraten
  - z.B. MPEG-Video, Audio mit Schweigeunterdrückung



# 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



# Deterministisch garantierte Dienstgüte



- Vorteile
  - QoS-Garantien auch im Worst Case erfüllt
  - hohe Zuverlässigkeit
- Nachteile
  - Überreservierung von Ressourcen
  - keine Ausnutzung des statistischen Multiplexing-Gewinns im Netz
  - unnötige Ablehnung von Reservierungsanfragen
  - harte Obergrenzen oft nicht zwingend für die Anwendung



# Probabilistisch garantierte Dienstgüte



- QoS-Werte sind „soft bounds“
- QoS-Kalkulation basiert auf
  - Durchschnittswerten bzw. stochastischen Beschreibungen der Verkehrslast
  - probabilistischen Obergrenzen für das Systemverhalten



# Probabilistisch garantierte Dienstgüte



- Vorteile
  - Ressourcen können statistisches Multiplexing ausnutzen
  - mehr Reservierungsanfragen können gleichzeitig berücksichtigt werden
- Nachteile
  - QoS kann zeitweise nicht voll erfüllt sein
  - schwerer implementierbar



# QoS Definitionen



- Es gibt keine allgemein anerkannte oder standardisierte QoS-Definition
  - D.h. kein fester Satz von QoS Parametern
- Beispiele für QoS-Beschreibungen
  - IP Version 6 (IPv6)
    - Flow Label (24 bit) kennzeichnet Pakete eines Stroms
    - Priority Field (4 bit) zur Klassifizierung nach Wichtigkeit
  - ATM
    - Dienstkategorien auf der Basis von QoS-Parametern
    - CBR, UBR, rt-VBR, nrt-VBR, ABR



# QoS und Reservierung



- ohne Reservierung keine Dienstgütegarantie
- Voraussetzungen und Notwendigkeiten
  - lokale Ressourcen-Verwaltung in den Knoten
  - Reservierungsprotokolle
  - Überwachung der Quellen
    - „source policing“
    - halten sie sich an die Vereinbarung?



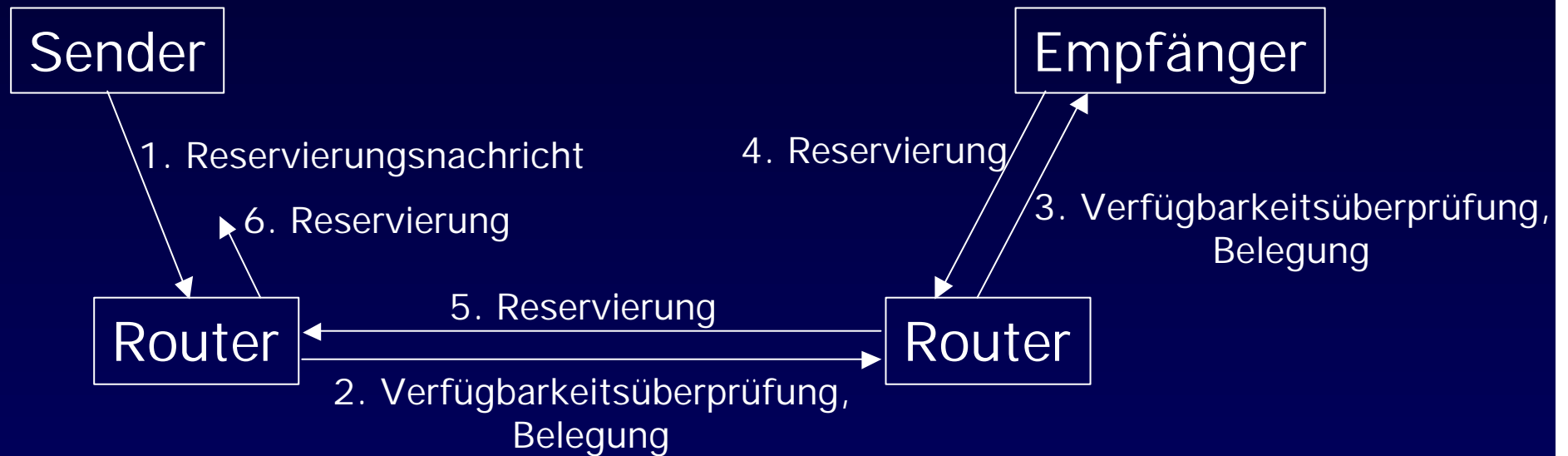
# Berechnung von Dienstgütemerkmalen



- Wann wird QoS berechnet?
  - Neuer Strom benötigt QoS-Garantien für eine spezielle Ressource
  - Bestehender Strom möchte seine QoS-Garantien modifizieren
- Allgemeine Kalkulationsprozedur
  - Würde die neue Anfrage die QoS-Garantien von bestehenden Strömen zerstören, dann ablehnen
  - Sonst,  
benötigte Ressourcen entlang des gesamten Pfades berechnen und reservieren
    - Dabei sind andere bereits vergebene QoS-Garantien zu berücksichtigen



# Senderorientierte Reservierung



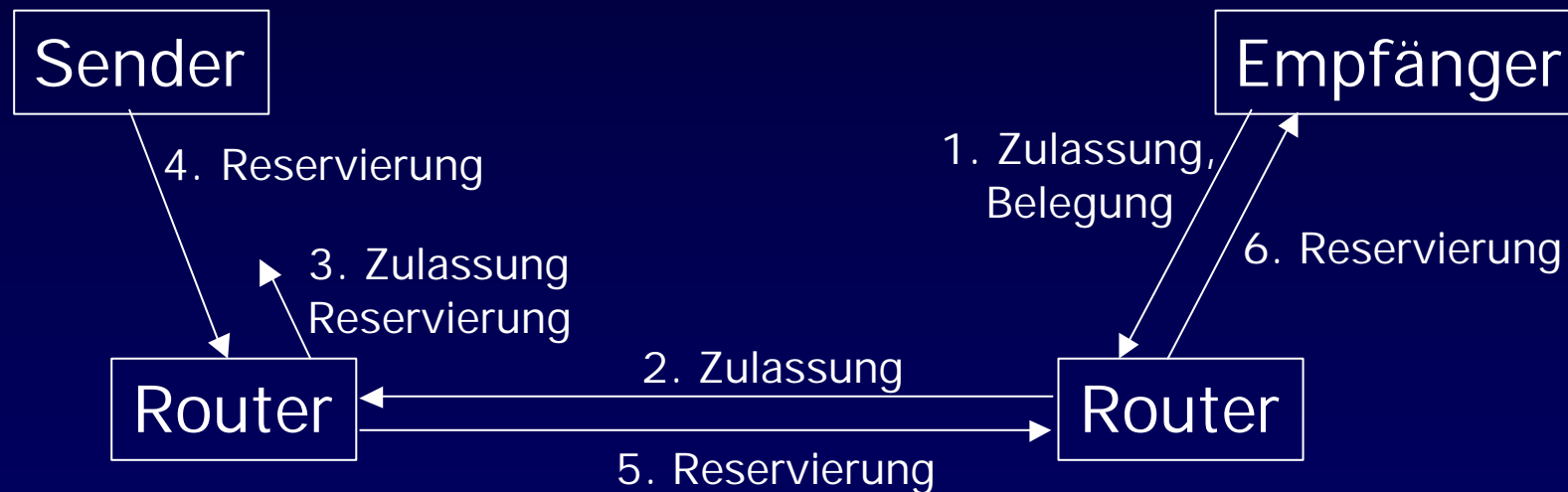
- Fallbeispiel: ST-II (stream protocol II)



# Empfängerorientierte Reservierung



- Voraussetzung
  - Sender schickt Information zur Datenquelle



- Fallbeispiel:  
RSVP (resource reservation protocol)





## 4 Kommunikation von Multimediadaten

### 4.2 Multicasting

# Multimedia und Multicast



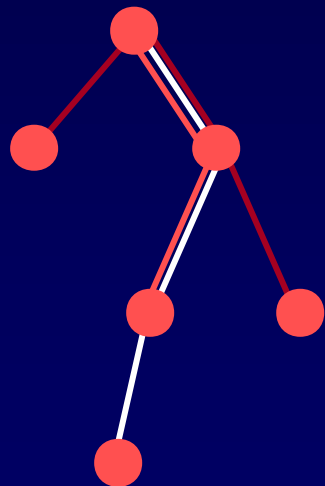
- Multimedia-Anwendungen besitzen oft 1:n Kommunikationsbeziehungen
- Beispiele
  - Videokonferenz
  - Tele-Kooperation (CSCW)
  - Video-on-Demand
  - Broadcast-Medien
- Digitale Video- und Audioströme haben hohe Datenraten ( $\geq 1,5$  MBit/s)
- Realisierung durch n Einzelverbindungen überlastet die meisten Netze



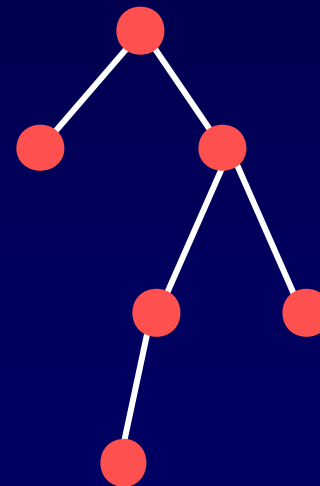
# Multicasteigenschaften



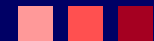
- Multicast im Netz
  - verringert die Last bei den Sendern
  - verringert die Last auf den Teilstrecken



Einzelverbindungen



Multicast-Verbindung



# Anforderungen an Multicast für MM



- Unterstützung von isochronen Datenströmen mit garantierter Dienstgüte für eine vereinbarte Verkehrslast  
(Vertragsmodell, „flowspec“)
  - Maximale Ende-zu-Ende-Verzögerung (delay)
  - Maximale Varianz in der Verzögerung (delay jitter)
  - Zuverlässigkeit



# Anforderungen an Multicast für MM



- Reservierung von Ressourcen auf allen Verbindungen und bei allen Knoten im Netz
  - Bandbreite
  - CPU-Leistung
  - Pufferplatz
  - Schedulability
- Konzepte und Protokolle für eine Gruppenadressierung
- Algorithmen für dynamisches Hinzufügen und Löschen von Teilnehmern



# Multicast in LANs



- Ethernet, Token Ring, FDDI
  - Topologie hat Broadcast-Eigenschaft
  - Schicht-2-Adressen erlauben Gruppenadressen
  - IP unterstützt Gruppenadressen
  - Ab Schicht 3 wurden in der Internet-Protokollsuite bisher nur Peer-to-Peer - Adressen unterstützt
    - aber im weltweiten Internet muss Multicast auch WAN-Strecken überbrücken



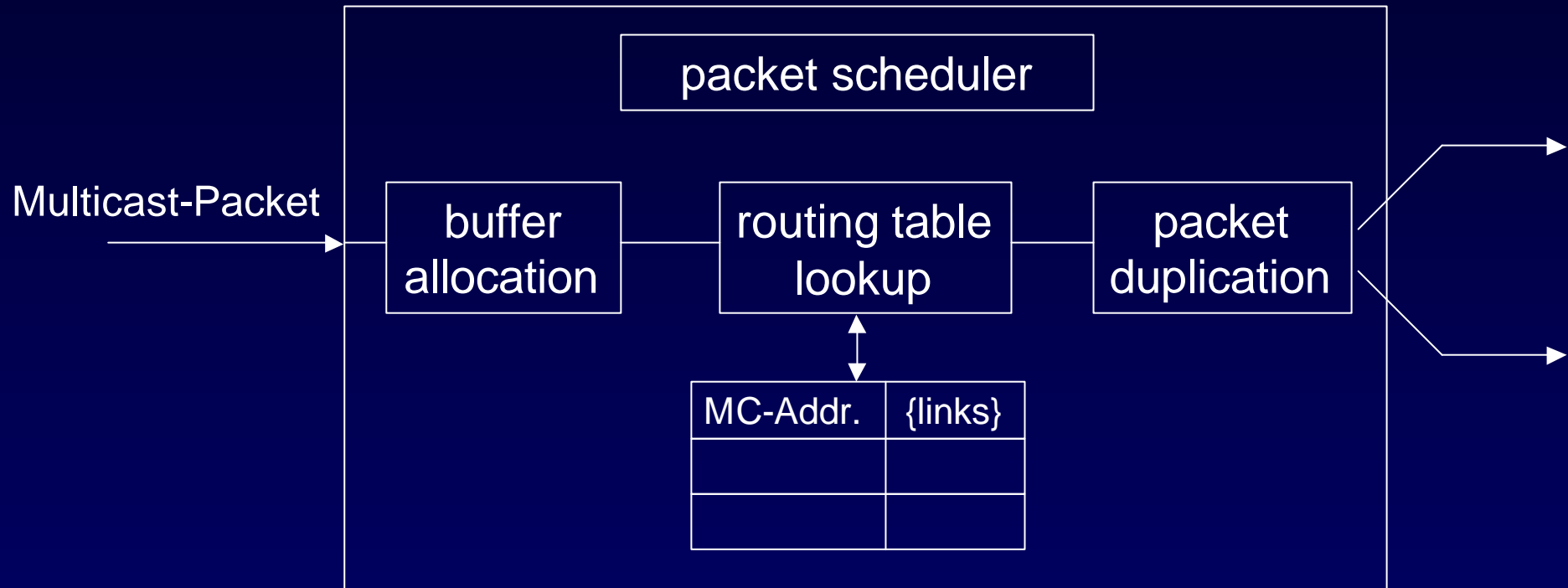
# Multicast in der Netzwerkschicht



- Prinzip
  - Duplizierung von Paketen so tief unten im Multicast-Baum wie möglich
- erfordert Multicast-Adressierungsschema in Schicht 3 und mehr Intelligenz in den Routern
- verbindungslos oder verbindungsorientiert?



# Multicast Router





## 4 Kommunikation von Multimediadaten

### 4.3 Medienskalisierung und Medienfilter

# Medienskalierung



- Skalierung = Anpassung des Datenvolumens einer verteilten Anwendung an die freien Kapazitäten der Ressourcen
- Anforderungen
  - schnelle und exakte Anpassung an freie Kapazität
  - robust gegen Paketverlust
  - universell einsetzbar (LAN, WAN, Kodierer, Multicast)



# Adaptivität bei Ressourcenmangel



- Nicht-Verfügbarkeit von Ressourcen kann auftreten
  - Ressourcen, die nicht über QoS-Reservierungsmechanismen verfügen
  - variable Bitraten
  - falsche Ressourcen-Bedarfsspezifikation der Anwendung



# Adaptivität bei Ressourcenmangel



- Ziele
  - dynamische Angleichung der Ressourcennutzung, um Überlastung der Ressourcen zu vermeiden
  - verringerte Dienstgüte soll durch den Nutzer möglichst wenig wahrgenommen werden
    - z.B. Verminderung der Bildqualität statt der Bildwiederholrate



# transparente - nichttransp. Skalierung



- Transparent
  - Skalierung nur innerhalb der Transportschicht
  - unsichtbar für das Anwendungsprogramm
- Nicht transparent
  - Dienstprimitive werden zur Verfügung gestellt
  - das Anwendungsprogramm reagiert darauf und agiert damit



# Skalierungsformen für Bewegtbilder



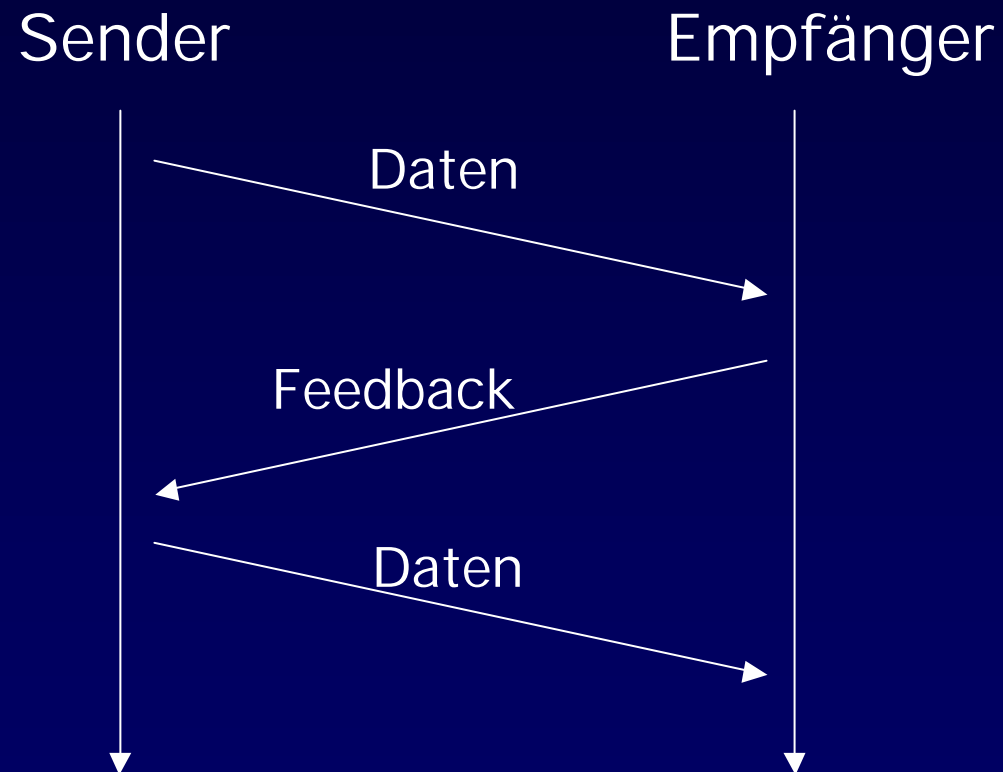
- zeitliche Skalierung
  - Anzahl der übertragenen Bilder pro Sekunde verringern
- räumliche Skalierung
  - Reduzieren der Bildgröße
- Frequenzskalierung
  - Weglassen von Frequenzen (z.B. DCT)
- Amplitudenskalierung
  - Reduzieren der Farbtiefe der Pixel oder der Anzahl Bits für die Graustufen



# Prinzipieller Ablauf der Skalierung



- Monitoring
- Feedback
- Adaption



# Skalierung vs. Reservierung



- Reservierung
  - in immer mehr Netzwerken verfügbar
  - sollte auch in höheren Schichten angewendet werden
- Skalierung
  - komplementäre Technik
  - für existierende Netzwerke geeignet
    - z.B. für IP V4
  - benötigt geeignete Kodierung des Datenstroms
    - z.B. hierarchische Video-Kodierung



# Skalierung vs. Reservierung



- Reservierung und Skalierung können sich ergänzen
  - Beispiel
    - ein Bitstrom mit Reservierung überschreitet kurzfristig die Obergrenze der vereinbarten Bitrate
    - der Strom wird beim Sender herunterskaliert



# Layered Encoding



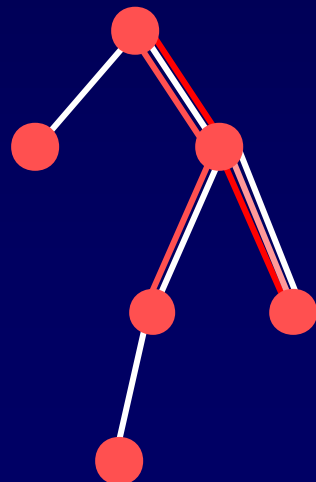
- 1 Transportstrom = n Netzwerkströme
- Basis-Netzwerkstrom (layer 0)
  - Grobbild
  - Übertragung über reservierte und garantierte Bandbreite oder mit hoher Priorität der Pakete
- Zusätzliche Netzwerkströme (higher layers)
  - mehr Daten für bessere Bildqualität
  - Übertragung über eine "best-effort"-Verbindung oder mit niedrigerer Priorität der Pakete



# Medienfilter



- Unterschiedliche Empfänger erfordern unterschiedliche Qualitäten der Medienströme
  - Kompressionsverfahren (z.B. MPEG-2) bieten eine hierarchische Kodierung (layers)
  - Datenstrom kann in Unterströme geteilt werden
  - nur der Teil der Information, der vom Empfänger gewünscht wird, wird dorthin übertragen
  - Irrelevante Teile werden von Filtern im Netz entfernt



- Layer 0, Basisqualität
- Layer 1, Delta für mittlere Qualität
- Layer 2, Delta für hohe Qualität



# Adaptive Anwendungen



- Idee
  - Nicht das Netz passt sich an den Bedarf der Anwendung an, sondern die Anwendung an die aktuelle Belastungssituation des Netzes
  - verbessert vor allem den Betrieb von Multimedia-Anwendungen in Netzen ohne QoS-Unterstützung



# Adaptive Anwendungen



- Beispiel
  - Während ein Video aus einer Video-on-Demand-Datenbank abläuft, steigt die Netzbelastung
  - das Netz meldet Verstopfungsgefahr an alle Quellen
  - die Video-Quelle verändert die Quantisierungstabelle des MPEG-Encoders und generiert eine niedrigere Bitrate bei niedrigerer Bildqualität
- bei klassischen Kontrollalgorithmen hingegen
  - Netzverstopfung reduziert die Paketrate
  - die Bildwiederholrate beim Empfänger sinkt

