

Internetanwendungstechnik

Dienstgüte (Quality of Service, QoS)

Gero Mühl

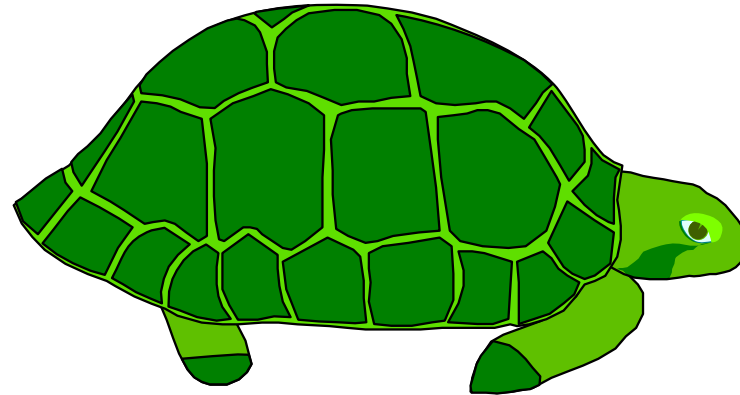
Technische Universität Berlin

Fakultät IV – Elektrotechnik und Informatik

Kommunikations- und Betriebssysteme (KBS)

Einsteinufer 17, Sekr. EN6, 10587 Berlin

World Wide Wait



- > Synonym für Probleme im heutigen Internet
- > Das Internet ist „best-effort“ – „so gut es eben geht“
- > Es gibt keine **Garantien** für Leistung und Zuverlässigkeit
- > Lokal: Betriebssystem steuert Ressourcenvergabe.
Aber: Wer steuert die Ressourcenvergabe im Internet?

Probleme im Internet

> Neue Anwendungen

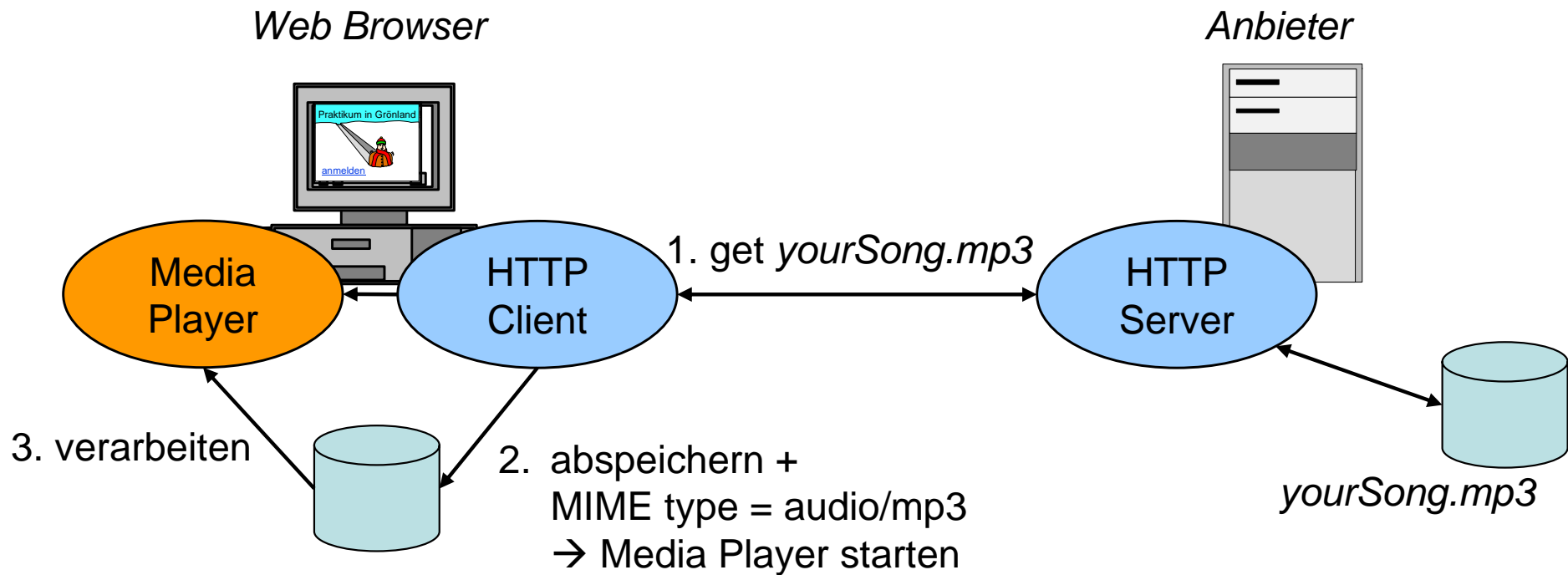
- > Audio
- > Video
- > Internet Telephony
- > Tele Learning
- > ...

> Neue Anforderungen

- > Echtzeit
- > Synchronisation
- > Multicast („Fernsehen“ per Internet)
- > ...

Dienst	Bitrate
Audio MP3	100-200 Kbit/s
MPEG Video	2 Mbit/s
Standard Video	140 Mbit/s
HDTV (1080i50/MPEG-2)	27 Mbit/s

Browser und Multimedia



Dienstgüte – Quality of Service (QoS)

- > Dienstgüte-Management (QoS Management) in Netzen
 - > Alle Verfahren, die den Datenfluss so beeinflussen, dass der Dienst mit einer festgelegten Qualität beim Empfänger ankommt
- > Allgemein: Dienstgüte in verteilten Systemen
 - > Nicht-funktionale Eigenschaften eines Dienstangebots
- > Anforderungen
 - > Dienstgüte wird vom Empfänger bzw. Kunden festgestellt
 - > Dienstgüte erfordert Ende-zu-Ende-Sicht
 - > Dienstgüte auf der Anwendungsschicht betrifft alle darunter liegenden Schichten

Dienstgüte – Quality of Service (QoS)

- > Beispiele für Diensteigenschaften aus Benutzersicht
 - > Wiedergabequalität
 - > Antwortzeit
 - > Zuverlässigkeit
 - > Verschlüsselung
 - > ...
- > Anforderungen eines Dienstes an das Netzwerk
 - > Fehlerrate
 - > Verzögerung
 - > Jitter (Schwankungen)
 - > Bandbreite
 - > Synchronisation (z.B. „Lippen-Synchronisation“ bei Videokonferenz)

Vor allem Multimedia-Anwendungen verlangen nach
QoS-Zusicherungen.

Multimedia – Anforderungen

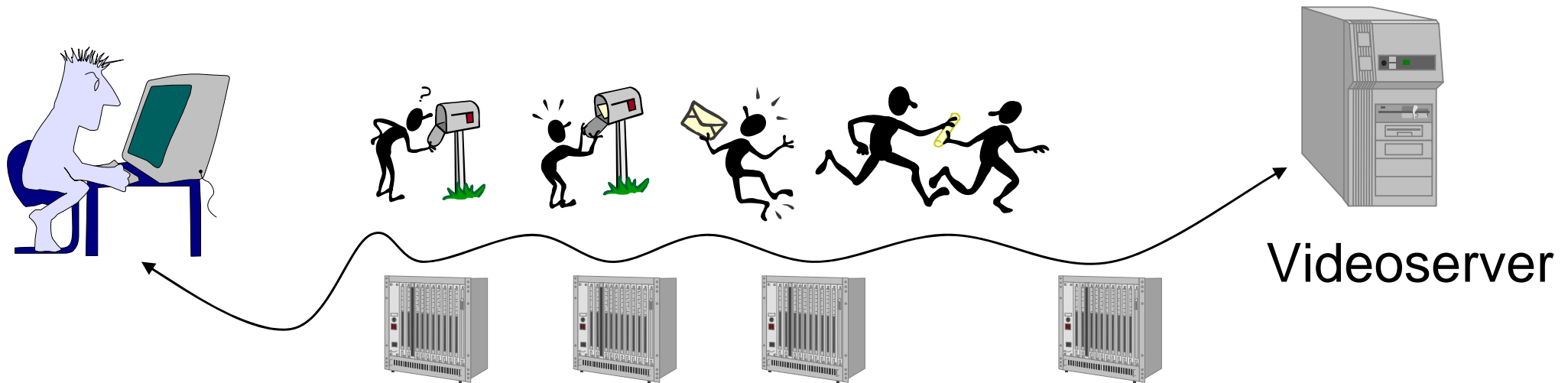
- > Kontinuierliche Daten (Streams, Flows)
 - > Zeitliche Abhängigkeiten innerhalb der Daten
- > Multicast-Übertragung
 - > Eine Quelle – mehrere Abnehmer
- > Typische Dienstgüte-Parameter
 - > Bandbreite
 - > Verzögerung: Ende-zu-Ende Verzögerung
 - > Jitter: Varianz innerhalb eines Datenstroms

Das Internet kann mit „best-effort“ keine Garantien geben.

Anwendungen und QoS-Parameter

Anwendung	Fehlerrate	Verzögerung	Schwankung	Bandbreite
E-Mail	Hoch	Niedrig	Niedrig	Niedrig
Dateitransfer	Hoch	Niedrig	Niedrig	Mittel
Web-Zugriff	Hoch	Mittel	Niedrig	Mittel
Remote Login	Hoch	Mittel	Mittel	Niedrig
Audio on demand	Niedrig	Niedrig	Hoch	Mittel
Video on demand	Niedrig	Niedrig	Hoch	Hoch
Telephonie	Niedrig	Hoch	Hoch	Niedrig
Video-Konferenz	Niedrig	Hoch	Hoch	Hoch

Datenübertragung im Internet



- > Datenströme werden paketweise übertragen
- > Verbindungsloses IP → nach jedem „Hop“ neue Wegentscheidung
- > Keine Reservierungen → nicht-deterministisches Übertragungsverhalten

Reservieren oder nicht – das ist hier die Frage!

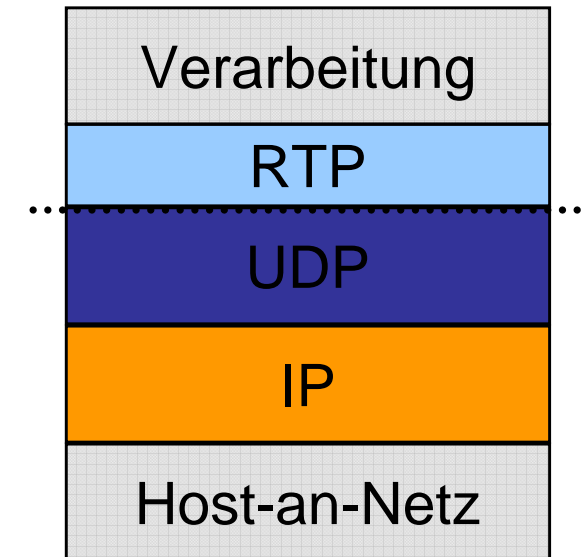
QoS im Internet



1. Ansatz: „Wir benutzen das, was schon da ist...“

Real-Time Transport Protocol (RTP)

- > Basiert (üblicherweise) auf UDP
 - > Router sehen keine „RTP-Pakete“
 - > Nutzung von Unicast oder Multicast
 - > Multiplexen mehrerer Datenströme
- > Sehr einfaches Protokoll
 - > Beschrieben in RFC 3550
 - > Sequentielle Nummerierung der Pakete
 - > Zeitstempel zur Abstandsregelung der Pakete
 - > Keine Flusssteuerung, Fehlerbehandlung, Bestätigungen, Wiederholungsanforderungen
 - > Keine Reservierungen oder QoS-Garantien
- > Steuerung und Feedback durch *RTCP*
(*Real-Time Transport Control Protocol*)



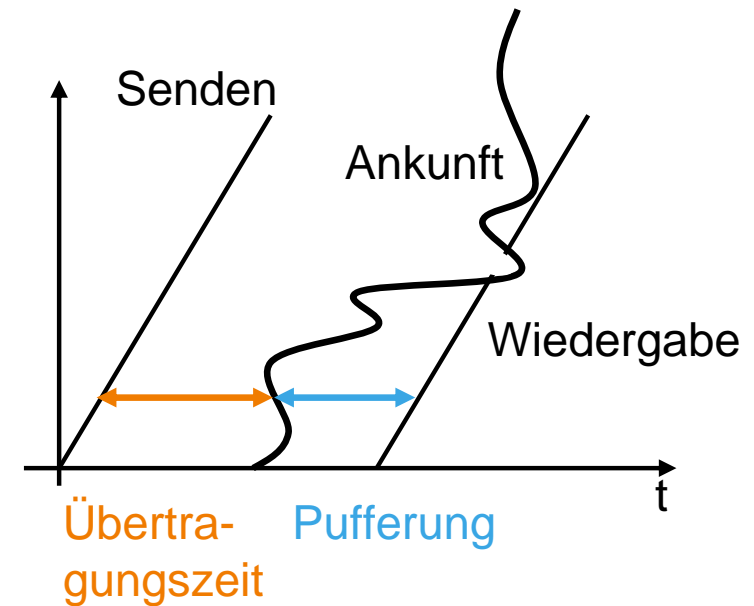
Einige RTP Audio-Nutzdatentypen

Typnummer	Format	Abtastrate	Durchsatz
0	PCM	8 KHz	64 Kbps
1	1016	8 KHz	4,8 Kbps
3	GSM	8 KHz	13 Kbps
7	LPC	8 KHz	2,4 Kbps
14	MP3	90 KHz	96-128 Kbps

2. Ansatz: „Wir erweitern die Protokolle und Dienste ...“

Allgemeine QoS-Techniken

- > Überdimensionierung
- > Pufferung beim Empfänger
 - > Gleicht Schwankungen aus
 - > Üblich sind ca. 10 sec Pufferung
- > Eingangssteuerung (Admission Control)
 - > „Türsteher“ entscheidet, ob Datenstrom akzeptiert wird
- > Verkehrsformung (Traffic Shaping)
 - > „Glätten“ des Datenstroms nach vorheriger Vereinbarung
- > Reservierung von Ressourcen



Datenstromspezifikation (Flow Spec)

- > Beschreibt Charakteristik und Anforderungen des Datenstroms
- > Grundlage des „QoS Vertrags“ (Service Level Agreement, SLA)
- > Beispiel

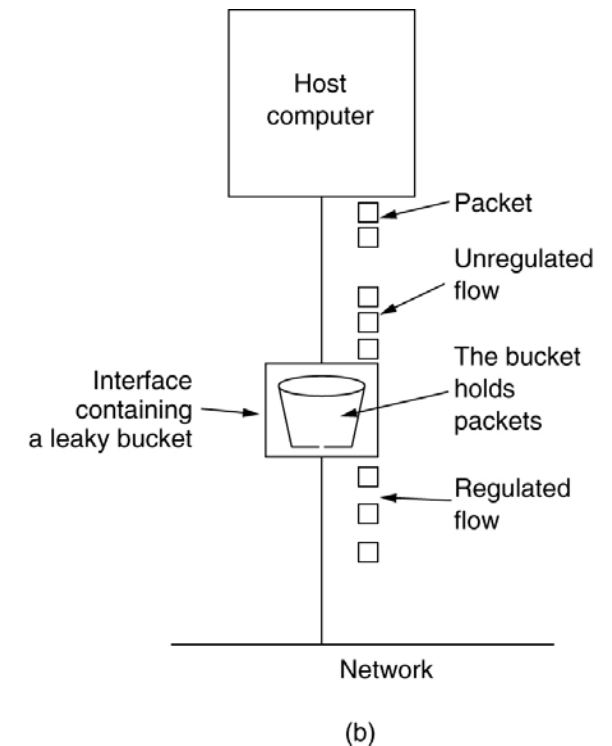
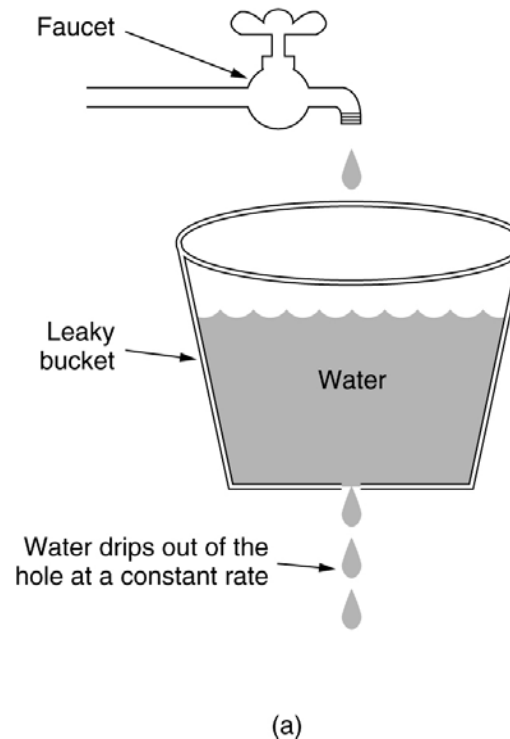
Parameter	Maßeinheit
Mittlere Datenrate	Bytes / sec
Maximale Datenrate	Bytes / sec
Maximale Verzögerung	sec
Minimale Paketlänge	Bytes
Maximale Paketlänge	Bytes

Verkehrsformung (Traffic Shaping)

- > Eine vorrangige Ursache für Überlastsituationen sind **Verkehrspitzen**
- > Diese entstehen beispielsweise, wenn Rechner mit stark ungleichmäßig Rate senden
- > Ziel
 - > Beeinflussung sendender Rechner, so dass sie gleichmäßiger senden und damit weniger Überlastsituationen auftreten
- > Vorgehen
 - > Beim Sender werden Algorithmen ausgeführt, die den Verkehr formen

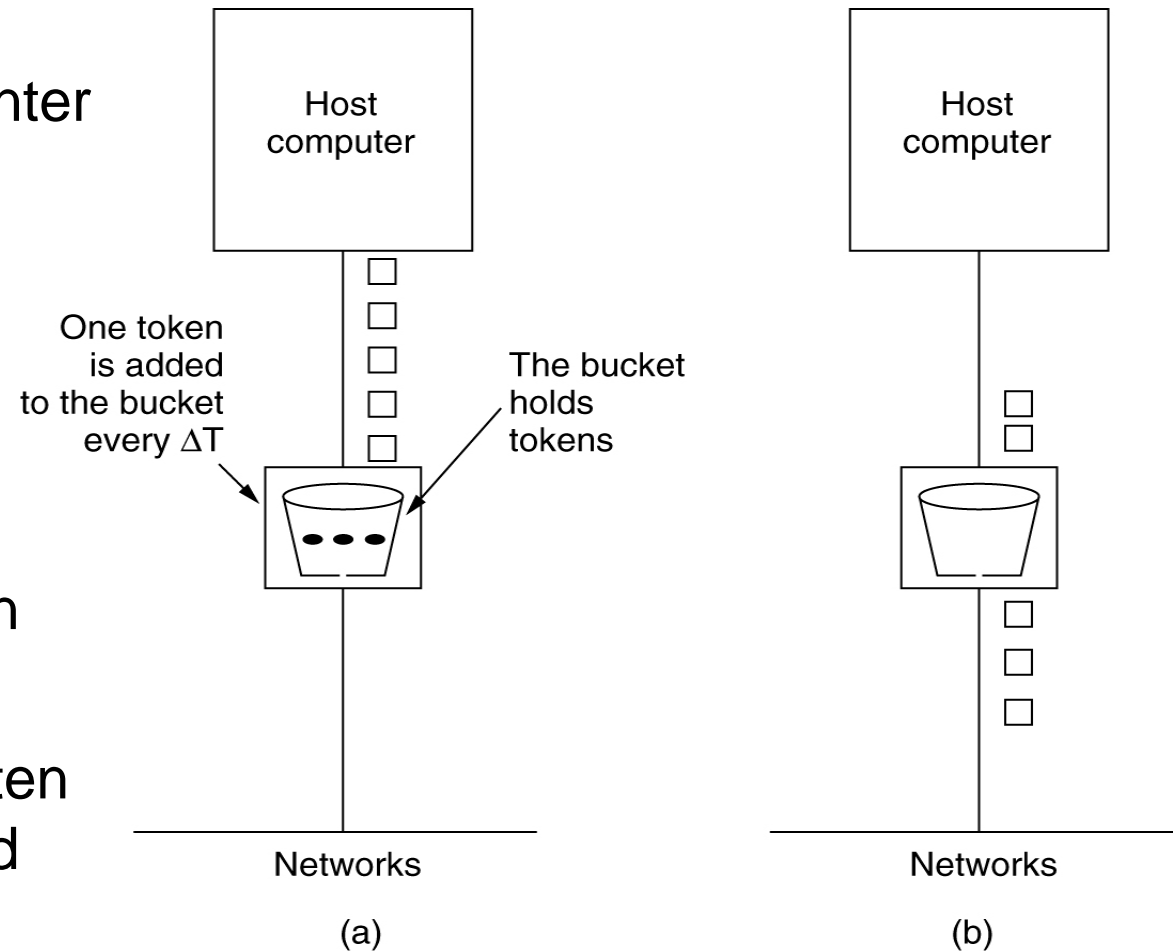
„Leaky Bucket“

- > Begrenzt **maximale** Senderate
- > Gesendete Pakete kommen in den Eimer
- > Die Pakete verlassen den Eimer mit einer konstanten Rate
- > Eimer hat eine gewisse Kapazität
- > Bei vollem Eimer werden Pakete verworfen
- > Implementierung des Eimers als Warteschlange



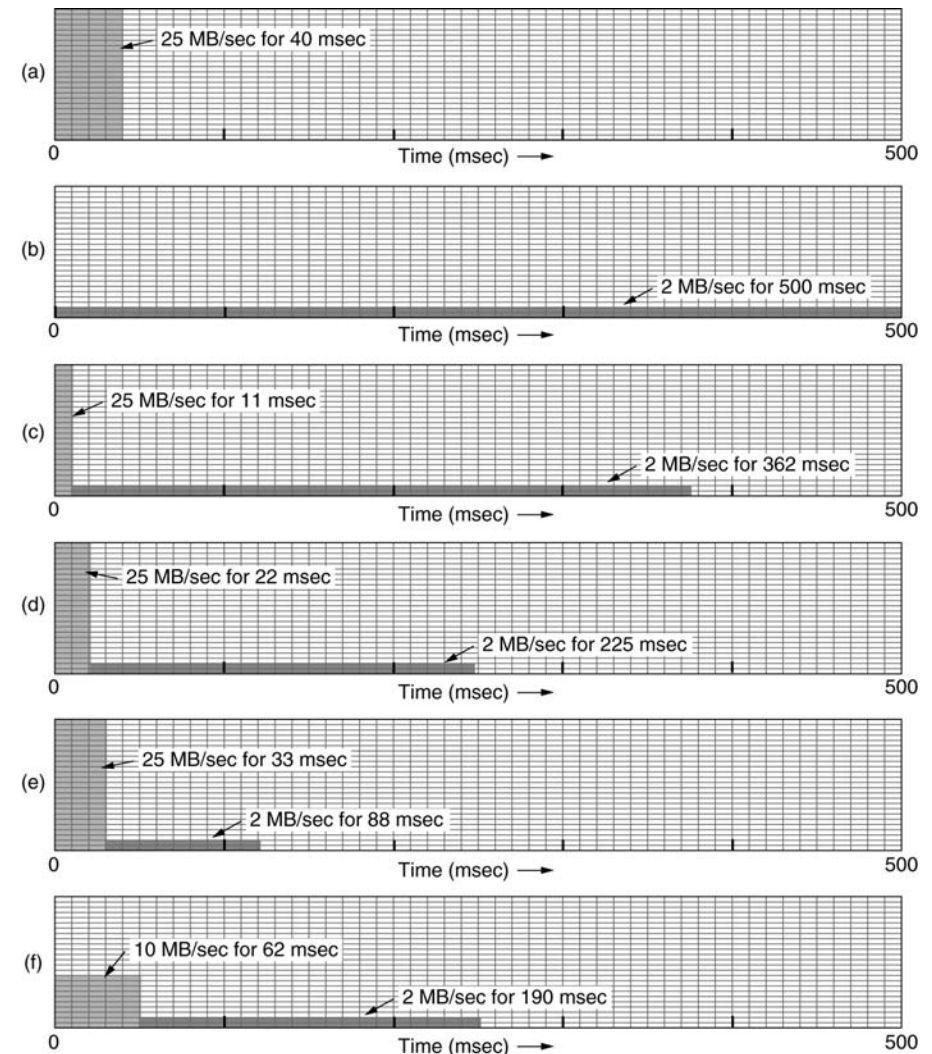
„Token Bucket“

- > Begrenzt **durchschnittliche** Senderate
- > **Tokens** werden mit konstanter Rate in den Eimer gelegt
- > Bis zu **n** Tokens können gesammelt werden
- > Jede Nachricht nimmt ein Token
- > Bursts bis zu **n** Nachrichten gehen durch
- > Es werden keine Nachrichten verworfen, stattdessen wird die Applikation beim Senden blockiert



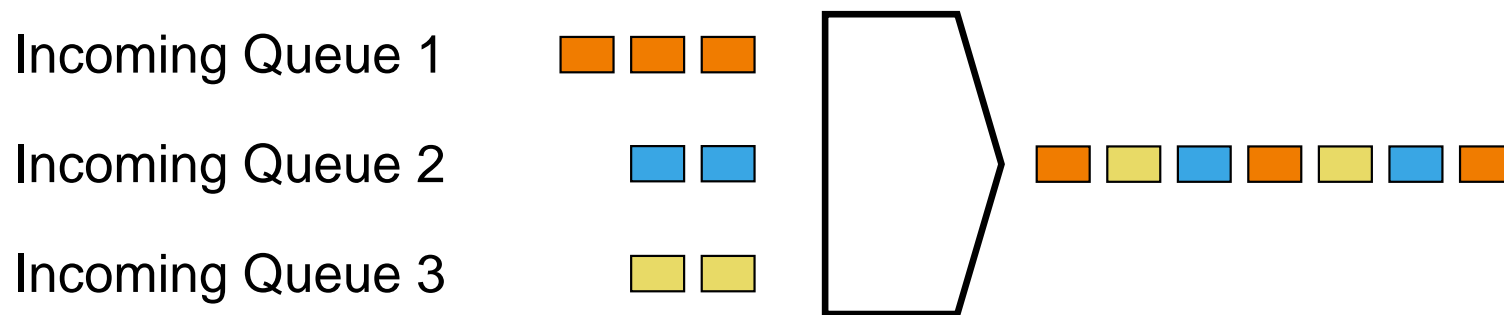
Verkehrsformung

- > Ohne Verkehrsformung (a)
- > 2 MB/sec Leaky Bucket (b)
- > 2 MB/sec Token Bucket
 - > mit Kapazität von 250 KB (c)
 - > mit Kapazität von 500 KB (d)
 - > mit Kapazität von 750 KB (e)
- > 500 KB Token Bucket mit nachfolgendem 10 MB/sec Leaky Bucket (f)



Weiterleitungsstrategien

- > Nachrichten/Pakete werden von Zwischenknoten im Netz weiter in Richtung ihres Ziels geleitet → **Routing**
- > Behandlung der Nachrichten durch Router erfolgt oft nach
 - > **First Come First Served (FCFS)** Strategie bzw. synonym
 - > **First In First Out (FIFO)** Strategie
- > Mehrere Eingangswarteschlangen werden üblicherweise nach der **Round Robin** Strategie bedient

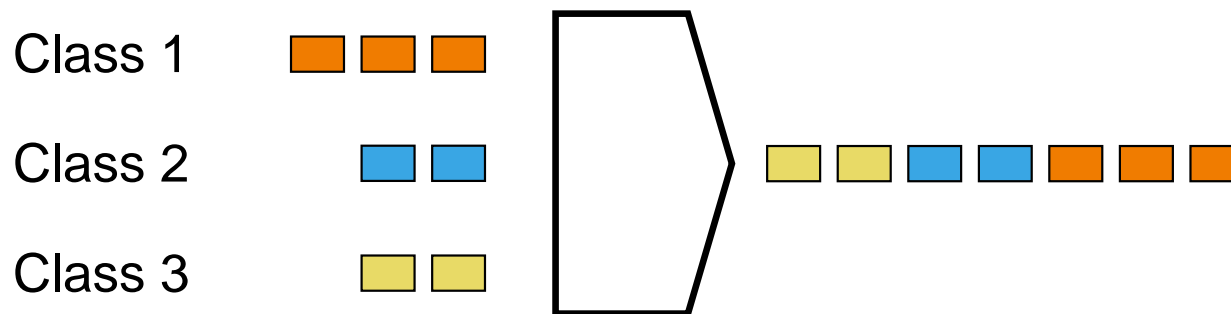


Weiterleitungsstrategien mit Prioritäten

- > Nachrichten können auch Prioritäten zugeordnet werden
- > Nachrichten höherer Priorität werden bevorzugt behandelt
- > Beispiele
 - > Highest Priority First
 - > Weighted Fair Queuing

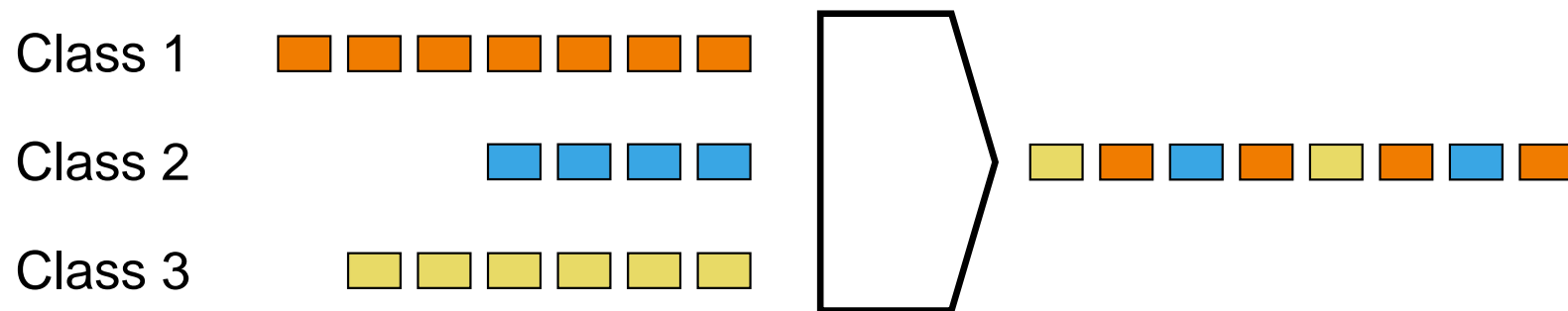
Highest Priority First

- > Separate Warteschlangen für jede Nachrichtenklasse
- > Von allen wartenden Nachrichten wird zunächst die mit der höchsten Priorität weitergeleitet
- > Hochpriore Nachrichten erfahren minimale Wartezeit
- > Niederpriore Nachrichten können hingegen beliebig lange verzögert werden → Verhungern (Starving)
- > Beispiel
 - > Klasse 1 mit hoher, Klasse 2 mit mittlerer und Klasse 3 mit geringer Priorität



Weighted Fair Queuing

- > Erweitert Round Robin um Prioritäten
- > Niederpriore Nachrichten können nicht verhungern
- > Hochpriore Nachrichten können verzögert werden, jedoch nicht beliebig lange
- > Beispiel
 - > Klasse 1 mit Gewicht 2, Klassen 2 und 3 mit Gewicht 1



QoS-Erweiterungen für das Internet

Integrates Services (**IntServ**)

- > Annahme
 - > Ressourcen sind knapp
- > Folgerung
 - > **Reservieren** der Ressourcen
 - > „fein-granular“, „flow-based“

Differentiated Services (**DiffServ**)

- > Annahme
 - > Es gibt genug Ressourcen
- > Folgerung
 - > Ressourcen **verteilen**
 - > „grob-granular“, „class-based“

Integrated Services (IntServ)

- > Philosophie
 - > Ressourcen sind knapp
 - > QoS-Anforderungen nur durch Reservierung zu erfüllen
- > Beispiel: zwei Dienstklassen
 - > Guaranteed Service (GS)
 - > Controlled Load (CL)
 - > Anderer Verkehr: Best-Effort
- > Flow-based
 - > Jeder(!) Knoten zwischen Sender und Empfänger muss für einen „Flow“ eine Reservierung vornehmen
- > **Resource reSerVation Protocol (RSVP)** zwischen Sender und Empfänger [RFC 2205, 1997]

Guaranteed Service vs. Controlled Load

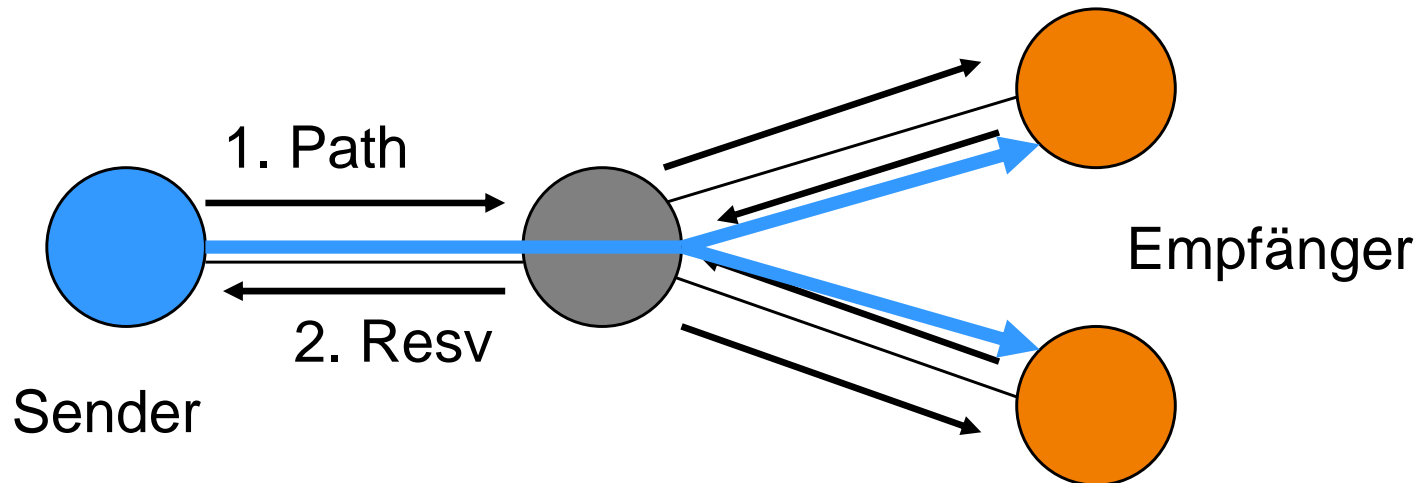
- > Guaranteed Service (GS) [RFC 2212]
 - > Netz sichert eine Bandbreite zu
 - > Wenn sich die Anwendung an diese Bandbreite hält, gibt es keinen Verlust durch Stauungen
 - > Dem Datenverkehr innerhalb der Bandbreite wird eine obere Schranke für die Transportverzögerung zugesichert

- > Controlled Load (CL) [RFC 2211]
 - > Netz emuliert in etwa die Dienstgüte, die ein IP-Netz unter geringer Auslastung erbringt
 - > Der Verlust aufgrund von Stauungen im Netz ist sehr gering
 - > Ein hoher Prozentsatz der IP-Pakete bleibt unter der maximalen Übertragungsverzögerung

Resource Reservation Protocol (RSVP)

- > Wir betrachten „Sender“ und „Empfänger“ eines Flows
 - > RSVP verfolgt einen empfängerorientierten Ansatz
 - > Reservierungszustand gilt nur für begrenzte Zeit und muss daher periodisch erneuert werden
- 1. Sender schickt „Path-Nachricht“ an den Empfänger
 - > Übermittelt Flow Spec
 - > Ermittelt den Weg des Flows durch das Netz
- 2. Empfänger antwortet mit „Reserve-Nachricht“
 - > Jeder Router auf dem Weg reserviert die notwendigen Ressourcen (sofern er kann, sonst Fehler!)
 - > Solange der Empfänger die Reservierung aufrechterhalten will, schickt er z.B. alle 30 sec eine weitere Reserve-Nachricht

Empfängerbasierte Reservierung



1. Sender „beantragt“ Pfad
2. Empfänger reserviert entlang des Pfads entsprechend seinem Ressourcenbedarf
3. a) Übertragung
b) Periodische Auffrischung der Reservierung

Diskussion – Integrated Services

- > Eigenschaften
 - > Bietet Garantien für die Übertragung
 - > Unterstützt Uni- und Multicast
 - > Empfänger fordern nur benötigte Ressourcen an
 - > Periodische Auffrischung verhindert unnötige Ressourcenblockaden

- > Aber ...
 - > Skalierbarkeit: Für jeden http-Request eine Reservierung???
 - > Sicherheit: Ressourcendiebstahl???
 - > Verwaltung: Jeder Flow ein Eintrag in jedem Knoten???

→ IntServ eher für lokale Netze geeignet !?

Differentiated Services (DiffServ)

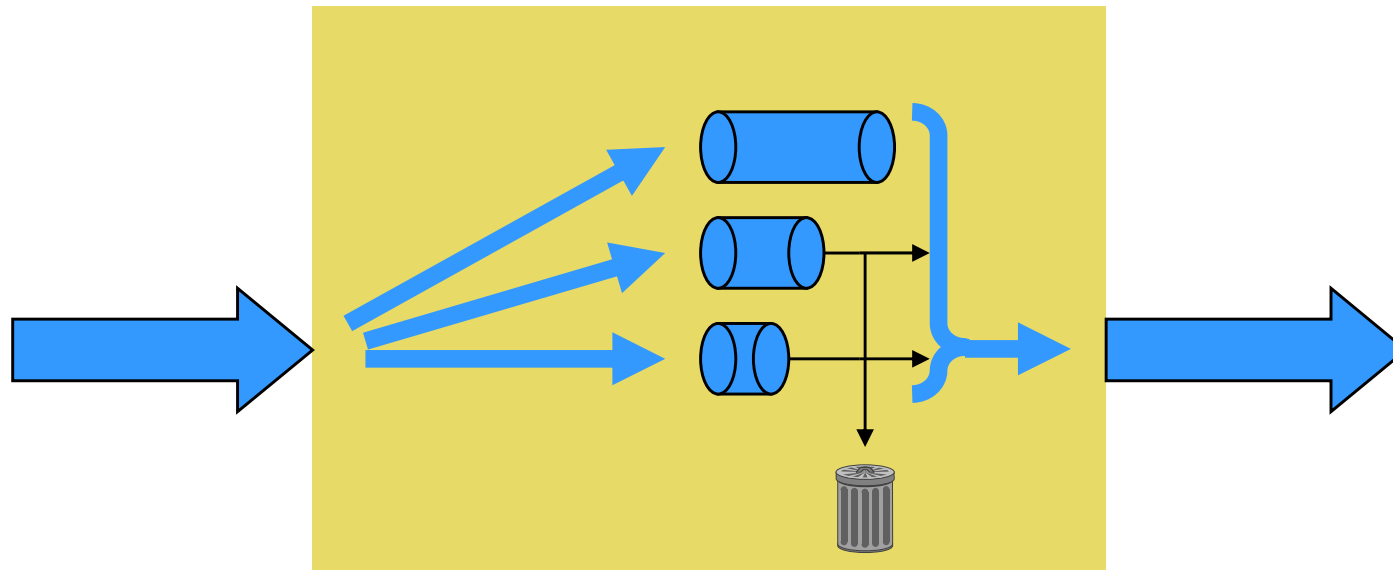
- > Philosophie
 - > Ressourcen sind in ausreichendem Maße verfügbar
 - > Zuteilung an unterschiedliche Anwendungen ist zu lösen
 - > Minimierung von Verwaltungsaufwand

- > Lösung: „Class-based“
 - > Aggregation von Flows
 - > Zusammenfassen von Flows derselben (Prioritäts-) Klasse
 - > „Reservierung“ über die Lebensdauer einzelner Flows hinaus

DiffServ – Klassifikation von Paketen

- > DiffServ stellt Dienstklassen bereit, z.B.
 - > **Expedited Forwarding**: Steuere den Verkehr an allen Knoten im Netz so, dass „expedited Verkehr“ keine Verzögerung erfährt
 - > **Assured Service**: Verschiedene Verkehrsklassen mit unterschiedlicher „Drop Precedence“
- > Jedes Paket ist einer Dienstklasse zugeordnet
 - > IPv4 benutzt Type of Service (TOS) Octet
 - > IPv6 benutzt Traffic Class Octet
- > Jeder Router bewertet Paket anhand
 - > Dienstklasse
 - > Absender/Empfänger-Adresse
 - > Evtl. zusätzlichen Informationen (Hop Count ...)

2-Bit DiffServ Architektur [RFC 2638]



> Drei Dienstklassen

- > Premium (P-Bit): direkte Weiterleitung
- > Assured (A-Bit): Puffern, Paketverlust mit geringer Wahrscheinlichkeit
- > Best-Effort: Puffern, Paketverlust mit höherer Wahrscheinlichkeit

Diskussion – Differentiated Services

- > Bietet keine Garantien
- > Bevorzugung von Paketen aufgrund ihrer Klassifikation
- > Statische Kontrolle im Netz
 - > Feste Reservierung für Dienstklassen (Flow-Aggregation)
 - > keine dynamische Anpassung
- > Geringer Verwaltungsaufwand
- > Skalierbarkeit

→ DiffServ geeignet für Weitverkehrsnetze (WAN) !?

Koexistenz von IntServ und DiffServ

IntServ

- > Bietet Garantien für einzelne Flows
- > Verwaltet beschränkte Ressourcen
- > Skaliert schlecht, da Zustand per Flow

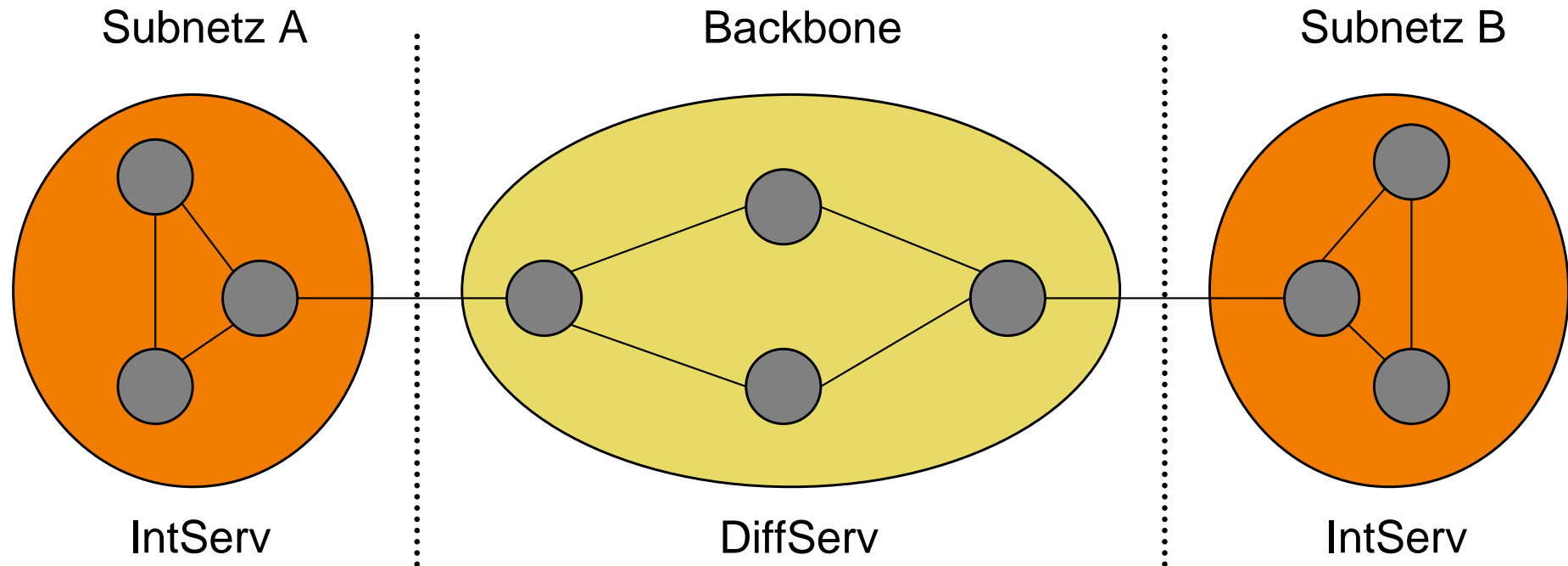
DiffServ

- > Bietet Dienstklassen
- > Teilt ausreichende Ressourcen zu
- > Skaliert, da statische Konfiguration und einfache Bearbeitung

Kombination ?

- > IntServ in den Endnetzen
- > DiffServ in den Backbones
- > RFC 2998

QoS im Internet



- > Endsysteme
 - > Reservieren
 - > Garantien

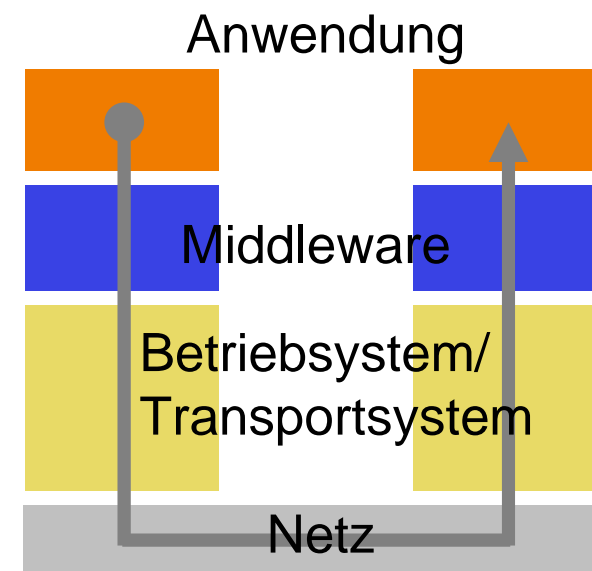
- > Weitverkehr
 - > Dienstklassen zuteilen
 - > Einfache - schnelle! - Bearbeitung

Zusammenfassung

- > QoS im Internet erfordert neue Protokolle
 - > Neue Hardware, neue Software
- > Verschiedene Strategien möglich
 - > Beschränkte Ressourcen → Reservieren
 - > Genügend Ressourcen → Verwalten
- > IntServ und DiffServ stellen passende Lösungen dar
- > Auch eine Kombination ist sinnvoll

Ausblick

- > QoS ist mehr als Bandbreite und Verzögerung
 - > Echtzeit
 - > Fehlertoleranz
 - > Sicherheit
 - > Datenintegrität
 - > Präzision
 - >



Auch die Middleware oberhalb von Betriebssystem und Datentransport muss QoS unterstützen!

Fragen?



Literatur

- > B. Carpenter, D. Kandlur, *Diversifying Internet Delivery*, IEEE Spectrum, November 1999, S. 57-61
- > Y. Bernet, *The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network*, IEEE Communications Magazine, Februar 2000, S. 154-162
- > D. Verma, *Supporting Service Level Agreements on IP Networks*, Macmillan Technical Publishing, ISBN 1578701465
- > Internet 2, <http://qbone.internet2.edu/>
- > IETF Differentiated Services (DiffServ) WG, <http://www.ietf.org/html.charters/OLD/diffserv-charter.html>
- > IETF Integrated Services over Specific Link Layers (issl) WG, <http://www.ietf.org/html.charters/OLD/issl-charter.html>