

# Dienstgüte

# Datenübertragung im Internet

- Datenfluss = Strom von Paketen von einer Quelle zu einem Ziel
  - IP verbindungslos
    - nach jedem „Hop“ neue Wegentscheidung
    - Folgapaket kann viel längere oder kürzere Zeit zum Transport beanspruchen
- nichtvorhersehbares Übertragungsverhalten



# Dienstqualität

- alles, was nicht funktionell beschrieben ist (durch die Schnittstelle)
- Beispiele für Merkmale
  - Fehlerrate: Anzahl der Übertragungsfehler pro Zeiteinheit
  - Verzögerung: Zeitdifferenz zwischen dem Abschicken und dem (endgültigen) Ankommen beim Empfänger (d.h. Präsentation)
  - Schwankung: Zeitweise Veränderung der Datenrate
  - Bandbreite (eigentlich Datenübertragungsrate): Maximale Datenrate, die der Übertragungskanal ermöglicht.

# Dienstgüte (Quality of Service)

- Dienstgüte-Management (QoS-Management) in Netzen
  - alle Verfahren, die den Datenfluss so beeinflussen, dass der Dienst mit einer festgelegten Qualität beim Empfänger ankommt
  - 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

# Datenflussspezifikation

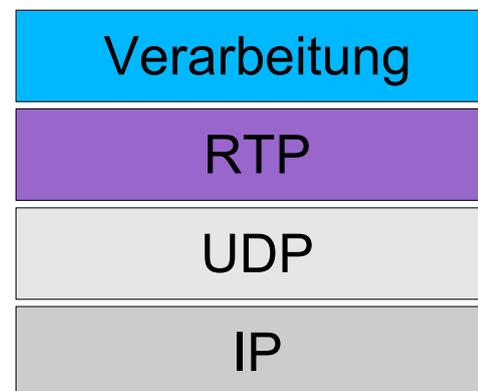
- Datenflussspezifikation
  - auch Flow Spec genannt
  - beschreibt Charakteristik und Anforderungen des Datenstroms
  - Grundlage des „QoS-Vertrags“ (Service Level Agreement, SLA)

Beispiel

Parameter	Maßeinheit
Mittlere Datenrate	Byte/sek
Maximale Datenrate	Byte/sek
Maximale Verzögerung	sek
Minimale Paketlänge	Byte
Maximale Paketlänge	Byte

# 1. Ansatz: Spezielle Protokolle

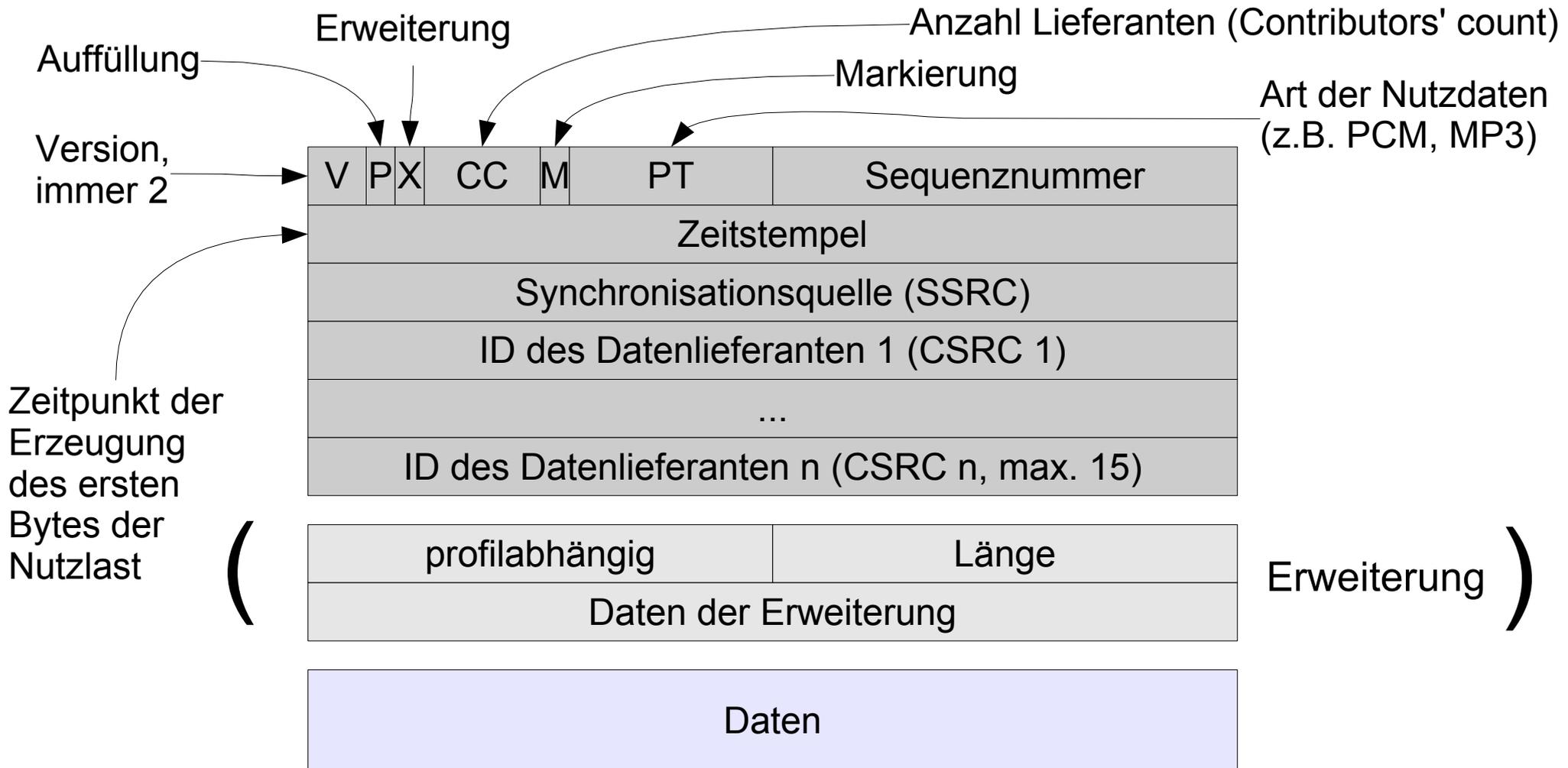
- **Real-Time Transport Protocol [RFC 1889, 3550]**
  - setzt (üblicherweise) auf UDP auf
  - Router sehen keine „RTP-Pakete“
  - Multiplexen mehrerer Datenströme auf UDP (Unicast oder Multicast)
  - Kombinieren („Mischen“) von verschiedenen Strömen (Datenlieferanten, contributors) zu einem neuen Strom mit neuen Qualitäten
  - Einsatz insbesondere bei H.323 und SIP (Internet-Telefonie)



# RTP – RTSP – RTCP

- Datenübertragung: RTP
  - sequentielle Nummerierung der RTP-Pakete
  - Zeitstempel zur Abstandsregelung der Pakete
  - keine Flusssteuerung, Fehlerbehandlung, Bestätigungen, Wiederholungsanforderungen
- Verbindungssteuerung: RTSP (Realtime Streaming Protocol)
  - basiert auf HTTP
  - keine Nutzdatenübertragung
  - Sitzungsverwaltung
- QoS-Kontrolle: RTCP (Realtime Control Protocol)
  - Aushandlung von QoS-Parametern
  - Prüfung der Einhaltung

# RTP-Protokolleinheit



\*original: Synchronization Source, identifiziert diesen Strom

# RTP

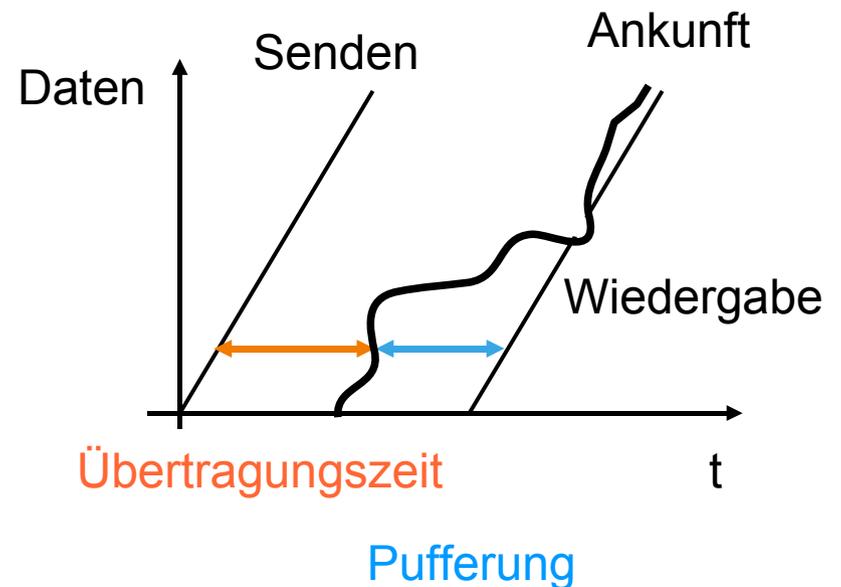
- Einige RTP-Nutzdatentypen

<b>Typ</b>	<b>Format</b>	<b>Abtastrate</b>	<b>Durchsatz</b>
0	PCM	8 kHz	64 kBit/s
1	1016	8 kHz	4,8 kBit/s
3	GSM	8 kHz	13 kBit/s
7	LPC	8 kHz	2,4 kBit/s
14	MP3	90 kHz	96-128 kBit/s

- Profile
  - können spezifische Erweiterungen des Protokolls vereinbaren

## 2. Ansatz: Nutzen der existierenden Infrastruktur

- Großzügige Auslegung der Infrastruktur
  - Für ausreichende Kapazitäten sorgen
- Puffern der eintreffenden Daten vor der Weiterverarbeitung
  - Empfängerseite glättet ungleichmäßige Übertragung
  - erhöht die Verzögerung
- Datenstromformung (Traffic shaping)
  - Sender glättet den abgehenden Datenstrom („Leaky Bucket“, „Token Bucket“)
  - Sender kann dem Netzbetreiber eine Zusicherung geben, mit welchen Datenraten zu rechnen ist
  - Netzbetreiber sichert seinen Dienst auf dieser Basis zu

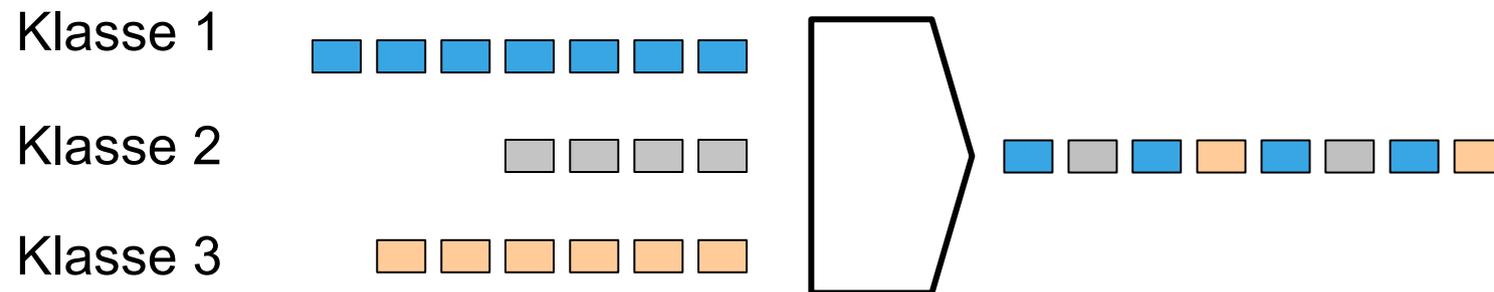


## 3. Ansatz: Ressourcenmanagement

- Ressourcenreservierung
  - CPU-Zeit: Wird die Rechenkapazität der Router überschritten?
  - Bandbreite: Eine 2Mbit/s-Leitung kann keine drei Ströme mit 1Mbit/s übertragen
  - Puffer: Pakete müssen gepuffert werden, bis die Anwendung sie abholt
- Zulassung
  - Jeder Router muss Zusage geben können, ob er die Dienstgüte für jeden Strom aufrecht erhalten kann.
  - Router muss anhand der Flussspezifikation entscheiden, ob er den neuen Fluss akzeptiert oder ablehnt

# Scheduling der Pakete im Router

- Weighted Fair Queueing
  - Jeder Datenflusstyp erhält eigene Warteschlange
  - Round Robin mit Bevorzugung (=Gewichtung)
  - Beispiel: Klasse 1 hat Gewicht „2“, die anderen beiden Gewicht „1“



# Zuteilung oder Reservierung?

- Zwei Philosophien für die Zuteilung von Ressourcen
  - „**fein-granular**“, „**flow-based**“: Ressourcen sind knapp → muss man reservieren
  - „**grob-granular**“, „**class-based**“: Es gibt genug Ressourcen → kann man verteilen
- **Integrated Services (IntServ)**
  - Annahme: Ressourcen sind knapp, QoS-Anforderungen nur durch Reservierung zu erfüllen
  - Reservierung über Dienstklassen
- **Differentiated Services (DiffServ)**
  - Annahme: Ressourcen sind vorhanden; Zuteilung muss geregelt werden
  - Weiterleitung über Dienstklassen

# Dienstklassen der Integrated Services

- 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
  - „Garantierte Lieferung“ (kein Paketverlust)
  - „Garantierte Lieferfrist“ (ggf. früher als erwartet → Puffern beim Empfänger!)

# Dienstklassen der Integrated Services

- Controlled Load (CL) [RFC 2211]
  - Netz emuliert in etwa die Dienstgüte, die ein (Best-effort-) IP-Netz unter geringer Auslastung erbringt
  - Netz sorgt für Reservierungen, sodass sich das Netz immer gleich verhält
  - Verhältnismäßig (beabsichtigt) einfacher Dienst
- Ergebnis
  - Der Verlust aufgrund von Stauungen im Netz ist sehr gering
  - Ein hoher Prozentsatz der IP-Pakete bleibt unter der maximalen Übertragungsverzögerung
- Günstig für Anwendungen, die Probleme bei hohem Netzverkehr bekommen

# Integrated Services: Konzept

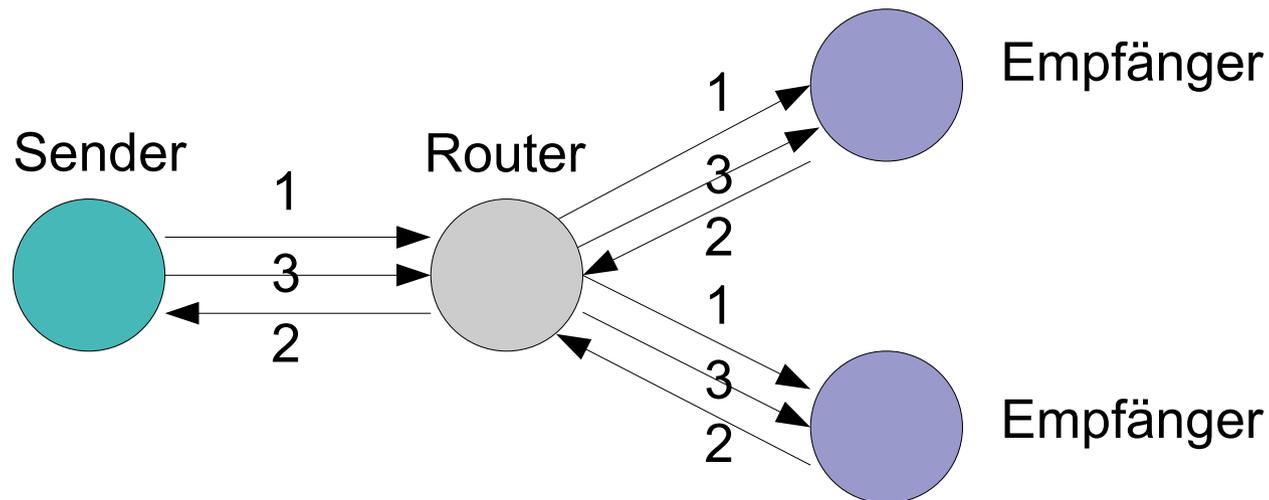
- Datenfluss-basiert (Flow-based)
  - Jeder(!) Knoten zwischen Sender und Empfänger muss für einen Datenfluss eine Reservierung vornehmen
- Resource Reservation Protocol (RSVP) zwischen Sender und Empfänger [RFC 2205, 1997]
  - RSVP verfolgt einen empfängerorientierten Ansatz: Der Empfänger muss die Dienstgüte beantragen
  - Reservierungszustand gilt nur für begrenzte Zeit und muss daher periodisch erneuert werden
  - Unterstützt IPv4 und IPv6

# RSVP-Grundkonzepte

- Sender schickt „Path“-Nachricht an den Empfänger
  - übermittelt Datenflussspezifikation
  - ermittelt den Weg des Datenflusses durch das Netz
- Empfänger antwortet mit „Reserve“-Nachricht
  - Jeder Router auf dem Weg reserviert die notwendigen Ressourcen (sofern er kann, sonst Fehler!)
  - Solange der Empfänger die Reservierung aufrecht erhalten will, schickt er z.B. alle 30 sek. eine weitere Reserve-Nachricht

# RSVP

- Empfängerbasierte Reservierung
  1. Sender „beantragt“ Pfad (legt damit Pfad fest, entlang dem die Reservierung stattfinden muss)
  2. Empfänger reserviert entlang des Pfads entsprechend seinem Ressourcenbedarf
  3. a) Übertragung  
b) Periodische Auffrischung der Reservierung (beide Richtungen)



# RSVP-Nachrichten

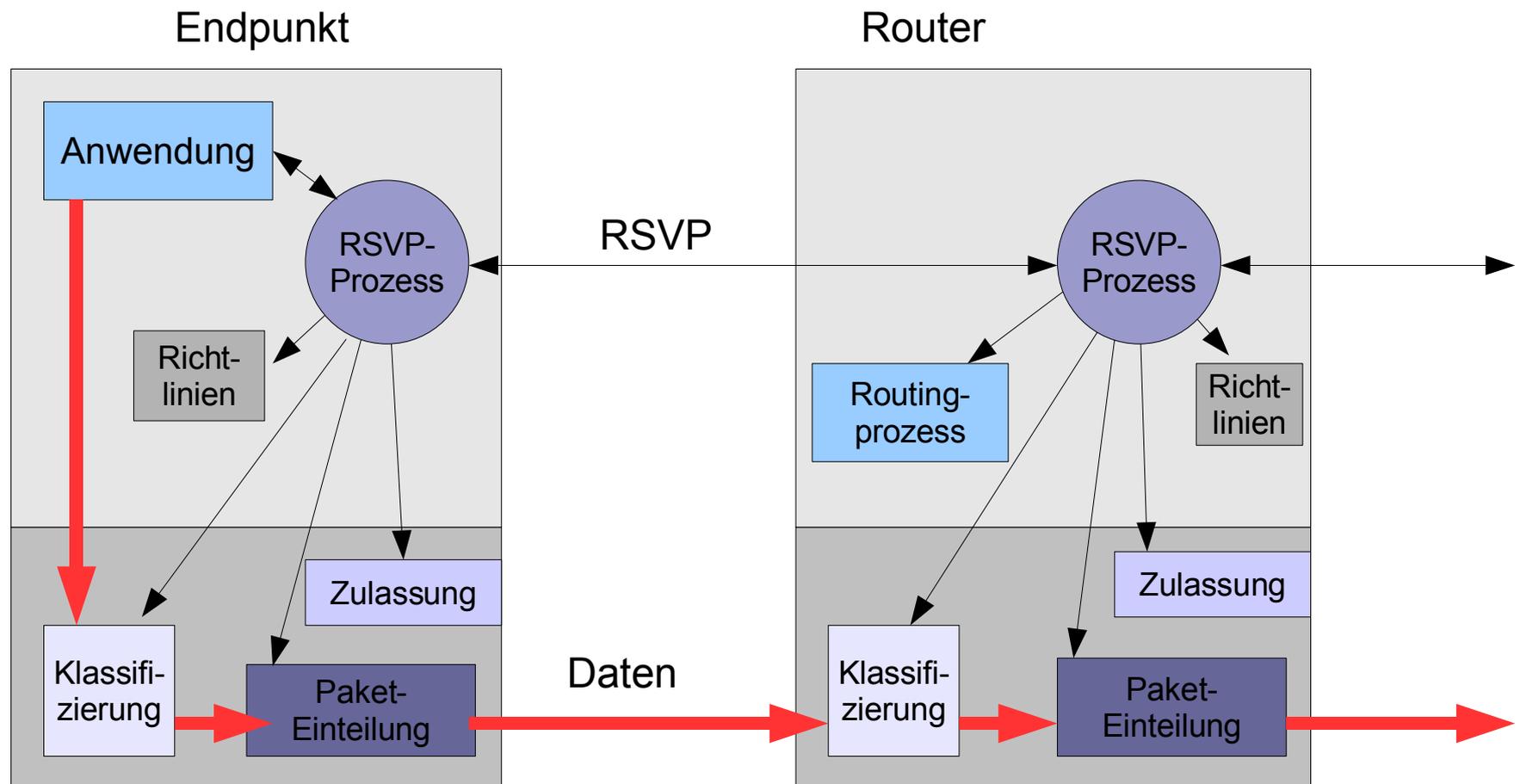
- Pfadnachricht (Path message)
  - Sender → Empfänger; Charakterisierung der Daten pro Sender
- Reservierungsnachricht (Reservation message)
  - Empfänger → Sender; Reservierung der Ressourcen auf dem Weg
- Fehleranzeige bei Pfadnachricht (Path message error indication)
  - Fehlerrückmeldung für fehlgeschlagene Pfadnachricht

# RSVP-Nachrichten

- Fehleranzeige bei Reservierungsnachricht (Reservation message error indication)
  - Fehlerrückmeldung für fehlgeschlagene Reservierung
- Pfadabbaunachricht (Path teardown message)
  - Expliziter Abbau des Pfads
- Reservierungsabbaunachricht (Reservation teardown message)
  - Freigabe der reservierten Ressourcen

# RSVP-Architektur

- Nach RFC 2205



# RSVP-Architektur

- Aufbau der Reservierung
  - Richtlinien (Policy control): Darf der Anwender diese Reservierung beantragen?
  - Zulassung (Admission control): Kann dieser Knoten die Dienstgüte erbringen?
- Betrieb
  - Klassifizierung
    - Bestimmung der Dienstgütekategorie anhand der Pakete
    - Ggf. auch Bestimmung der Route
  - Paketeinteilung (Packet scheduler)
    - Bestimmung, welche Pakete wann weitergeleitet werden
    - Muss für die Einhaltung der Dienstgüte sorgen

# Diskussion: Integrated Services

- Vorteile
  - bietet Garantien für die Übertragung
  - unterstützt Uni- und Multicast
  - Empfänger fordern nur benötigte Ressourcen an; periodische Auffrischung verhindert unnötige Ressourcenblockaden
- Nachteile
  - Skalierung: Für jeden http-Request eine Reservierung
  - Sicherheit: Ressourcendiebstahl
  - Verwaltung: Für jeden Fluss ein Eintrag in jedem Knoten

→ **IntServ eher für lokale Netze geeignet**

# Differentiated Services (RFC 2474, 2475)

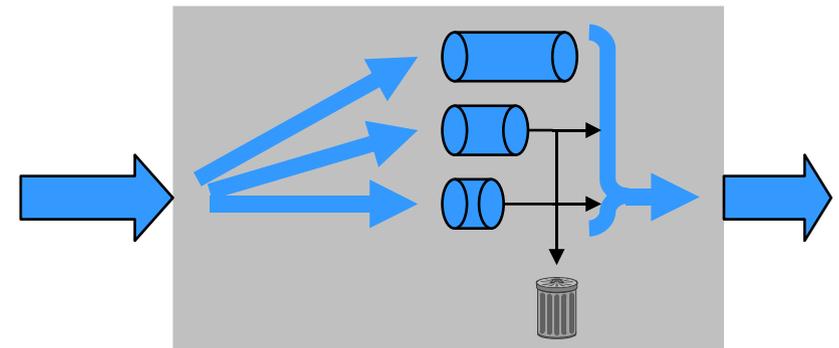
- Philosophie: Ressourcen sind in ausreichendem Maße verfügbar
  - Zuteilung an unterschiedliche Anwendungen ist zu lösen
  - Minimierung von Verwaltungsaufwand
- Lösung: Klassenbasierung („Class-based“)
  - Aggregation von Datenflüssen
  - Zusammenfassen von Flüssen derselben (Prioritäts-) Klasse
  - „Reservierung“ über die Lebensdauer einzelner Flüsse hinaus

# DiffServ: Klassifikation von Paketen

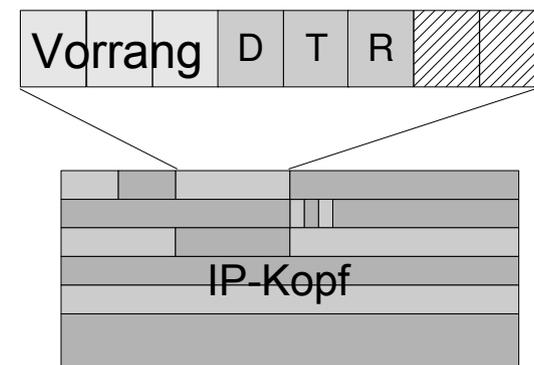
- DiffServ stellt Dienstklassen bereit, z.B.
  - beschleunigte Weiterleitung (**Expedited Forwarding**)
    - steuere den Verkehr an allen Knoten im Netz so, dass „beschleunigter Verkehr“ keine Verzögerung erfährt
  - Gesicherter Dienst (**Assured Service**)
    - Verkehrsklassen mit unterschiedlichem „Paketverwurfsrang“ („Drop Precedence“)
- Jedes Paket ist einer Dienstklasse zugeordnet
  - IPv4 - Type of Service (TOS)
  - IPv6 - Traffic Class
- Jeder Router bewertet Paket anhand
  - Dienstklasse
  - Absender/Empfänger-Adresse
  - Evtl. zusätzlichen Informationen (Hop Count ...)

# Eine 2-Bit-DiffServ-Architektur (RFC 2638)

- Einteilung von Bandbreite
- Drei Dienstklassen
  - **Premium:** Garantie für vereinbarte Spitzenlast; minimale Verzögerung (Dienst muss verfügbar sein, sobald er genutzt wird)
  - **Assured:** Gleiche Verzögerungen wie „Best effort“; Zusage der Weiterleitung von Pakete, wenn der Verkehr in einem vereinbarten Kapazitätsprofil bleibt; Spitzenlast kann zum Paketverlust führen
  - **Best-Effort:** Keine Zusagen bezüglich Paketbewahrung oder -verlust



- Dienstklassen werden im Vorrang-Teil des Diensttypfeldes gesetzt\*



\* „Wir überlassen die genaue Festlegung dem Standardisierungsprozess...“

# Eigenschaften von DiffServ

- 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 eher geeignet für Weitverkehrsnetze (WAN)**

# Koexistenz von IntServ und DiffServ

- IntServ
  - Bietet Garantien für einzelne Flüsse
  - Verwaltet beschränkte Ressourcen
  - Skaliert schlecht, da Zustand per Fluss
- DiffServ
  - Bietet Dienstklassen
  - Teilt ausreichende Ressourcen zu
  - Skaliert gut, da statische Konfiguration und einfache Bearbeitung

→ **IntServ in den Endnetzen – DiffServ in den Backbones [RFC 2998]**

# 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
- Multimedia-Anwendungen im Internet werden möglich

# 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://www.internet2.edu/qbone2>
- IETF Differentiated Services (DiffServ) WG, <http://www.ietf.org/html.charters/diffserv-charter.html>
- IETF Integrated Services over Specific Link Layers (issll) WG, <http://www.ietf.org/html.charters/issll-charter.html>