

Next Generation Internet

4. Dienstgüte

INSTITUT FÜR TELEMATIK



Überblick Kapitel 4

I. Einführung

1. Einführung

II. Internet-Architektur

2. Internet-Architektur
3. NAT & IPv6
4. Dienstgüte

- 4.1 Was ist Dienstgüte?
- 4.2 Wozu Dienstgüte im Internet?
- 4.3 Komponenten zur Bereitstellung von Dienstgüte
- 4.4 Ressourcenverwaltung
- 4.5 Dienstgütearchitekturen im Internet
- 4.6 Einsatz von Dienstgütemechanismen

III. Multicast

5. Grundlagen
6. Multicast Routing
7. Multicast Transport

IV. Flexible Dienste und Selbstorganisation

8. Neuere Transportprotokolle
9. Aktive Netze
10. Peer-to-Peer

4.1 Was ist Dienstgüte?

- Dienst wird mit bestimmter **Güte** bzw. **Qualität** assoziiert
 - umfasst zahlreiche Aspekte
 - beschrieben durch Menge von Dienstmerkmalen, die Eigenschaften des Dienstes näher charakterisieren
 - zum Beispiel leistungsbezogene, betriebliche oder sicherheitsbezogene Merkmale
- Grad der Zusicherung je Merkmal
 - **Garantiert** (deterministisch)
 - **Statistisch**
 - Keine Zusicherung (**Best Effort**)
- ITU-T Definition E.800: „*Collective effect of service performances which determine the degree of **satisfaction** of a **user** of the service*“
 - Definition geht eher in Richtung **Quality of Experience (QoE)**

Dienstgüteparameter

■ Dienstgüteparameter (quantitativ oder qualitativ):

■ quantitative:

- Nachprüfbar durch Messung

■ qualitative:

- Evtl. nachprüfbar durch Vergleich (relative Spez.)
- Beeinflusst durch „Human Perception“
(z.B. Messen durch Ermittlung des Mean Opinion Score)

■ leistungsspezifische:

- Durchsatz
- Ende-zu-Ende-Verzögerung
- Verzögerungsschwankung (Delay-Jitter, meist kurz als Jitter bezeichnet)
- Zuverlässigkeit (Übertragungsfehler, Paketverlust, doppelte Pakete,...)
- Priorität (relativ)
- Verbindungsaufbauverzögerung und -fehlerwahrscheinlichkeit

Dienstklassen

■ Bestmögliche Klasse (Best-Effort):

- Dienstklasse des heutigen Internets
- keine Ressourcenreservierung
- mit verfügbaren Ressourcen so gut (und schnell) umgehen, wie möglich
- Konflikte sind „vorprogrammiert“

■ Statistische Klasse:

- Dienstgüteparameter werden nur statistisch eingehalten
z.B.: „80% der Pakete haben eine Verzögerung $< 100\text{ms}$ “
- Ressourcen werden bis zu einem gewissen Grad überbelegt
- Konflikte möglich

■ Deterministische Klasse:

- Dienstgüteparameter werden immer eingehalten (harte Garantie)
- Ressourcen stehen einem Nutzer exklusiv zur Verfügung
- keine Konflikte möglich, aber „Besetztfall“ (keine Ressourcen mehr übrig)

4.2 Wozu Dienstgüte im Internet?

- Fehlende Unterstützung für manche Anwendungen (insbesondere Echtzeitanwendungen, vor allem interaktive)
 - Derzeit manchmal schlechte Qualität bei „Audio/Video-Streaming“
- **Datenstrom (Flow)**: Folge von Paketen zwischen zwei Punkten im Netzwerk,
- **Ende-zu-Ende-Datenstrom („Micro-Flow“)**: Datenstrom zwischen zwei Anwendungen
- **Ressourcenmangel im Netzwerk**
→ Schwankungen der Dienstqualität einzelner Datenströme

Netzressourcen

■ Einfaches Routermodell für paketbasierte Netze

- Zwei Pakete mit gleichem Zielausgang: entweder verwerfen oder zwischenspeichern

■ Puffer

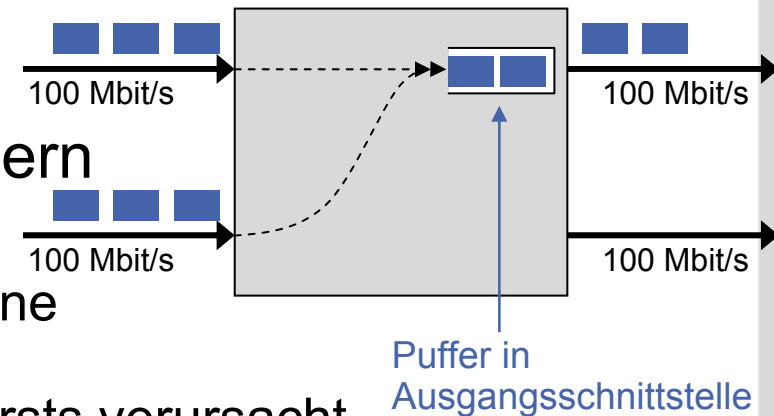
- zwecks Anpassung an verschiedene Bandbreiten
- mildert Paketverlust, der durch Bursts verursacht
- Paketverlust tritt auf, wenn Puffer voll (Tail-Drop)

■ Ressourcen

- Leitungskapazität (Datenrate, „Bandbreite“)
- Puffergröße

■ Pufferfüllstand hängt von Verkehrsmatrix ab

- Verkehrsmatrix: gibt an, welche Verkehrslast zwischen jedem Ein- und Ausgang fließt



Wo ist das Problem?

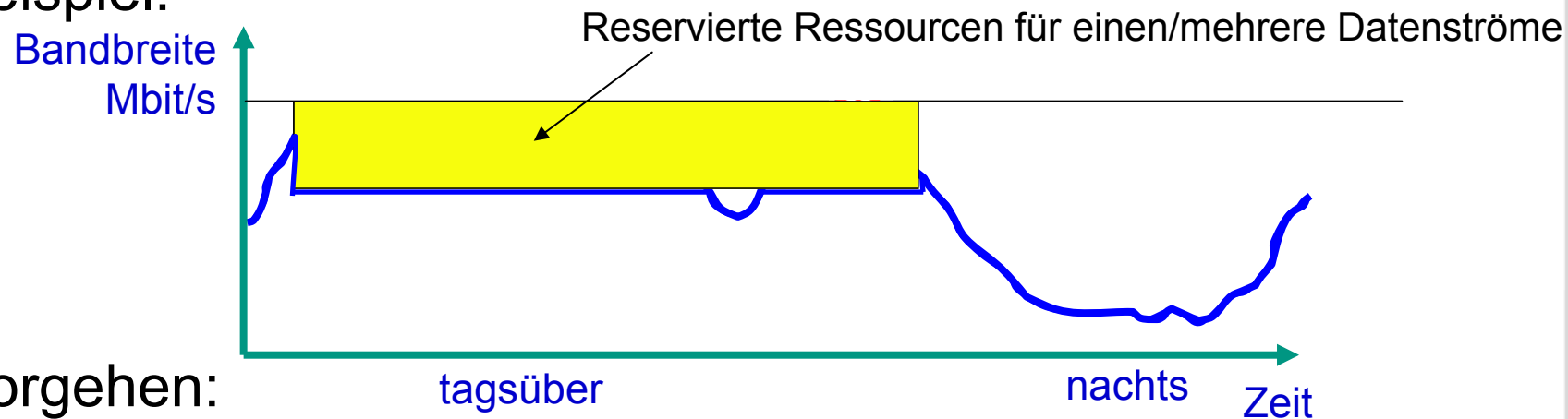
- **Ressourcenmangel** (Link Bandbreite/Puffer) im Netz verursacht Verzögerungsschwankungen und/oder Paketverlust
→ Datenstrom erfährt Schwankungen in der Dienstgüte
- Fehlende Unterstützung für einige Anwendungen, insbesondere **interaktive Echtzeitanwendungen**
 - Interaktive Anwendungen haben strenge Anforderungen an die Verzögerung: wenn ein Paket zu spät kommt ist es nutzlos
 - zeitweise schlechte oder unakzeptable Qualität für Audio/Video-Anwendungen

Garantierte Dienste im Next Generation Internet

- Internet bietet derzeit **keine** garantierten Dienste
 - notwendig für Multimedia- und Echtzeit-Anwendungen
 - Erfolg der Paketweiterleitung abhängig von Netzlast
- Internet der nächsten Generation muss Dienstgüte anbieten
 - garantierte Bandbreiten
 - minimale Paketlaufzeiten
 - begrenzter Jitter

→ Quality of Service

■ Beispiel:

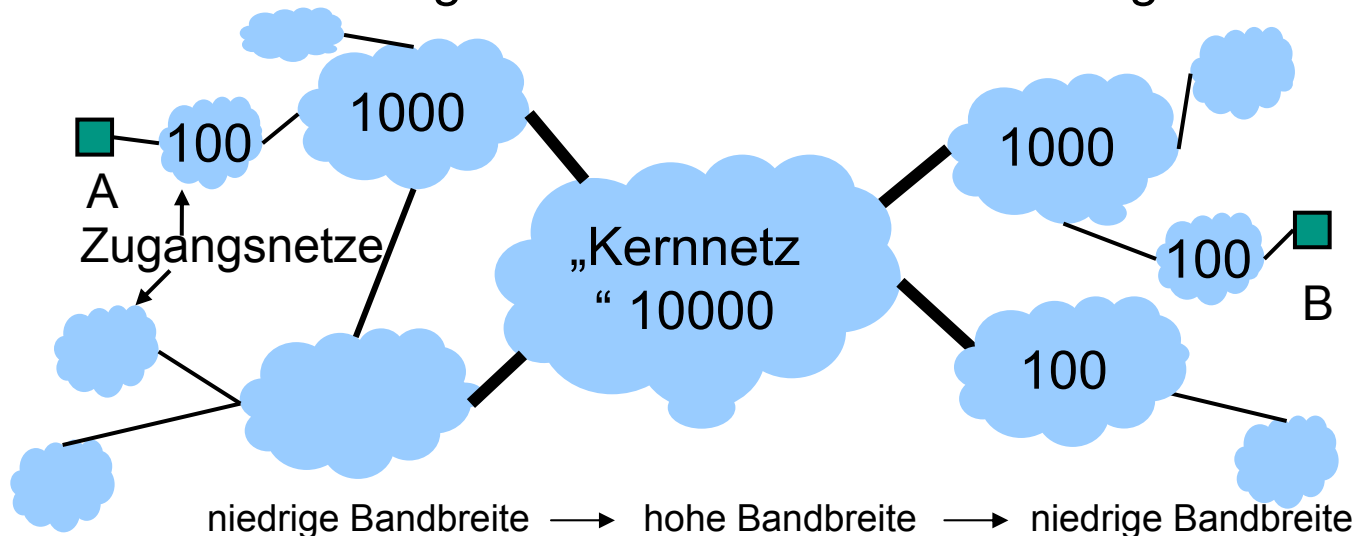


■ Vorgehen:

- Bereitstellung von Dienstqualität durch gezielte **Ressourcenmangelverwaltung** (z.B. Ressourcenreservierung)

Mythos „Over-Provisioning“

- „Over-Provisioning“: Last je Link meistens $< 50\%$
 - Kein Ressourcenmangel
- Übertragungskapazitätshierarchie vorhanden
 - Probleme
 - Verteilung (Verkehrsmatrix) für eingehenden Verkehr „unbekannt“
 - DDoS-Flooding für einzelne Anschlüsse möglich





Warteschlangen sind unvermeidbar

- Wenn man die Bandbreitenhierarchie wieder hinabsteigt
 - Pakete zwischenspeichern
 - Füllstand der Warteschlange variiert, Überlauf möglich
→ Dienstgüte schwankt
- Wie groß sollte Pufferkapazität sein?
 - Antwort ist anwendungsabhängig
 - Größere Puffer erhöhen die Auslastung, aber erhöhen auch die Verzögerung
 - Diskussion
 - Daumenregel empfiehlt eine Umlaufzeit, z.B. 250 ms
 - Einige Forscher schlagen vor, kürzere Puffer zu Verwenden
 - führt zu höheren Verlustrate, insbes. für UDP-Ströme
 - niedrigere Verzögerung

Dienstgüteunterstützung?

Gegenargumente

- „Unendliche“ Bandbreite (echtes Over-Provisioning)
- Einfache Prioritäten sind ausreichend
 - gute Grundlage, aber alleine nicht ausreichend
- Adaptive Anwendungen
 - Adaptive Anwendungen können sich an die aktuellen Gegebenheiten im Netz anpassen
 - schon, aber nicht beliebig weit nach unten, z.B. gewisse Mindestqualität für Sprachcodec notwendig
 - Aktuelle Entwicklung in der Mobilkommunikation
 - mehr Intelligenz in den Anwendungen (klassisches E2E-Argument)
 - Aber: Basisdienste sind wichtig, z.B. hinsichtlich Verzögerungen

"Traditionelle" Anwendungen

- Auch als „**elastische Anwendungen**“ bezeichnet
 - Hier besteht **keine** harte **obere Zeitgrenze** für die Auslieferung der Daten
 - Daten werden durch längere Verzögerungszeiten nicht unbrauchbar
 - Können problemlos im heutigen Internet realisiert werden
- Die **Verzögerungsanforderungen** unterschiedlicher elastischer Anwendungen können stark variieren
 - Die folgenden drei Kategorien können unterschieden werden (mit steigender Toleranz hinsichtlich von Verzögerungen):
 - Interaktiv (z.B. Telnet)
 - Interaktiv mit Bursts (z.B. FTP)
 - Asynchron (z.B. E-Mail)

Realzeit-Anwendungen

■ Problem:

- Realzeit-Anwendungen stellen zeitliche Anforderungen an Auslieferung der Daten

■ Beispielsweise: Audio- und Videoanwendungen

■ Redundanz

- Aus redundanter Information resultiert eine gewisse Toleranz hinsichtlich Datenverlusten
- Zu spät eintreffende Daten stellen in diesem Sinne auch Verluste (d.h. Fehler) dar

■ Adaptivität

- **Verzögerungsadaptivität:** Pufferschranke kann sich z.B. an der mittleren Verzögerungszeit im Netz orientieren‘

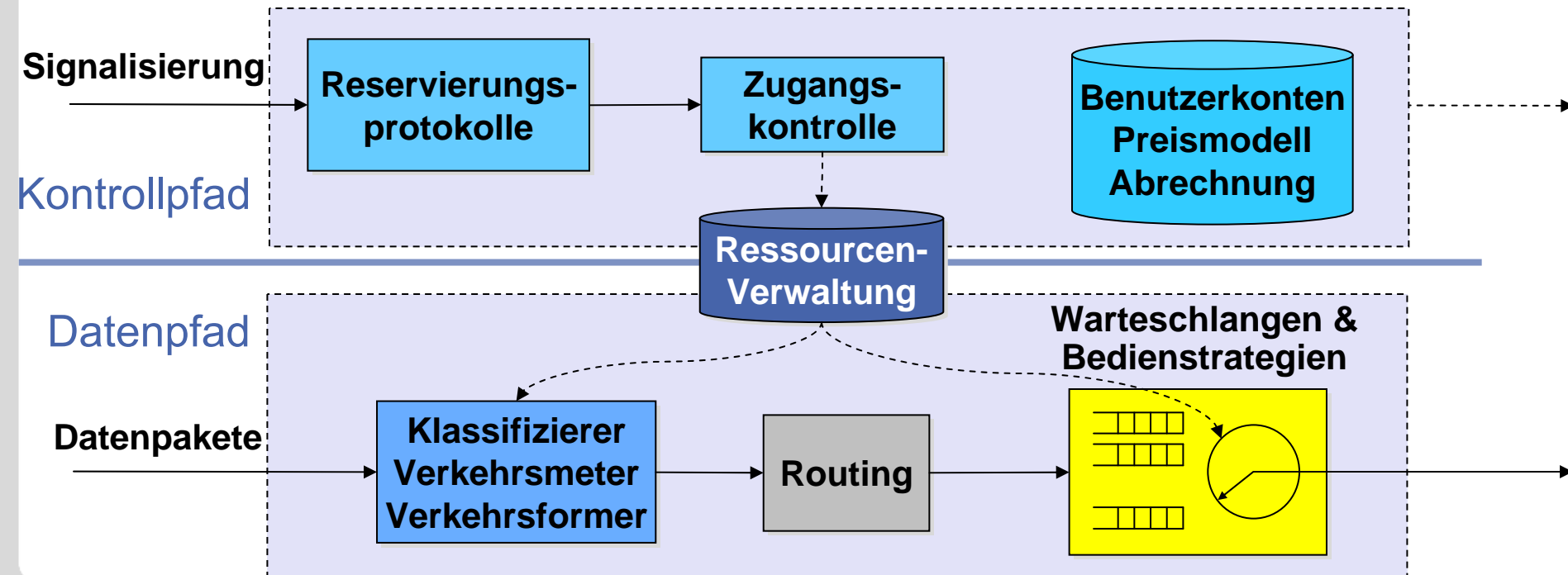


- **Ratenadaptivität:** Videoanwendungen können z.B. Bitrate zugunsten der Qualität senken



4.3 Komponenten zur Bereitstellung von Dienstgüte

- Was wird benötigt, um Dienstgüte zur Verfügung zu stellen?
 - Differenzierung der Pakete im Datenpfad → „Weiterleitungsklassen“ sowie „Policing“, d.h. Überwachung der ankommenden Verkehrsmenge
 - Ressourcenbasierte Zugangskontrolle für Weiterleitungsklassen in der Kontrollebene (Vermeidung von Überlast innerhalb einer Klasse)



Verkehrsbeeinflussung

■ Beispiel aus dem realen Leben: Verkehrsbeeinflussungsanlagen



■ Verkehrsbeeinflussungsmechanismen im Netz

- Verkehrsmeter
- Verkehrsformer
- Klassifikationselemente

Komponenten im Datenpfad

■ Verkehrsmeter

- Prüfen der Datenpakete auf Konformität zum Verkehrsvertrag (Nutzungskontrolle)

■ Verkehrsformer

- Ändern der Charakteristik eines Datenstroms/Aggregats
- Wiederherstellen von Konformität

■ Klassifikationselemente

- Einteilung von einzelnen Datenpaketen in Datenströme
- Datenströme: Einheiten auf denen Dienstgütemechanismen arbeiten

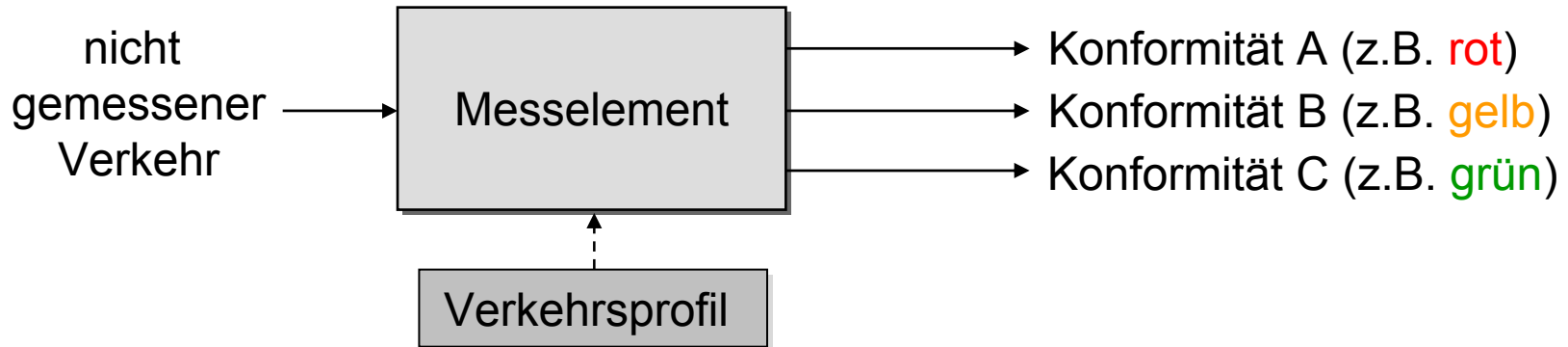
■ Warteschlangen

- Unterschiedliche Behandlung von Datenströmen

■ Bedienstrategien für Warteschlangen

- Zuteilung der Ressourcen (z. B. Bandbreite, Rechenzeit, Speicher) durch Scheduler

4.3.1 Verkehrsmeter

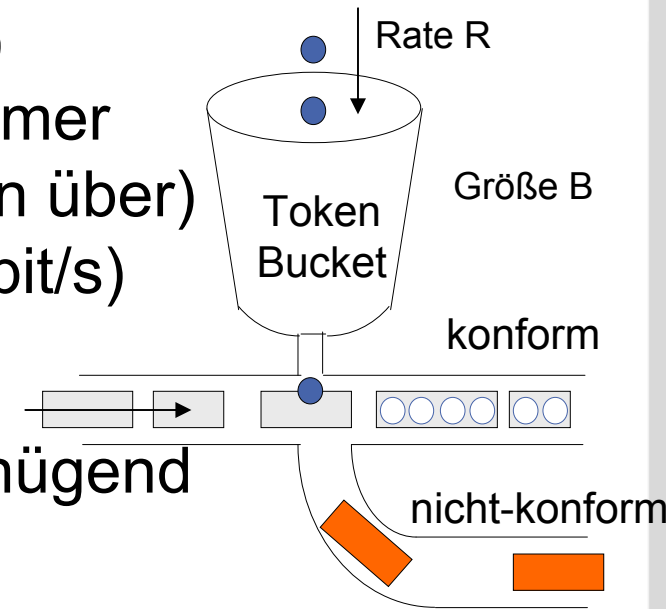


■ Verkehrsmeter begutachten Datenströme:


- Überprüfen der Konformität mit Verkehrsvertrag, keine Beeinflussung des Verkehrs
- Eingabe:
 - Verkehrsprofil (Datenstrom, Dienstgüteparameter, Dienstklasse)
- Ausgabe:
 - konform, nicht-konform
 - mehrere Konformitätsklassen (0, .., n) oder (rot, gelb, grün)
- Beispiele:
 - Token Bucket (überprüft Rate und Burst)
 - Average Rate Meter (Moving Window)

Verkehrsmeter: Token Bucket

- Eimer mit Größe B **Token** (**Senderechte**)
- Token „tröpfeln“ mit der Rate R in den Eimer
- maximal B Token im Eimer (Token laufen über)
- überwacht die Einhaltung einer Rate R (bit/s) mit einer Toleranz (Burst) B (byte)
- Pakete werden als **konform** markiert, wenn bei Ankunft eines Pakets noch genügend Token im Eimer vorhanden sind (oft: 1 Byte = 1 Token)
 - **Strenge Variante**: für Paket mit Länge L müssen mindestens L Token vorhanden sein
 - **Lockere Variante**: Bucket muss mindestens ein Token enthalten, fehlende Tokens werden im Voraus geborgt (Tokenstand wird negativ)
- Andernfalls wird ein ankommendes Paket als **nicht-konform** markiert



Verkehrsbeschreibung

- Token Buckets können als Grundlage zur mathematisch vollständigen Modellierung des erzeugten Verkehrs dienen  [RFC2210, RFC2215]
 - **Token Rate** r (mittlere Rate), **Bucket Tiefe** b
 - **Peak rate** p : evtl. maximale Verkehrsgenerierungsrate, max. physikalische Rate, oder unendlich
 - **Minimum policed unit** m : Größe des kleinsten Pakets, das durch die Anwendung generiert wird.
 - maximale Paket-Rate kann durch b und m berechnet werden
 - Berechnung des maximalen zusätzlichen Bandbreitenaufwands, um Pakete über bestimmte Link-Layer-Technik zu übertragen, mit Hilfe des Verhältnisses der Link-Layer-Kopfgröße zu m
 - **Maximum packet size** M : größtes konformes Paket (größere Pakete werden als nicht konform betrachtet)

4.3.2. Verkehrsformer

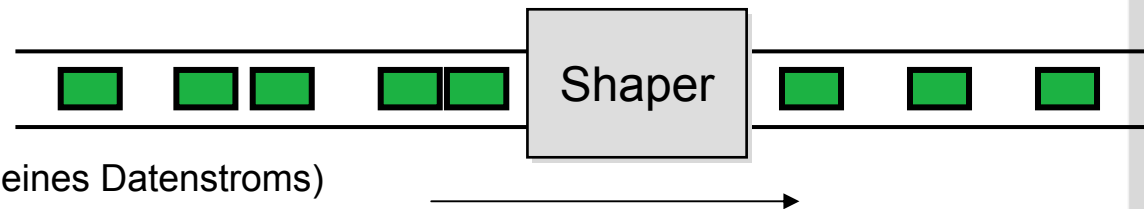
■ Verkehrsformer beeinflussen **aktiv** die Charakteristik des Verkehrs:

- Ziel: Wiederherstellung von Konformität bzw. eines bestimmten Verkehrsmusters

Beispiele:

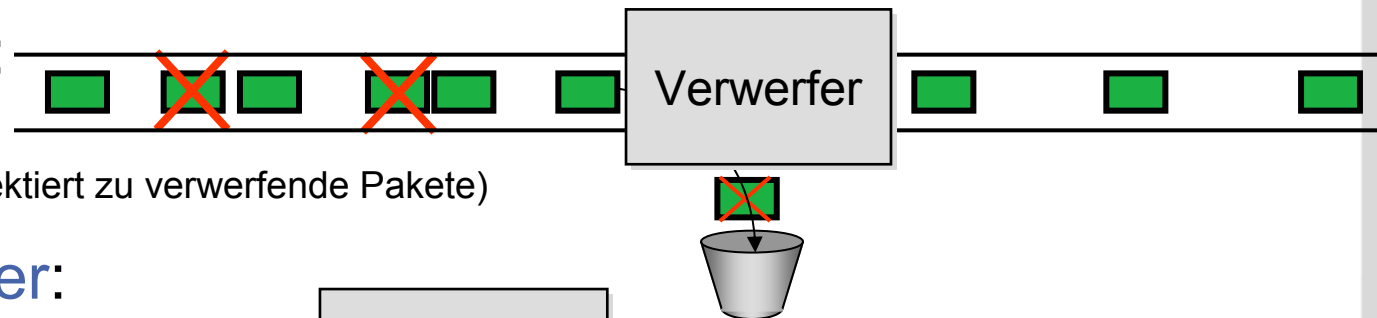
■ **Verkehrsglätter:**

(glättet alle Pakete eines Datenstroms)



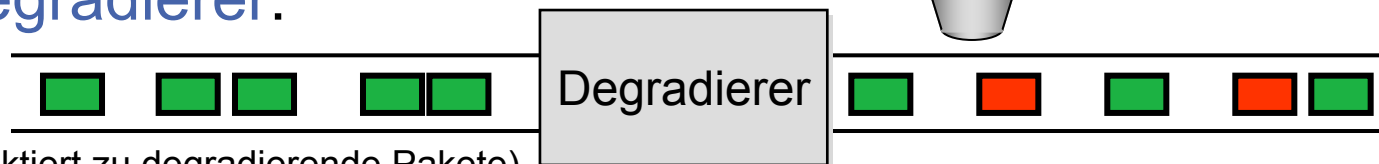
■ **Verwerfer:**

(selektiert zu verwerfende Pakete)



■ **Degradierer:**

(selektiert zu degradierende Pakete)

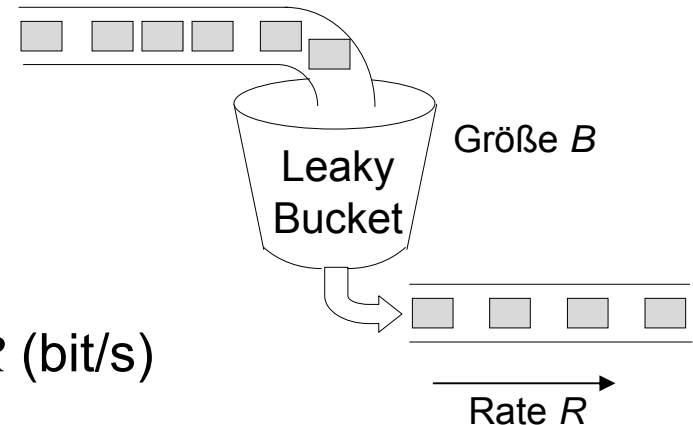


- ordnet dem Datenstrom eine niedrigere Dienstkategorie zu

Verkehrsmeter und -former: Leaky Bucket

■ Leaky-Bucket:

- Eimer mit Größe B byte
- Pakete werden in den Eimer gepackt und „tröpfeln“ mit der Rate R aus dem Eimer heraus (rinnender Eimer)
- max. B byte im Eimer
- überwacht die Einhaltung einer Rate R (bit/s) mit einer Toleranz (Burst) B (byte)
- **glättet** den Ausgangsstrom auf die Rate R
- Ein Paket der Größe L wird gesendet, wenn L byte aus dem Eimer getropft sind
- Hat ein neu eintreffendes Paket keinen Platz mehr im Eimer, wird es verworfen, bzw. unterschiedlich behandelt
- Pakete im Eimer werden so lange verzögert, bis das Senden konform zur Rate ist (zusätzliche Verzögerung evtl. auch von Nachteil!)

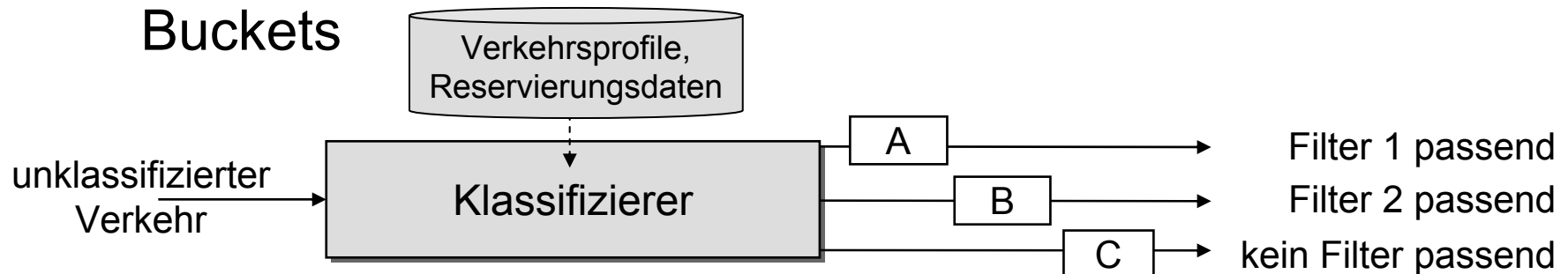


4.3.3 Klassifikationselemente

■ **Klassifizierer** identifizieren Datenströme aus der Menge aller Pakete

■ **Zuordnung**: Paket → Verkehrsprofil

■ **Verkehrsprofil**: z.B. erlaubte Parameter für Token-Buckets



Klassifizierer

■ Arten von Klassifizierern:

■ Klassifizierer für aggregiertes Verhalten (Aggregate Classifier)

- viele Datenströme erfahren das gleiche Verhalten (Behavior Aggregate)
- bspw.: DS (DiffServ) Codepoint im Paket
- sehr einfache und schnelle Klassifikation (= Zugriff auf Array)

■ Klassifizierer für mehrere Paketkopffelder (Multi-Field Classifier)

- Kombination eines oder mehrerer Paketkopffelder
bspw. IP-Adressen, IP-Protokollfeld, Port-Nummern bei TCP/UDP-Paketen (müssen zugänglich sein)
- sehr komplizierte und aufwändige Klassifikation (viele Verfahren, aber kein perfektes)

■ Klassifizierung nach Eingangs-Interface

■ Klassifikationsinformationen stehen in Datenbasis

4.3.4 Warteschlangen und Bedienstrategien

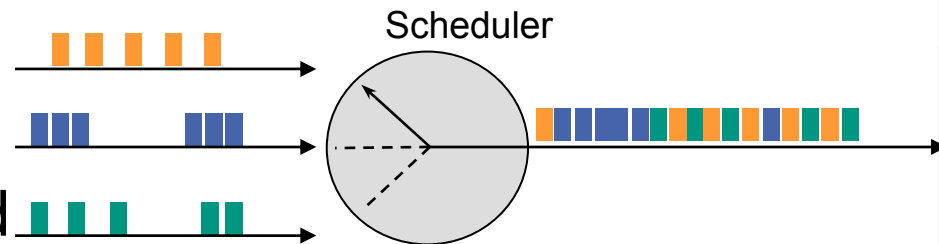
- Warteschlangen speichern Pakete:
 - wie sie ankommen (FIFO - First In First Out)
 - nach zeitlicher Dringlichkeit (EDF - Earliest Deadline First)
 - nach Priorität (verschiedene logische Warteschlangen)

- **Bedienstrategien (Scheduling Algorithmen)**

bestimmen die nächste zu bedienende

Warteschlange bzw. das nächste zu bedienende Paket:

- Legt die Reihenfolge der exklusiven Ressourcennutzung (Bandbreite, Speicher) abhängig von einem Scheduling-Algorithmus fest
- Scheduler sind in allen Schichten erforderlich, in denen gemultiplext wird



Beispiele für Bedienstrategien

- **Round Robin** (Reihum-Auswahl):
alle nacheinander – keiner wird bevorzugt
- **Simple Priority Queueing/Strict Priority Queueing**:
 - immer erst hochpriorie Warteschlange bedienen
 - permanente Bevorzugung der hochpriorien Warteschlange
- **Weighted Fair Queueing**:
 - Round Robin mit Gewichtung
 - Warteschlange mit hohem Gewicht wird öfter bedient



Scheduling-Algorithmen: Anforderungen und Eigenschaften

■ Anforderungen

■ Einfache Implementierung

- Nur wenig Zeit zur Bearbeitung bei Gbit/s
- Zeitkomplexität möglichst $O(1)$, nicht $O(n)$, wobei n =Anzahl Datenströme
- geringer Speicherbedarf

■ Fairness (f. Best-Effort-Dienste)

→ wenn Kriterium „Max-min Fair Share“ erfüllt

- Absicherung (gegen sich fehlverhaltende Datenströme)

■ Eigenschaften

- Arbeitserhaltend / Nicht-arbeitserhaltend
- Verdrängend / Nicht-verdrängend
- Leistungsgarantien

Max-min Fair Share (1)

■ Formale Definition

- Ressourcen werden gemäß steigenden Anforderungen zugeteilt
- Keine Anforderung erhält mehr Ressourcen als benötigt
- Anforderungen, die ungenügend bedient werden, erhalten gleichen Anteil an noch verfügbaren Ressourcen

■ Vorgehensweise

- Menge von Anforderungen $1, \dots, n$ mit Ressourcenanforderungen x_1, \dots, x_n . Die Anforderungen seien so angeordnet, dass gilt: $x_1 \leq x_2 \leq \dots \leq x_n$
- Der Server habe eine Kapazität C . Dann erhält x_1 den Anteil C/n zugeteilt.

Max-min Fair Share (2)

- Dieser Anteil kann mehr sein als tatsächlich benötigt wird. In einem solchen Fall steht $C/n - x_1$ als ungenutzte Ressource weiter zur Verfügung und wird auf die weiteren Anforderungen gleichmäßig aufgeteilt: Jede Anforderung erhält $C/n + (C/n - x_1)/(n-1)$. Dies setzt sich solange fort, bis eine Anforderung nicht erfüllt werden kann.
- Ergebnis
 - Maximiert minimalen Anteil einer Anforderung, deren Forderung nicht erfüllt werden konnte
- Erweiterung: Einführung von Gewichten

Beispiel: Max-min Fair Share

■ Aufgabe

- Berechnen Sie die Allokation von Ressourcen gemäß Max-min Fair Share für vier Anforderungen von je 2, 2.6, 4 und 5 Anteilen einer Ressource der Kapazität 10

■ Lösung

- Vorgehen in mehreren Runden
- Erste Runde
 - Aufteilen der Ressource in vier Teile à 2.5
 - Rest von 0.5 für die nächste Runde
- Zweite Runde
 -

Grundlegende Alternativen

■ Vier grundlegende Entscheidungen

- Wieviele **Prioritätsstufen** werden unterschieden?
- Ist eine Stufe **Work Conserving** (arbeitserhaltend) oder nicht?
 - Arbeitserhaltend heißt: der Scheduler ist nur dann im Leerlauf, wenn sich keine Anforderungen in den Warteschlangen befinden
 - In der Praxis heute nahezu immer Work Conserving
- Grad der **Aggregation**, der auf einer Stufe gewählt wird
- **Bedienreihenfolge innerhalb einer Stufe** muss festgelegt werden

Prioritätsstufen

- Jedem Datenstrom wird eine Priorität zugeordnet
- Beispiel: 3 unterschiedliche Prioritätsklassen
 - Hohe Priorität für wichtige Dateneinheiten (z.B. zur Netzwerksteuerung)
 - Mittlere Priorität für garantierte Dienste
 - Niedrige Priorität für Best-Effort Dienste
- Problem
 - Absicherung der Datenströme ist nicht gegeben. Ein sich fehl verhaltender Datenstrom höherer Priorität beeinflusst diejenigen niedrigerer Priorität.
- Implementierung
 - Einfach. Wenig Zustandshaltung erforderlich.

Conservation Law (1)

- Falls ein Scheduler **Work-Conserving** (arbeitserhaltend) ist, gilt folgendes
 - Die Summe der mittleren Warteschlangenverzögerungen einer Menge gemultiplexer Datenströme, gewichtet mit ihrem jeweiligen Anteil an der Last, ist **unabhängig** vom Scheduling-Algorithmus.
 - Work-Conserving-Eigenschaft:
 - Ein Scheduler ist nur dann im Leerlauf, wenn sich keine Anforderungen in der Warteschlange bzw. den Warteschlangen befinden.

Conservation Law (2)

■ Formal

- N Datenströme, jeweils mit einer mittleren Datenrate von λ_i und einer mittleren Bedienzeit pro Dateneinheit von x_i . Die mittlere Auslastung des Übertragungsabschnitts durch Datenstrom i ergibt sich als: $\rho_i = \lambda_i x_i$

Sei q_i die mittlere Wartezeit am Scheduler, dann besagt das **Conservation Law** das folgende: $\sum_{i=1}^N \rho_i q_i = \text{Constant}$

- Folge: Scheduler kann die mittlere Warteschlangenverzögerung eines Datenstroms im Vergleich zum FIFO-Algorithmus nur zu Lasten anderer Datenströme reduzieren

Aggregation

■ Aggregation

- Mehrere Datenströme werden aggregiert betrachtet hinsichtlich der Bedienreihenfolge

■ Vorteil

- Zustandshaltung reduziert sich → Skalierbarkeit
 - In großen Netzen ein sehr wichtiges Argument

■ Beispiel

- Einteilung der Datenströme in Klassen
- Aggregierte Betrachtung der Klassen

■ Problem

- Datenströme einer Klasse sind nicht voneinander isoliert, d.h. sie beeinflussen sich mehr oder weniger gegenseitig

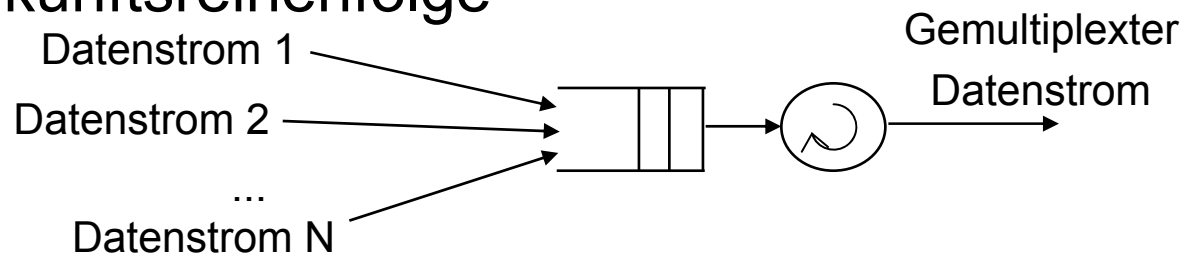
Beispiele zur Bedienreihenfolge

- Die folgenden Beispiele zur Bedienreihenfolge werden diskutiert
 - Bedienreihenfolge für Best-Effort-Dienste
 - First In First Out (FIFO)
 - Generalized Processor Sharing (GPS)
 - Round Robin (RR)
 - Bedienreihenfolge bei Dienstgarantien
 - Fair Queueing
 - Weighted Fair Queueing (WFQ)
 - Earliest Deadline First (EDF) / Earliest Due Date (EDD)

FCFS-Scheduling (1)

■ FCFS (First-Come-First-Served) Strategie

- Betrachtung nur einer Ausgangs-Warteschlange:
FIFO-Queue (First-In-First-Out)
- Bearbeitung der Dateneinheiten (oder Anforderungen) in Ankunftsreihenfolge



- Ist kein Pufferplatz mehr frei, so werden neue Dateneinheiten verworfen („Tail-Drop“)
- Verantwortung für Staukontrolle wird zu den Endsystemen verschoben

■ Vorteil

- Einfach zu implementieren

FCFS-Scheduling (2)

■ Nachteil

- Kann nicht zwischen verschiedenen Datenströmen unterscheiden → Nicht möglich, einzelnen Verbindungen unterschiedliche Verzögerung zuzuordnen
- Datenströme mit langen Dateneinheiten erhalten besseren Service
- „Gierige“ Verbindungen beeinflussen andere

■ FCFS und Prioritäten

- Warteschlangen mit unterschiedlichen Prioritäten. Innerhalb einer Warteschlange: FCFS-Scheduling
- Höchste Priorität wird stets zuerst bedient (**Simple Priority Scheduling** oder **Strict Priority Scheduling**)
 - Keine definitive Zuordnung einer Verzögerung für einzelne Datenströme möglich

Generalized Processor Sharing (GPS)

- Ziel bei Best-Effort-Diensten
 - Erzielung von „Max-min Fair Share“, d.h. Fairness.
- Generalized Processor Sharing (GPS)
 - Ideales, Work Conserving Scheduling. Nicht implementierbar.
 - Restbandbreite wird fair zwischen konkurrierenden Datenströme aufgeteilt
- Vorgehen
 - Dateneinheiten werden unterschiedlichen Warteschlangen zugeordnet
 - Jede nicht leere Warteschlange wird regelmäßig besucht und eine unendlich kleine Menge der Dateneinheit bedient (unmöglich in der Praxis)
 - In jeder endlichen Zeit können damit alle Warteschlangen besucht werden

GPS formal betrachtet

■ Formal

- Datenstrom i erhält ein Gewicht $\phi(i)$
- Server bedient $\Sigma(i, \tau, t)$ Daten des i -ten Datenstroms im Intervall $[\tau, t]$
- Für alle anderen Datenströme j , die ebenfalls Daten zu bedienen haben, gilt:

$$\frac{\Sigma(i, \tau, t)}{\Sigma(j, \tau, t)} \geq \frac{\phi(i)}{\phi(j)}$$

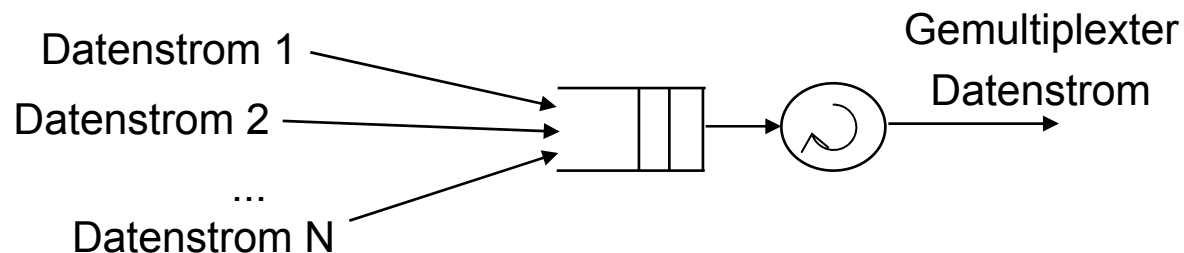
Round-Robin-Scheduling

■ Vorgehen

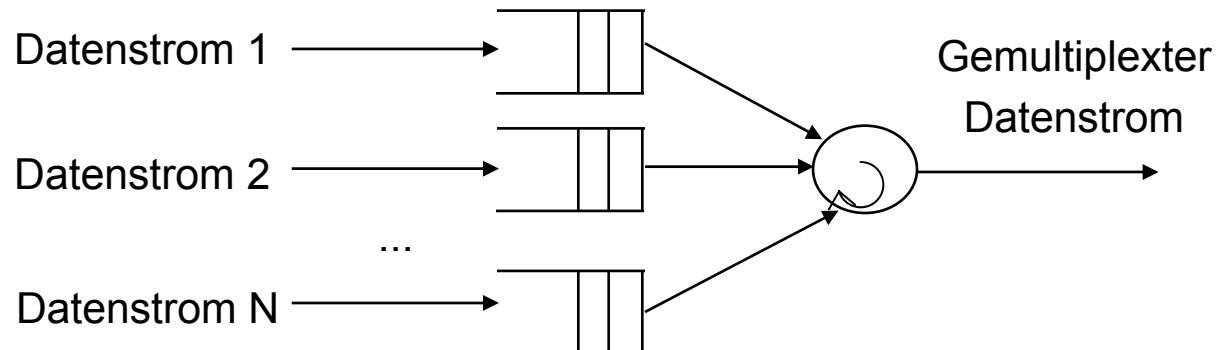
- Bedient in einer Runde eine Dateneinheit jeder nicht leeren Warteschlange (Reihumverfahren)
- Emuliert GPS gut, wenn alle Datenströme gleiche Gewichte haben und die Dateneinheiten alle gleich lang sind

■ FIFO versus Round-Robin

■ FIFO



■ Round-Robin



Weighted Round-Robin (WRR)

- Berücksichtigt Gewichte der Datenströme
- Unfair, wenn Pakete unterschiedliche Länge besitzen oder Gewichte verschieden (oft beabsichtigt)
- Bei **fester Paketgröße**: es werden, dem Gewicht entsprechend, mehrere Pakete eines Datenstroms je Runde bedient. Gewichte werden auf Ganzzahlen normalisiert.
- Vorteil
 - Verkehr einer unfairen Quelle belastet andere Datenströme nicht
- Probleme
 - Verschieden lange Dateneinheiten

Fair Queueing (1)

■ Approximation eines „Bit-für-Bit“-Round-Robin

- Ermitteln des Zeitpunktes, zu dem das Senden einer Dateneinheit mit einem Bit-für-Bit Round-Robin theoretisch beendet wäre (**Endezeit**);
Pakete werden dann in Reihenfolge ihrer virtuellen **Endezeit** bedient

■ Vorgehensweise

- Eine **Rundenzahl** wird bestimmt, welche die Anzahl der Bit-für-Bit-Runden angibt, die zu einem gegebenen Zeitpunkt beendet sind
 - Jede Runde nimmt eine variable (reale) Zeitspanne in Anspruch
 - Trifft Paket der Länge D auf
 - leere Warteschlange (inaktiver Datenstrom) bei Rundenzahl R , so ist seine **Endezeit** $R+D$
 - volle Warteschlange (aktiver Datenstrom), so ist seine **Endezeit** $F+D$, wenn F die Rundenzahl des vorhergehenden Pakets bezeichnet

Fair Queueing (2)

- Ist die Rundenzahl bekannt, so kann die Endezeit also folgendermaßen berechnet werden
 - $D(i,k,t)$: Länge der k -ten Dateneinheit, die zur (realen) Zeit t beim Datenstrom i ankommt
 - $R(t)$ sei die Rundenzahl zum (realen) Zeitpunkt t
 - $F(i,k-1,t)$ sei Endezeit der $k-1$ -ten Dateneinheit des Datenstroms i zum Zeitpunkt t
 - Endezeit $F(i,k,t) = \max \{ F(i,k-1,t), R(t) \} + D(i,k,t)$
- Definition der **Rundenzahl** als Variable, die invers zur Anzahl aktiver Datenströme wächst, d.h. $R(t)$ wächst mit zunehmender Anzahl aktiver Datenströme langsamer
- $F(i,k,t)$ und $R(t)$ sind virtuelle Zeiten!
- Probleme: Zustandshaltung und keine Gewichtung

Weighted Fair Queueing (WFQ)

- Einführung einer Gewichtung pro Datenstrom ermöglicht Priorisierung

- Berechnung der **Endezeit**

- $$F(i, k, t) = \max \{ F(i, k - 1, t), R(t) \} + \frac{D(i, k, t)}{\phi(i)}$$

- **Rundenzahl**

- erhöht sich mit $\frac{1}{\sum_{i=1}^{N(t)} \phi(i)}$, $N(t)$ ist die zum Zeitpunkt t aktive Anzahl an Datenströmen

- **Positive Eigenschaften**

- Gezielte Bevorzugung bestimmter Datenströme möglich
 - Datenströme sind voneinander abgesichert
 - Obere Schranke für Ende-zu-Ende-Verzögerung kann teilweise angegeben werden

Weighted Fair Queueing (Forts.)

■ Nachteile

- Gewicht als zusätzliche Angabe je Datenstrom erforderlich
- Geringfügig höherer Berechnungsaufwand als bei Fair Queueing
- Benutzer sollten (ratenbasierte) Flusskontrolle implementieren, ansonsten können Verluste von Dateneinheiten die Folge sein

Beispiel: Weighted Fair Queueing

■ Aufgabe

- Dateneinheiten der Größen 1, 2, und 2 erreichen einen WFQ-Scheduler zum Zeitpunkt 0. Sie gehören zu den gleichgewichteten Datenströmen A, B und C. Für Datenstrom A wird zum Zeitpunkt 4 eine weitere Dateneinheit der Länge 2 empfangen. Die Datenrate sei eine Größeneinheit/Sekunde. Bestimmen Sie die Endezeiten der Dateneinheiten. Bei welcher Rundenzahl befindet sich das System im Leerlauf?

■ Lösung

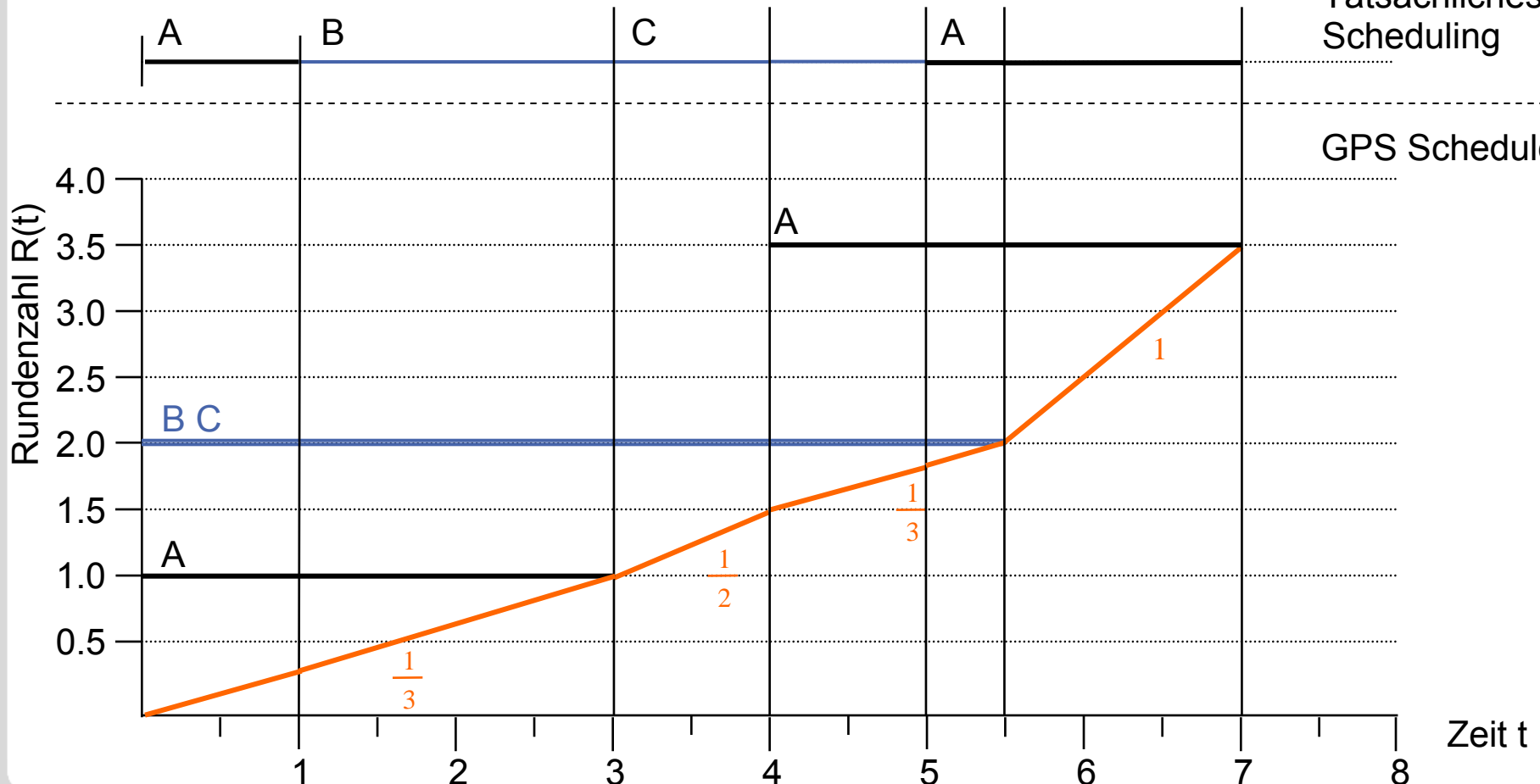
- Endezeiten der Datenströme zum Zeitpunkt 0: 0
- Endezeit der ersten Dateneinheit: $\max(0,0) + 1 = 1$

WFQ: Beispiel 1

- Prinzip: Beendigungszeiten eines GPS-Scheduler bestimmt Bearbeitungsreihenfolge
- $F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + D(i,k,t)$
- $F(A,0,0) = 0 + 1 = 1$, $F(B,0,0) = 0 + 2 = 2$, $F(C,0,0) = 0 + 2 = 2$
- $F(A,1,4) = \max\{1, 1.5\} + 2 = 3.5$

Tatsächliches Scheduling

GPS Scheduler



WFQ: Beispiel 2

- Datenströme A, B und C haben folgende Dateneinheiten zu verschicken:
 $D(A,0,0)=2$, $D(B,0,0)=2$, $D(C,0,1)=1$, $D(C,1,2)=1$
- Das heißt: Datenstrom A und B haben jeweils zum Zeitpunkt 0 eine Dateneinheit der Größeneinheit 2 zu senden, Datenstrom C habe zu den Zeitpunkten 1 und 2 jeweils eine Dateneinheit der Größe 1 zu senden.
Datenstrom C habe außerdem die Gewichtung $\phi(C)=2$, Datenströme A und B jeweils $\phi(i)=1$. Ausgangsdatenrate: eine Größeneinheit/s.

Noch zu sendende Datenmenge	A	2	1.5	1.25	1	0.75	0.5	0
	B	2	1.5	1.25	1	0.75	0.5	0
	C	0	1	1.5	1	0.5	0	0
	Reale Zeit t[s]	0	1	2	3	4	5	6
	Rundenzahl R(t)	0	0.5	0.75	1	1.25	1.5	2

- $F(A,0,0)=2$, $F(B,0,0)=2$, $F(C,0,1)=\max(0,R(1))+1/2=\max(0,0.5)+1/2=1$
- $F(C,1,2)=\max(0,R(2))+1/2=\max(1,0.75)+1/2=1.5$
- Nach 6s ist der Scheduler fertig und die durch die berechneten Endezeitpunkte bestimmte Sendereihenfolge sieht folgendermaßen aus: C0,C1,A,B (alternativ: C0,C1,B,A)

WFQ – Verzögerungsgarantie (1)

- WFQ limitiert die den einzelnen Datenströmen zugeordnete Bandbreite, da dem Datenstrom i beim h -ten Hop bzw. Scheduler der folgende Anteil der Bandbreite zugeordnet wird: $\frac{\phi(i,h)}{\sum_{j=1}^{N_h} \phi(j,h)}$ (N_h : Anzahl der Datenströme im Scheduler h)

- Für die Ende-zu-Ende-Verzögerung lässt sich eine Schranke angeben

- Voraussetzung: Quelle i kann zu jedem Zeitpunkt nur $\sigma(i) + \rho(i)t$ Bits senden, wobei σ (Bucket-Tiefe, max. Burstgröße) und ρ (Rate) die Parameter eines Token-Bucket sind
- Für die Bedienrate $g(i,h)$, die einem Datenstrom i vom h -ten durchlaufenen Scheduler zugewiesen wird, gilt: $g(i,h) = r(h) \cdot \frac{\phi(i,h)}{\sum_{j=1}^{N_h} \phi(j,h)}$ $r(h)$ ist die Datenrate des Übertragungsabschnitts (Link)

WFQ – Verzögerungsgarantie (2)

- Annahme: $g(i) \geq \rho(i)$ wobei $g(i)$ das kleinste aller $g(i, h)$ ist, also $g(i) := \min_{h=1..H} g(i, h)$

- $g(i)$ ist minimal garantierte Datenrate entlang des Wegs

- Ansonsten steigt die Warteschlange an einem Scheduler ohne Grenze

- L_{max} : größte erlaubte Dateneinheit, H : Anzahl der Scheduler

- Obere Schranke für die Verzögerung eines Pakets des Datenstroms i ist dann:

$$\frac{\sigma(i)}{g(i)} + \sum_{h=1}^H \left(\frac{L_{max}}{r(h)} + \frac{L_{max}}{g(i, h)} \right)$$

$\frac{\sigma(i)}{g(i)}$ Bedienzeit für maximalen Burst mit minimal garantierter Rate

$\frac{L_{max}}{r(h)}$ Übertragungsdauer eines Pakets mit max. Länge bei Linkspeed (nicht verdrängbares Paket das gerade zuvor bedient wird)

$\frac{L_{max}}{g(i, h)}$ Bedienzeit für maximal langes Paket mit zugesicherter Rate

- aber: Verzögerungsgarantie abhängig von Datenrate

Earliest Deadline First (EDF)

■ Problem

- WFQ-Garantie für Verzögerung abhängig von zugesicherter Rate des Datenstroms

■ Vorgehensweise EDF

- Jeder Dateneinheit wird eine **Deadline** zugewiesen. Scheduler bearbeitet Dateneinheiten gemäß deren Deadline.
- Beim Verbindungsaufbau wird ein Verkehrsvertrag mit dem Scheduler ausgehandelt auf der Basis der maximalen Rate (Peakrate) einer Verbindung
 - Begrenzung der Worst-case-Verzögerung wird garantiert
 - Nachteil: Ressourcenreservierung auf der Basis der Peakrate führt zu schlechter Ausnutzung der Bandbreite

■ Vorteil

- Garantie der Ende-zu-Ende-Verzögerung unabhängig von der Bandbreite des Datenstroms

■ Nachteil

- Aufwändige Implementierung (Zeitstempel, Sortierung)
- Paketformat muss Zeitstempel mitführen

4.4 Ressourcenverwaltung

■ Problem

- Bereitstehende Ressourcen sind beschränkt
- Anwendungen konkurrieren um gemeinsam zu benutzende Ressourcen
- **Ressourcen**: Übertragungskapazität („Bandbreite“), Pufferkapazität, Rechenzeit

■ Aufgaben

- Regelung, wer Ressourcen benutzen darf
 - **Zugangskontrolle** (vor Reservierung und Nutzung)
 - Ressourcen-basiert
 - Politik-basiert (Policy-based)
 - **Nutzungskontrolle** (während Nutzung, Policing)
- Zugriff auf Ressourcen
 - Exklusiver Zugriff bei expliziter Reservierung von Ressourcen
 - Konkurrierender Zugriff:
 - Bedienstrategien, Verteilstrategien, etc.

Dienstgüteaushandlung

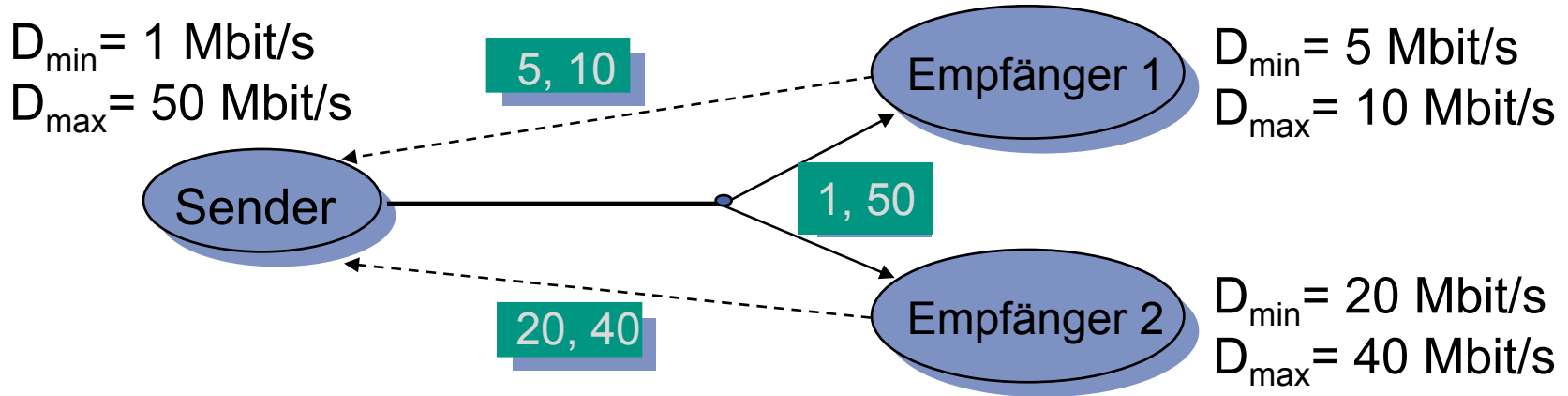
■ Problem

- Kommunikationspartner müssen Dienstgütewünsche kommunizieren und die bereitstellbare Dienstgüte aushandeln
 - Wer ist noch in den Prozess der Aushandlung involviert?
- Was wird ausgehandelt? Parameterwerte eines Dienstes
- Aushandlung könnte mehrere Iterationen benötigen
 - Besser: Intervall oder Liste diskreter akzeptabler Werte vorgeben, Antwort liefert tatsächlich reservierte Werte

■ Ansätze

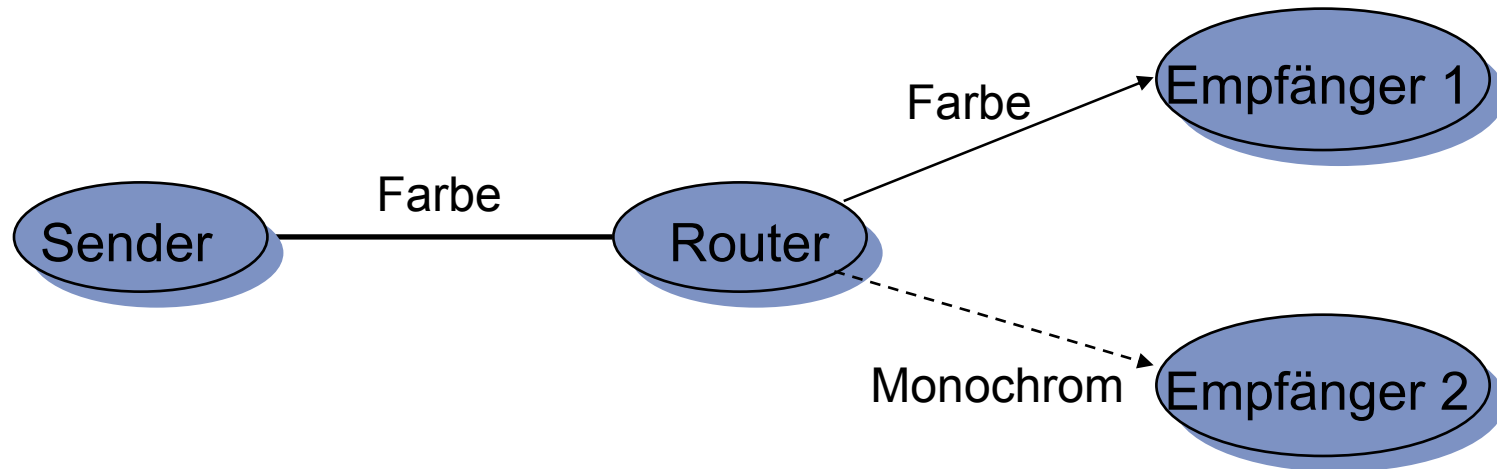
- Reservierungen auf Anforderung, d.h. „On-Demand“
- Langzeit-Reservierungen
 - Statische Reservierung, z.B. durch manuelles Einrichten
- Vorab-Reservierungen
 - Reservierung erfolgt vor eigentlicher Nutzung

Dienstgüteaushandlung bei Multicast



- Bei mehreren Empfängern kann es während der Dienstgüteaushandlung zu Konflikten kommen
- Konfliktauflösung beim Sender gemäß der gewählten Gruppensemantik
 - Abweisung des Verbindungsaufbauwunsches
 - Ablehnung eines einzelnen Empfängers
 - Aufbau der Multicast-Verbindung mit unterschiedlicher Dienstgüte (erfordert entsprechende Unterstützung des Kommunikationssystems)

Unterschiedliche Dienstgüten bei Multicast



- Filtern eines Datenstromes in Zwischensystemen ermöglicht die Bereitstellung unterschiedlicher Dienstgüten für eine Multicast-Verbindung
- Das **Filtern** erfordert eine **hierarchische Kodierung** der Benutzerdaten, z.B.:
 - MPEG-kodierte Videoströme (MPEG-2)
 - Trennung von Film- und zusätzlicher Farb-Information
 - Filtern ermöglicht die effiziente Nutzung von Betriebsmitteln durch eine „maßgeschneiderte“ Reservierung
 - Filtern ermöglicht Konfliktauflösung bei Dienstgüteaushandlung

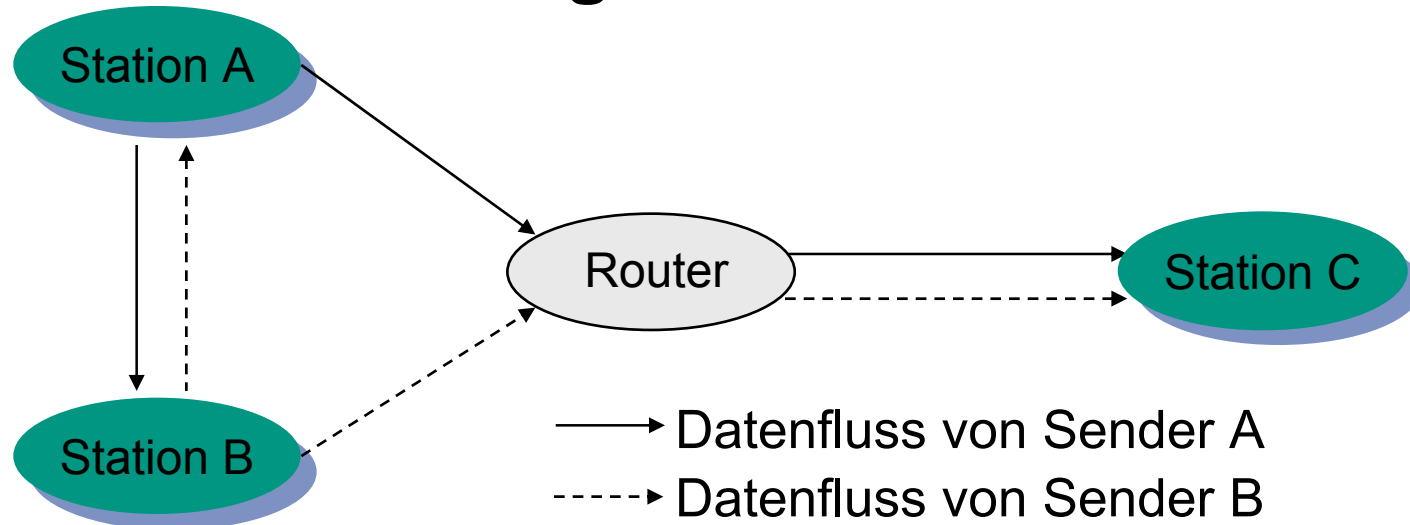
Layered Multicast

- Hierarchische Codierung des Datenstroms als Grundvoraussetzung
- Aufteilung des Datenstroms in mehrere Schichten, die sukzessive hinzugenommen werden können, um eine bessere Qualität zu erhalten, z.B. Basisschicht sowie 2 weitere Schichten zur Erhöhung der Auflösung (beispielsweise räumlich oder zeitlich)
- Je Schicht eine separate Multicast-Gruppe, d.h. Multicast-Datenstrom
- Empfänger mit geringerer Bandbreitenanforderung abonnieren nur wenige Datenströme
- Damit auch empfängerbasierte Staukontrolle möglich



[McJV96]

Gemeinsame Nutzung von Ressourcen



- Abhängig von der Kommunikationsbeziehung zwischen den Teilnehmern können reservierte Ressourcen gemeinsam genutzt werden
- Beispiel:
 - Audio-Konferenz: zu einem Zeitpunkt ist meist nur ein Sprecher aktiv
 - Ressourcen werden nur einmal reserviert, können jedoch von allen Sprechern genutzt werden

4.5 Dienstgütearchitekturen im Internet (1)

- Einführung von Dienstgütemechanismen schwierig und teuer
- Probleme:
 - Änderung der Router im Internet → teuer
 - Reservierung von Netzwerkressourcen → aufwändig
 - Abrechnung der genutzten Dienste → unklar
 - Architektur muss skalierbar sein → unklar
 - Änderung der Protokolle in allen Endsystemen → „unmöglich“
- Aber:
 - bessere Dienste für den Kunden
 - (und mehr Geld für Provider)
 - Besserer Schutz gegen Angriffe

Dienstgütearchitekturen im Internet (2)

■ Bisher mehrere Ansätze:

■ IP Type of Service


■ Integrated Services

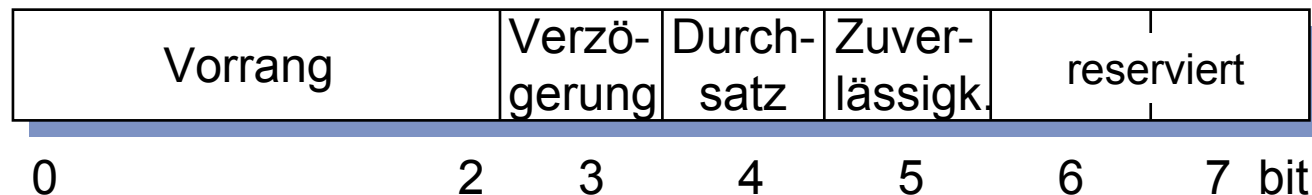
- Drei festgelegte Dienste: Guaranteed Service, Controlled Load, Best-Effort
- Empfängerorientiertes Signalisierungsprotokoll RSVP
- Konzept für Multicast
- Soft-State für Reservierungsinformation

■ Differentiated Services

- Unterstützung für max. 64 verschiedene Dienstklassen
- Implizite Signalisierung der Dienstklasse in der Dateneinheit
- Heute einsetzbar, aber tatsächlich erreichbare Dienstgüte und Managementarchitektur unklar

4.5.1 Dienstgüte durch Type of Service-Feld

- Dienstqualität in IP-Datagrammen ursprünglich durch Type-of-Service vorgesehen:  [RFC791]
 - **Vorrang**: verschiedene Prioritätsstufen
 - **Verzögerung**: normal (0), niedrig (1)
 - **Durchsatz**: normal (0), hoch (1)
 - **Zuverlässigkeit**: normal (0), hoch (1)



RFC 791: "The use of the Delay, Throughput and Reliability indications may increase the cost (in some sense) of the service. In many networks better performance for one of these parameters is coupled with worse performance on another."

- Es können mit dem Type of Service-Feld alleine keine Garantien gegeben werden. Weshalb?

4.5.2 Integrated Services Architecture

- Architektur für Integrierte Dienste (Mitte der 90er Jahre, RFC 1633):
 - Unterstützung multimedialer Anwendungen, z.B. Video-Konferenzen
 - Abkehr vom zustandslosen Router: **Zustand je Datenstrom in jedem Router**
 - besondere Behandlung der Pakete wie in Verkehrsprofil abgelegt
 - Erhalten der Robustheit durch „**Soft State**“-Reservierung
 - Ergänzung der bestehenden Internet-Architektur
 - Integration gruppenkommunikationsbasierter Anwendungen

Das Signalisierungsprotokoll RSVP

■ Ziel

- Signalisierung von Reservierungsanforderungen in IP-basierten Netzen
- Datentransfer findet weiterhin über IP statt

■ Konzept [RFC2205]

- Empfängerbasierte Signalisierung von Reservierungsanforderungen für unidirektionalen Datenfluss
- Unterstützung
 - Multicast-Kommunikation
 - heterogener Dienstqualität, d.h. unterschiedliche Empfänger können verschiedene Dienstqualität erfahren
 - verschiedener Reservierungsstile
- Empfänger erhält keine positive Quittung für Reservierung

Grundlegende Konzepte

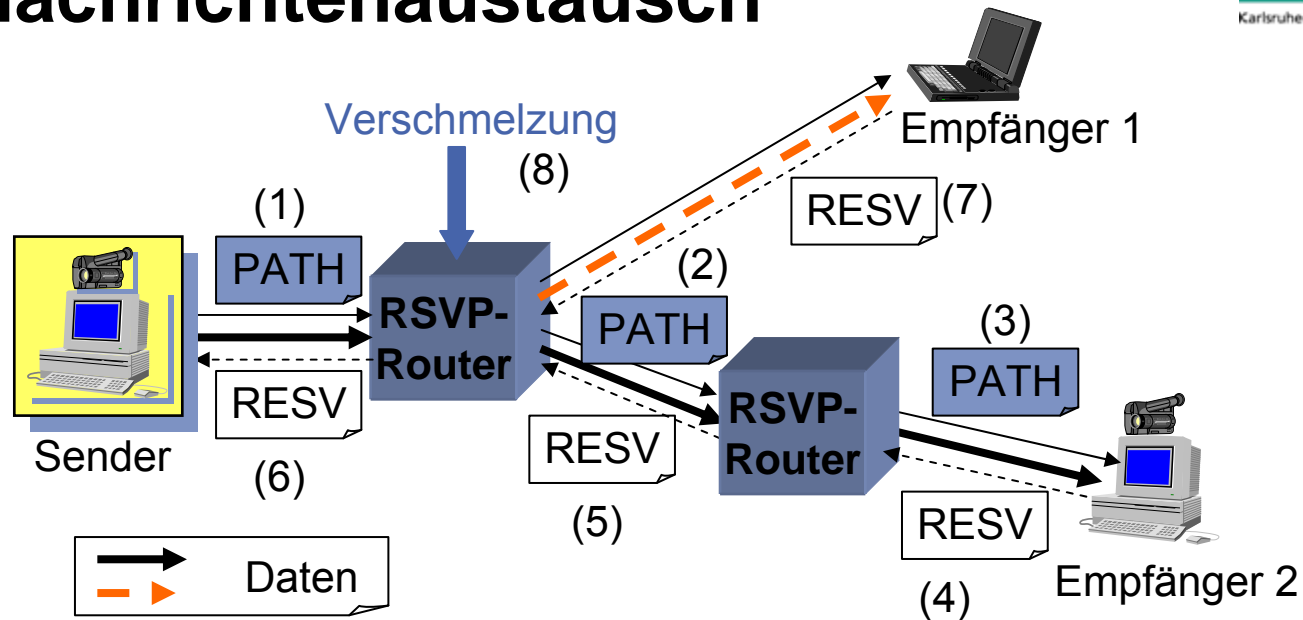
■ Session

- eine Menge von Datenströmen mit dem gleichen Ziel (Multipeer)
- Definiert durch: Ziel-IP-Adresse, Ziel-Protokoll-Kennung, Ziel-Port

■ Reservierungsanforderung

- Wird vom Empfänger (Ziel) gesendet und als **Flow Descriptor** bezeichnet. Sie besteht aus:
 - **FlowSpec**
 - Beschreibt die gewünschte Dienstqualität
 - Der Inhalt der FlowSpec ist nicht Gegenstand von RSVP
 - **FilterSpec**
 - Beschreibt, welche Dateneinheiten einer Session die Reservierungen nutzen dürfen
- Unterscheidung in FlowSpec u. FilterSpec ermöglicht explizite Trennung von Reservierung und Nutzung der Ressourcen

RSVP: Nachrichtenaustausch



- **PATH**-Dateneinheiten werden vom Sender verschickt. Verteilbaum wird dadurch aufgebaut (Soft-State), (1)–(3)
- **RESV**-Dateneinheiten (4)–(6) der Empfänger finden Weg zurück zur Quelle aufgrund der durch PATH-Nachrichten installierten Zustände
- Verschmelzung verschiedener Reservierungswünsche möglich

Probleme der Integrated Services Architektur

■ Mangelnde Skalierbarkeit [RFC2208]

- Jeder Router verwaltet Zustandsinformationen (Qualitätsparameter, Timer, Sender- und Empfängeradressen) pro Datenstrom
 - Leistung des Routers sinkt bei großer Anzahl von Datenströmen mit Reservierungen
- Paketweiterleitung (Forwarding) wird durch die Klassifizierung eines jeden Pakets komplexer
 - großer Aufwand in Hochleistungsnetzwerken (Anzahl der Pakete zur Weiterleitung und Anzahl der Reservierungen hoch)

■ Qualitätsparameter je Datenstrom frei wählbar

- Router muss eine sich dynamisch ändernde Anzahl von Dienstgüten unterstützen
 - Paket-Scheduling ist komplex und daher nicht so leistungsstark

4.5.3 Differentiated Services

- 1997 erste Vorschläge, um skalierbar Dienstgüte im Internet bereitzustellen
 - Diskussion von Differentiated Services in der IETF
 - Prinzip: Keep It Simple, Stupid (KISS)

Ähnliche Ansätze
von Dave Clark
und
Van Jacobson



David Clark




Van Jacobson

- „A Two-Bit Architecture“  [RFC2638]
- Anfang 1998: Arbeitsgruppe „Differentiated Services“ in der IETF (bis 3/2003)
 - Ziel: nur Basismechanismen (Building Blocks) definieren, aber keine Dienste

DiffServ – Ziele

- Qualitativ bessere, anwendungsunabhängige Dienste mittels einfacher, skalierbarer Mechanismen
- **Reduktion der Komplexität** im Netzinern
 - weniger Zustände
 - weniger Funktionalität
 - Vermeidung von Zuständen je Datenstrom
- Kompatibilität zu existierenden Anwendungen und Endsystemen (schnelle Etablierung)

DiffServ-Architektur – Konzepte (1)

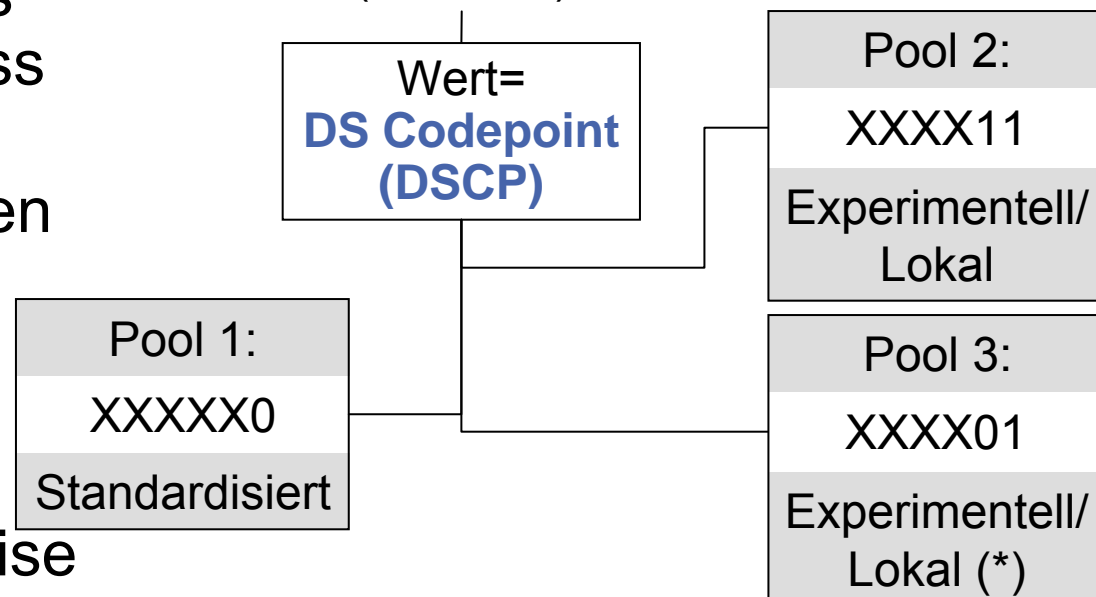
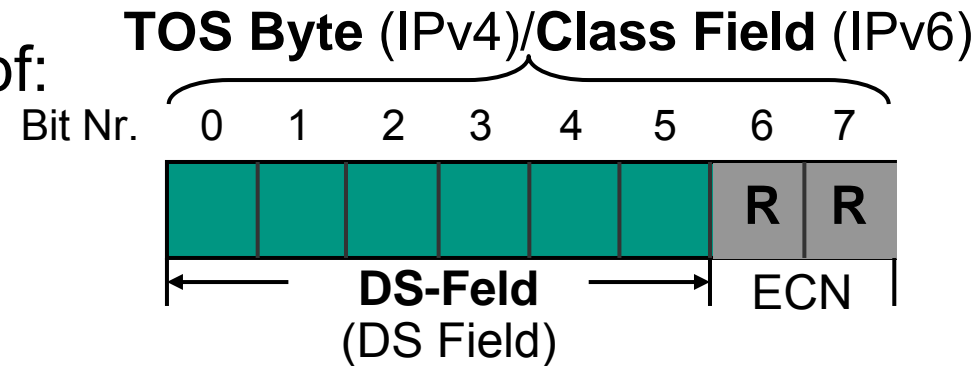
- Reduktion der Komplexität durch  [RFC2475]
 - Aggregation des Verkehrs im Netzzinnern zu **Dienstklassen** (Pakete gleicher Dienstklasse tragen entsprechende Markierung)
 - Vermeidung von Zuständen per **Ende-zu-Ende-Datenstrom** („Micro-Flow“) oder per Benutzer im Netzzinnern (aggregierte Klassifikationszustände, einfache Paketklassifizierer)
 - „Komplexere Funktionen“ wie Klassifizierung, Markierung, Überprüfung finden nur noch an Netzgrenzen statt
- Separation der eigentlichen **Weiterleitungsmechanismen** im Datenpfad von zugehörigen **Verwaltungsmechanismen** (z.B. Konfiguration, Ressourcenmanagement und -reservierung)

DiffServ-Architektur – Konzepte (2)

- Abschnittweises Weiterleitungsverhalten (Per-Hop-Behavior – PHB)
 - bestimmt Paketbehandlung für jedes Paket innerhalb eines Knotens
 - Implementierung durch Warteschlangenmechanismen und Scheduling
- Kennzeichnung des PHB im IP-Paketkopf anhand DS Codepoint (DSCP) → einfache Klassifikation
- Verfeinerung des ungenauen ToS-Ansatzes:
 - Explizite Rolle von Grenzknoten und verkehrsbeeinflussenden Mechanismen
 - PHB-Modell ist flexibler als relative Prioritäten oder Dienstmarkierungen

DiffServ Codepoints

- Im IPv4- bzw. IPv6-Kopf:
- DS-Feld ist prinzipiell **unstrukturiert!**
- Wert des Codepoints ordnet entsprechendes PHB eindeutig zu (muss konfigurierbar sein)
- Mehrere DSCPs können auf das gleiche PHB verweisen
- Es kann **mehr PHBs als DSCPs** geben
- DSCP hat normalerweise keinen Einfluss auf die Wegewahl

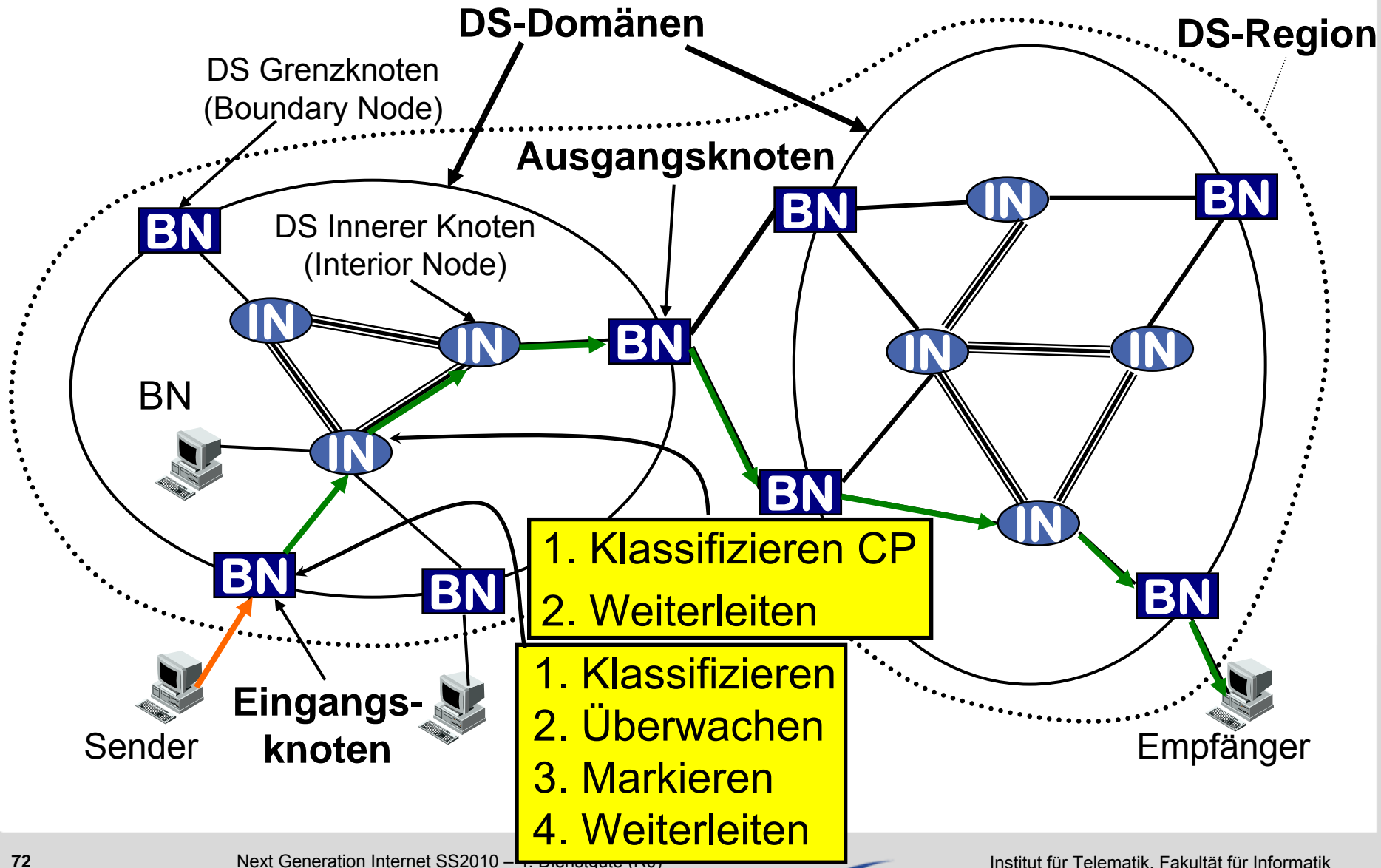


(*): Bereich kann bei Bedarf zur Standardisierung weiterer Werte herangezogen werden

ECN: Explicit Congestion Notification [RFC 3168]



Differentiated Services – Überblick



DiffServ – Ressourcenverwaltung (1)

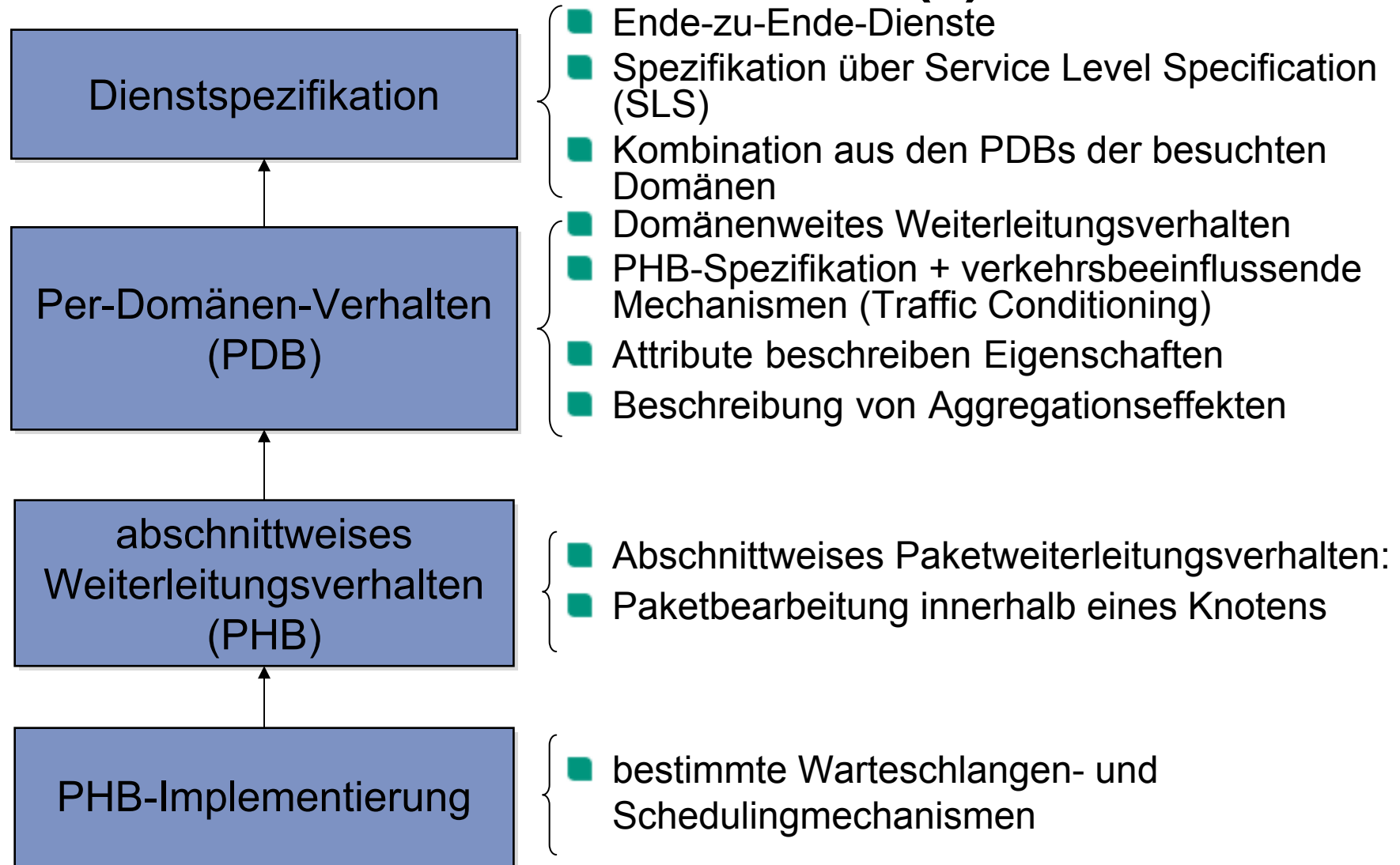
■ Ressourcennutzung

- Jedes DiffServ-Paket belegt die Ressourcen eines Knotens durch das zugeordnete jeweilige PHB
- **Überwachung** einzelner Ende-zu-Ende-Datenströme erfolgt üblicherweise **nur im ersten Grenzknoten**

■ Ressourcenreservierungen

- Üblicherweise per Aggregat
- Manuell/statisch → dynamisch
- Initiale, eher statische **Aufteilung** der Ressourcen **zwischen** den verschiedenen **PHBs** („**Ressourcenpartitionierung**“) für Knoten
- Über belegte/freie Ressourcen müssen nicht die Knoten selbst Buch führen (vgl. dazu IntServ?)
- Installation des Profils aktiviert Dienstnutzung

Differentiated Services – Dienste (1)



Differentiated Services – Dienste (2)

■ Dienstleistungsvereinbarung (**Service Level Agreement – SLA**):

- Dienstvertrag zwischen Dienstkunde und Dienstbetreiber, der den Weiterleitungsdienst festlegt, welchen der Kunde erhalten sollte
- beinhaltet u.a. Technical Service Level Specification
- enthalten darüber hinaus noch nicht-technische, d.h. geschäftliche bzw. wirtschaftliche Aspekte (z.B. Bedingungen für Regresszahlungen)



■ Komponenten zur Realisierung eines Dienstes:

- **PHB-Implementierungen**
(konkrete Realisierung eines PHBs)
- **Verkehrsbeeinflussungsmaßnahmen**
(Traffic Conditioning)
- **Bereitstellungsstrategien und Abrechnungsmodelle**

Differentiated Services – Dienste (3)


- Bilaterale Dienstverträge (SLAs) zwischen benachbarten Internet-Dienstbetreibern (Internet Service Providers)
 - statisch oder dynamisch
- Technische Dienstspezifikation (TSLs):
 - Leistungsparameter, z.B.: Durchsatz, Verzögerung, Verlust
 - Topologischer Gültigkeitsbereich
 - Verkehrsprofil (zeitl. Eigenschaften, z.B. Rate und Burst)
 - Markierungs- und Verkehrsformungsverhalten
 - Zusätzlich allgemeine Parameter, wie u.a.:
 - Verfügbarkeit
 - Zuverlässigkeit
 - Wegebeschränkungen

DiffServ-Bausteine: PHBs

- **Per-Hop Behavior (PHB):**  [RFC2474]
 - extern sichtbares Verhalten von Paketen (eines BA) in einem DS-Knoten
 - bestimmt Behandlung von Paketen anhand DSCP während der Weiterleitung innerhalb eines DS-Knotens
- **Bisher standardisierte PHBs:**
 - „**Default PHB**“ ist das bisherige Best-Effort-Verhalten (DSCP=000000)
 - **Class Selector PHBs:**  [RFC2474]
 - kompatibel zu IP-Precedence-Feld im ToS-Byte (DSCPs=XXX000)
 - größerer Wert = höhere Priorität
→ 8 relative Prioritätsklassen (inkl. Default-PHB)
 - **Expedited Forwarding PHB**
 - **Assured Forwarding PHB**

Expedited Forwarding PHB [RFC3246]

- Ausgangspunkt: Idee der virtuellen Standleitung (garantierte Bandbreite, kein Jitter)
- Expedited Forwarding PHB als Basis für Dienste mit
 - garantierter Bandbreite, niedrigem u. begrenztem Jitter, geringer Verzögerung, geringen Paketverlusten
- Ziel
 - ankommende Pakete „sehen nur leere Warteschlangen“
 - Paket verlässt den Router „sofort“ wieder → kein Stau
- Summe der Ankunftsrate muss kleiner als die minimale Ausgangsrate sein → **Zugangskontrolle notwendig** (warum?)

- Implementierung z.B. durch Simple Priority Queueing oder WFQ
 - WFQ-Variante weist leicht höheren Jitter auf als Simple Priority Queue Implementierung (weshalb?)
 - Aber WFQ bietet niedrigeren Klassen Schutz vor permanenter Verdrängung
- Ursprüngliche Spezifikation in RFC 2598 ungenau, daher Neudefinition in RFC 3246 (weitere Erläuterungen dazu in RFC 3247)  [RFC3247]
 - ursprüngliche Formulierung war zwar genau genommen mathematisch nicht korrekt und daher nicht erfüllbar

EF PHB – Formale Definition

- EF-Knoten an einer **Schnittstelle** I mit konfigurierter Rate R muss folgende Gleichungen erfüllen
 - (1) $d_j \leq f_j + E_a \quad \forall j > 0$
 - (2) $f_0 = 0, d_0 = 0 \quad f_j = \max(a_j, \min(d_{j-1}, f_{j-1})) + \frac{l_j}{R} \quad \forall j > 0$
- d_j Zeit zu der das letzte Bit des j . zu verschickenden EF-Pakets den Knoten über I tatsächlich verlässt
- f_j **Zielabgangszeit** für das j . EF-Paket über I , d.h. die „Ideal-Zeit“ zu der (oder zuvor) das letzte Bit des Pakets den Knoten verlassen sollte
- a_j **tatsächliche Ankunftszeit** des letzten Bits des j . EF-Pakets am Knoten mit Zielausgang I
- l_j Größe (in Bits) des j . EF-Pakets, das über I verschickt werden soll, bezieht sich auf das IP-Datagramm (IP-Kopf plus Nutzlast) und enthält keinen darunter liegenden (z.B. MAC-Schicht) Overhead.
- R ist die für EF konfigurierte Rate am Ausgang I (in bits/s)
- E_a Fehlerterm, obere Grenze für $d_j - f_j$

Assured Forwarding PHB (1)

Assured Forwarding PHB Group [RFC2597]

- Definiert Eigenschaften für „AF-Typ“
- Besser geeignet als EF für burst-artigen Verkehr

AF-Klasse:

- Instanz eines AF-Typs
- Gruppe aus m PHBs mit je unterschiedlicher Verwurfspriorität
- Vorgeschlagen: m=3, d.h. Verwurfsprioritäten niedrig, mittel, hoch; mindestens 2 unterschiedliche Prioritäten je AF-Klasse gefordert
- keine Umordnung von Paketen eines Micro-Flows zugelassen
- vollständig unabhängig von anderen AF-Klassen

n unabhängige AF-Klassen (4 mit standardisierten DSCPs)

AF-Klasse

Verwurfspriorität (y)	AF 1y	AF 2y	AF 3y	AF 4y
Niedrig (1) ●	001010	010010	011010	100010
Mittel (2) ●	001100	010100	011100	100100
Hoch (3) ●	001110	010110	011110	100110

Assured Forwarding PHB (2)

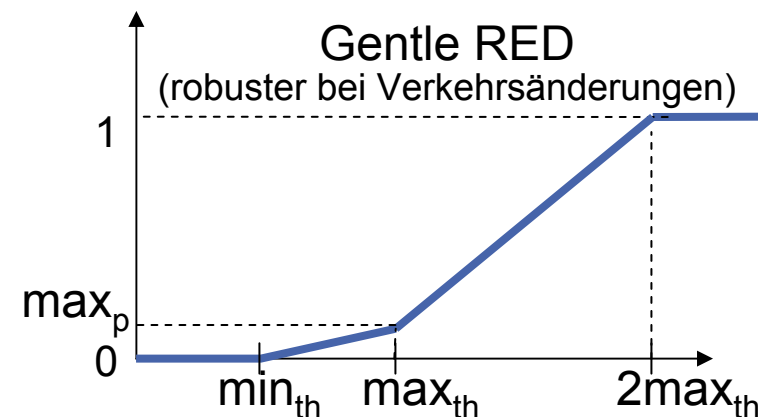
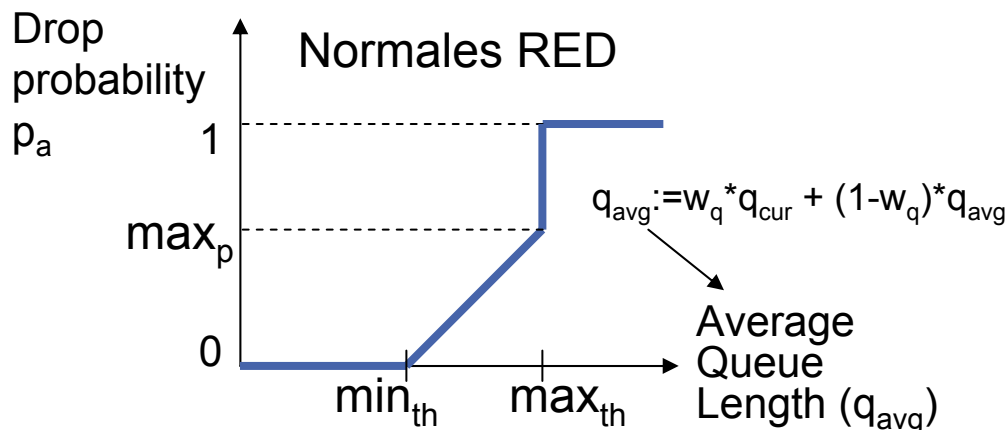
- Aktives Warteschlangenmanagement (z.B. RED) gefordert
- Verwurfswahrscheinlichkeiten für PHBs müssen zwischen Betreibern aus Konsistenzgründen abgestimmt werden (noch offener Punkt)
- Basis für Dienste mit
 - zugesicherter Bandbreite unterhalb vereinbarter Senderate (Garantie)
 - Nutzung weiterer Bandbreite, falls diese verfügbar
 - bevorzugtem Verwerfen von Paketen oberhalb der vereinbarten Senderate bei Ressourcenmangel
 - stoßartigem Verkehr, längere Bursts haben höhere Verwurfswahrscheinlichkeit

Assured Forwarding PHB (3)

- Oberhalb zugesicherter Rate (gelb):
 - statistisches Multiplexing zwischen Flows
 - Flows sollten sich diese Bandbreite fair teilen
 - Protokolle mit Staukontrolle gefragt: TCP, SCTP, DCCP
- Praktisches Problem: Pakete mit höherer Verwurfswahrscheinlichkeit (z.B. gelb/rot) beeinflussen solche mit niedriger (grüne), da in gleicher Warteschlange einsortiert. Auswirkungen?

Einschub: Active Queue Management

- Ursprung: normales Tail-Drop ruft teilweise Unfairness und Synchronisation von TCP-Strömen hervor  [RFC2309]
- Idee: Aktives Warteschlangenmanagement verhindert, das WS in Überlastbereich kommt
 - frühzeitiges Verwerfen von Paketen, bevor WS maximal gefüllt: Warteschlangenlänge wird kürzer
 - **Random Early Detection (RED)**  [FIJa93]




Einschub AQM – Diskussion

■ Vorteil

- Zufällige Auswahl sorgt für mehr Fairness
- Durchschnittliche Warteschlangenlänge ist kürzer
- Stausituationen können frühzeitig erkannt werden → ECN

■ Nachteile

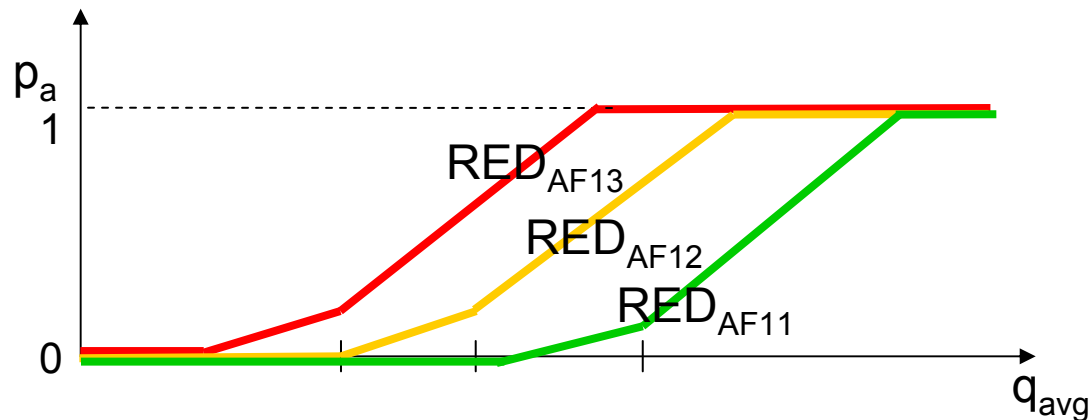
- Aufwändigerer Warteschlangenmechanismus
- RED-Parameter sind nicht einfach zu setzen, d.h. stark anwendungsabhängig, welche Parameter jeweils gute Leistung liefern  [LAJS03]

- Zahlreiche Varianten existieren: u.a. **Gentle RED**, **Adaptive RED** (ARED), usw.

- Alternative AQM-Ansätze bedienen sich teilweise d. Verfahren aus der Regelungstechnik  [ZhLW03]

Realisierung des AF PHB mit RED

- Jeder Verwurfspriorität („Farbe“) werden andere RED-Parameter zugeordnet → Weighted RED

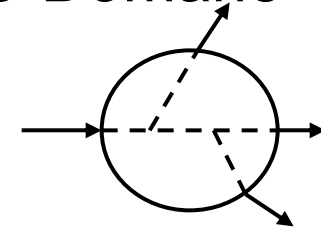


- Warteschlangenmanagement erhält Reihenfolge innerhalb einer AF-Klasse (nicht einfach möglich für mehrere Warteschlangen)

DiffServ-Bausteine: PDBs

■ Per Domain Behavior (PDB): [RFC3086]

- Ein Per-Domain-Behavior beschreibt das Verhalten, das Pakete auf ihrem Weg durch eine DS-Domäne erfahren:
 - beliebiger Eingang zu beliebigem Ausgang (One-to-One)
 - beliebiger Eingang zu jedem Ausgang (One-to-Any)
 - beliebiger Eingang zu bestimmten Ausgängen (One-to-Few)
- Regeln zur Verkehrsbeeinflussung (**Traffic Conditioning**) und zu verwendende **PHBs** müssen angegeben werden
- Berücksichtigung der **Aggregationseffekte**
- Attribute, die PDB charakterisieren, messbar und quantifizierbar
- „Vorstufe“ eines Dienstes



PDBs: Virtual Wire

- Nachbildung dedizierter physikalischer Leitung bestimmter Bandbreite
- Attribute:
 - garantierter Durchsatz
 - obere Schranke für Delay-Jitter
- Verkehrsbeeinflussung:
 - Strikte Überwachung (Policing) am Eingang (nicht-konforme Pakete verwerfen)
 - Shaping am Ausgang einer Domäne
- Ziel: Einhalten eines so genannten **Jitter-Window**, so dass obere Jitter-Schranke garantiert wird
- Vorgehen: Verzögern des ersten Pakets um maximales Jitter-Window → Zeit zum Ausgleichen von Schwankungen wird gewonnen, aber: zusätzliche u.U. unnötige Verzögerung der Pakete
- basiert auf alter EF-PHB-Definition, daher überholt


PDBs: Assured Rate (1)

- **Zugesicherte Rate** wenn Sender bis zu dieser Rate (**Committed Information Rate – CIR**) sendet
- **Attribute:**
 - zugesicherte Rate
 - niedrige Verwurfswahrscheinlichkeit für ratenkonformen Verkehr
- keine Zusicherung von Delay- oder Jitter-Schranken
- Ausnutzung weiterer verfügbarer Bandbreite möglich (aber nicht garantiert)
- Zugesicherte Rate wird unabhängig vom Verkehrstyp (z.B. UDP/TCP) erbracht, Trennung von UDP/TCP möglich
- Nutzung einer AF-Klasse

PDBs: Assured Rate (2)

- Ermittlung und Anpassung an zusätzlich nutzbaren Bandbreitenanteil gut mit TCP möglich
- Verkehrsbeeinflussung:
 - Klassifizierung, Messen (genaue Methode muss spezifiziert werden) und Markieren am Eingang
 - Traffic Shaping am Eingang nicht notwendig
- Messmethoden für zugesicherte Rate „CIR“:
 - Committed Burst Size (CBS) über Zeitintervall T1
 - Peak Information Rate (PIR), Peak Burst Size (PBS), Excess Burst Size (EBS), ggf. weiteres Zeitintervall T2
- Markierung als „grün“, „gelb“ und „rot“:
 - Verwerfen **roter** Pakete am Eingang durch Überwacher (Policer) möglich
 - Verwerfen **gelber** Pakete erfolgt hingegen nicht durch Eingangsüberwachung (Ingress-Policer), sondern nur durch PHB-Mechanismus
 - Verwerfen **grüner** Pakete erst nachdem alle gelben und roten Pakete verworfen wurden

PDBs: Lower Effort (LE)

- RFC 3662 (Bless, Wehrle)  [RFC3662]
- Grundidee: **Verkehrsklasse mit niedrigerer Priorität als Best-Effort** zum Schutz der Best-Effort-Klasse (Q-Bone: Non-Elevated Services)
- Bei zunehmender Last wird LE von BE aus dem Netz verdrängt → prima, um Schutzkapazität zu füllen
- Im Unterschied zu normalem Best-Effort, darf Lower-Effort-Verkehr auch „verhungern“, d.h. es können sämtliche Pakete verworfen werden
- Alternativ kann auch eine Mindestbandbreite für das LE PDB vorgehalten werden



PDBs: Lower Effort (2)

■ Anwendung

- Werkzeug für Netzadmins, um Auswirkungen auf BE zu begrenzen, ohne Verkehr komplett aus dem Netz zu verbannen (z.B. File-Sharing etc.), „Penalty-Box“
- aber: nicht als Ersatz für das Verwerfen bei unautorisiertem Verkehr gedacht
- Ähnlich wie Hintergrundpriorität für Prozesse (nice)

■ Extrem einfach zu konfigurieren (zwei Warteschlangen BE/LE, Strict Priority Scheduling zwischen diesen beiden)

- Class Selector PHB 1 mit niedrigerer Priorität oder AF PHB
- Kein Traffic Conditioning notwendig

Beispieldienste

■ Anwendung **Assured-Rate**-basierter Dienste

- Datenübertragung mit zeitlicher Obergrenze
- Video-Playback: variable Bitrate, Jitter-Kompensation möglich
- Kopplung virtueller privater Netze mit Mindestgarantie


■ Anwendung **Virtual-Wire**-basierter Dienste

- Kopplung virtueller privater Netze mit Qualität einer dedizierten Standleitung
- Voice over IP
- Interaktive (Audio/Video-) Anwendungen

Betreiberaspekte

- Notwendige Schritte beim Betreiber:
 - Auswahl der PHBs, ihrer Implementierung und deren Parameter, Verhältnis zu anderen PHBs festlegen (**Ressourcen-Partitionierung**)
 - Auswahl konkreter „Traffic Conditioning“-Mechanismen
 - SLA und „Traffic Conditioning Agreement“ mit anderen Betreibern und Kunden abstimmen
 - Routerkonfiguration (PHBs und Profile)
 - Messung und Überwachung der erreichten Qualität

DiffServ-Architektur – Management

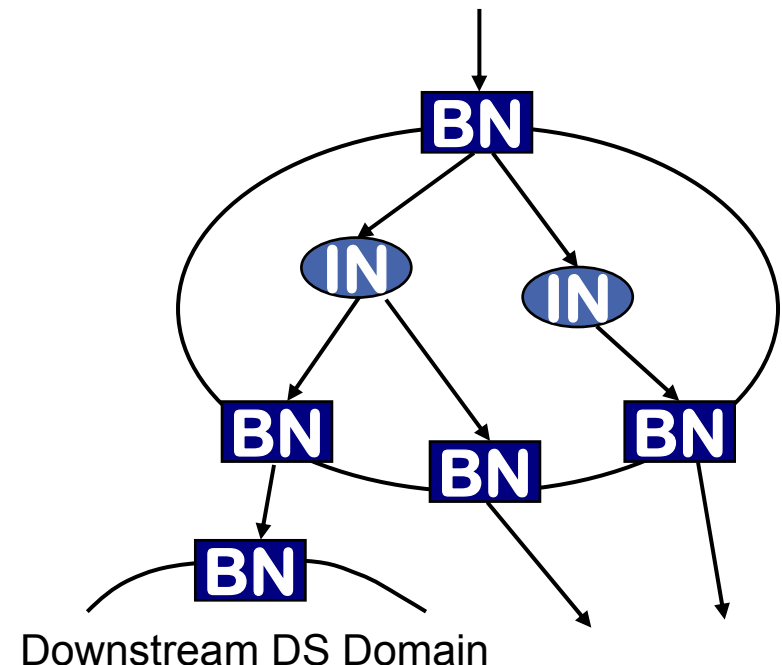
- Forschungsbedarf
- Beispiel „Virtual Wire“:
 - Zugangskontrolle notwendig
- **Dynamische Aushandlung** von SLAs
 - Kunde \leftrightarrow ISP, ISP \leftrightarrow ISP
 - Automatische Einrichtung/Anpassung der Profile
- Managementeinheit: **Bandwidth Broker**
(Van Jacobson)  [RFC2638]
 - Agent für das Bandbreitenmanagement pro Domäne
 - Bilaterale Dienstaushandlung (dynamische Dienstverträge)
 - Installation von Verkehrsprofilen in Grenz-Routern
 - Ursprünglich Einsatz im QBone (Internet-2) vorgesehen

DiffServ Management


- DiffServ Control Plane Elements (DCPEL)
 - Welche Komponenten sind sinnvoll für ein Management einzusetzen?
 - z.B. Kommunikation mit Routern zwecks Installation von Verkehrsprofilen
 - besteht Standardisierungsbedarf?
- **Signalisierung**: Nachfolger von RSVP in der IETF WG **NSIS** entwickelt
Ansatz: 2-schichtige Architektur
Transport/Signalisierungsanwendung
- **Skalierbarkeit der Kontrollebene** wichtig!
→ dazu Arbeiten am Institut

DiffServ-Architektur & Multicast

- DiffServ-Mechanismen sind orthogonal zu Multicast-Mechanismen
 - Im Datenpfad stellt dies kein Problem dar
 - Dienstgüten werden auch für Multicast-Datenströme erbracht



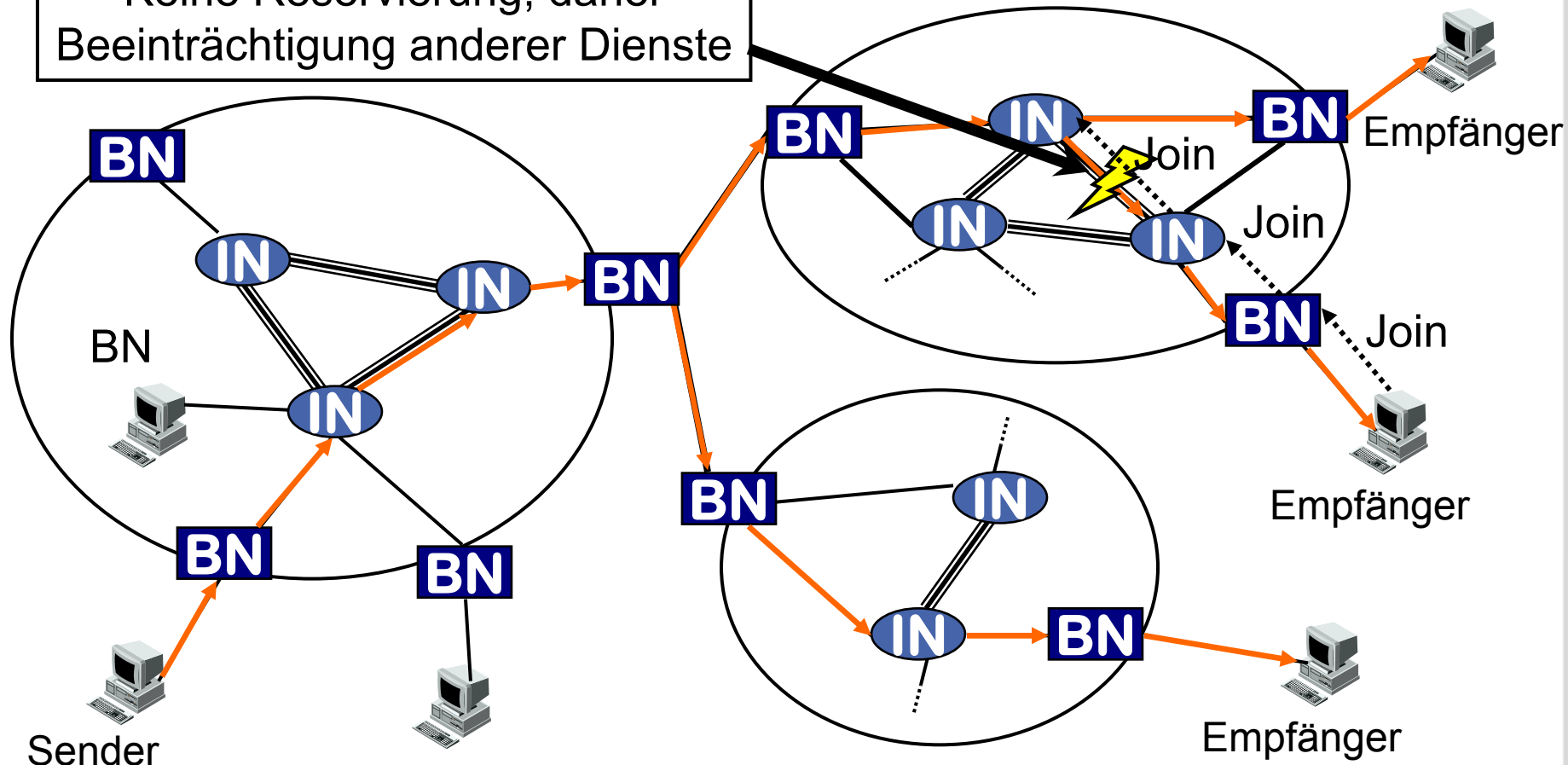
DiffServ & Multicast: Probleme (1)

- Probleme entstehen im Kontrollpfad durch die dynamische Gruppenzugehörigkeit  [RFC3754]
 - jeder kann sich einer Gruppe anschließen
 - es entstehen plötzlich „Verkehrsquellen“ im Netzzinnern
 - u.U. können andere Teilnehmer in der Qualität ihres Dienstes beeinträchtigt werden (NRS-Problem)
- Empfänger haben unterschiedliche Anforderungen
 - Wie können heterogene Gruppen unterstützt werden?
- Wechselnde Sender
 - Prinzipiell kann jeder Rechner Daten an eine Gruppe schicken
 - Welche erhalten Dienstgüte?

DiffServ & Multicast: Probleme (2)

■ Neglected Reservation Subtree Problem (NRS-Problem)

Keine Reservierung, daher
Beeinträchtigung anderer Dienste



NRS Problem

- NRS-Problem entsteht sobald neuer Zweig im Multicast-Baum entsteht, für den
 - ankommende Pakete für ein besseres PHB markiert wurden
 - noch keine Zugangskontrolle durchgeführt wurde
 - damit unkontrolliert Ressourcen verbraucht werden
- Das heißt aber auch sobald RSVP verwendet wird:
 - vor Reservierung mit RSVP muss der Empfänger PATH-Nachrichten empfangen
 - damit er RESV-Nachrichten schicken kann
 - PATH-Nachrichten werden per Multicast verteilt
 - der Empfänger muss der Multicast-Gruppe beitreten
 - NRS-Problem entsteht, falls PATH-Nachrichten nicht speziell gefiltert werden

DiffServ-Multicast Lösung

- Vorschlag zur Lösung des Problems durch das ITM
 - Ergänzen der Multicast-Routing-Tabelle um einen Eintrag (neuer Codepoint)
 - bei neuen Gruppenmitgliedern wird der Codepoint zunächst auf Best-Effort umgesetzt
 - nur bei Reservierung der Bandbreite wird der Codepoint des besseren Dienstes gesetzt
 - Diese Lösung ermöglicht die Verwendung von RSVP und erlaubt ein "Schnuppern" in eine Gruppe, d.h. Empfangen der Daten mit Best-Effort

Differentiated-Services vs. Integrated-Services

	Best-Effort	Integrated-Services	Differentiated Services
QoS-Garantie	keine	pro Datenstrom (Microflow)	für aggregierte Datenströme
Konfiguration	keine	pro Datenstrom bzw. Sitzung	pro Aggregat zwischen Domänen
Typ der Garantie	keine	individuell per Datenstrom	aggregiert
Dauer der Garantie	keine	kurzlebig (Sitzungsdauer)	langfristig(er)
Zustandshaltung	keine	pro Datenstrom	pro aggregierter Reservierung
Signalisierung	keine	RSVP	Keine, bzw. noch nicht definiert
Multicast-Unterstützung	IP-Multicast	empfänger-orientiert, heterogen	IP-Multicast, aber Ressourcen-Reservierungsprobleme

4.5.4 Dienstgütesignalisierung

- „**Signalisierung**“: Austausch von Daten zwischen Knoten, um Zustände in Netzknoten einzurichten, zu verwalten oder zu löschen

- Beispiele:
 - Klassisch: SS7 – Zeichengabe zur Leitungsvermittlung
 - Zugangskontrolle und Ressourcenreservierung für Dienstgüte (Quality-of-Service) → RSVP
 - Dynamische Konfiguration von Firewall-Löchern oder NAT Bindings
 - Dynamische Etablierung von Messpunkten

RSVP-Probleme

- RSVP hauptsächlich für Multicast ausgelegt
 - Per-Flow, Zusammenlegen von Reservierungen, ...
 - Einfacher Transport über IP oder UDP → Einschränkungen, z.B. Nachrichtengröße
 - Kaum Verbreitung von Multicast, Unicast dominiert
- RSVP-Erweiterungen zur Leistungssteigerung: Zuverlässigkeit, Summary Refreshes, Bündeln, usw.
- RSVP-TE für MPLS
- Fehlende Berücksichtigung mobiler Knoten


Next Steps in Signaling (NSIS)

Signalisierung im Internet:

Router, Ressourcen in der IP-Schicht

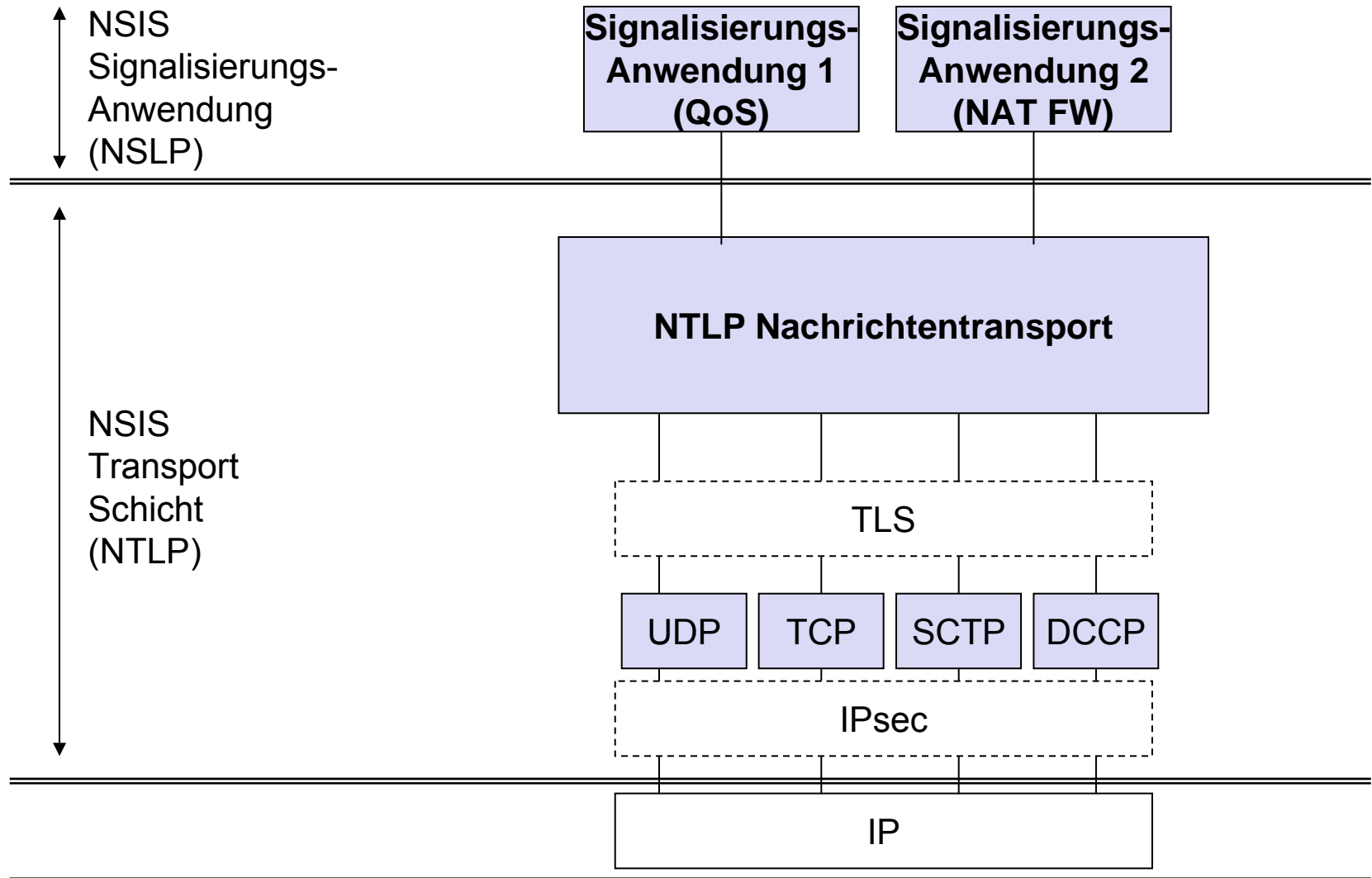
Ziele der IETF Arbeitsgruppe:

- Interworking zwischen verschiedenen QoS-Lösungen
- Unterstützung von Mobilität und drahtlosen Netzen der nächsten Generation
- Vereinfachte Lösung zur QoS-Signalisierung
- Wiederverwendung von Teilen existierender Lösungen, vorhandene IETF Signalisierungsprotokolle sollen als Basis dienen

- NSIS Requirements  [RFC3726]
- Aus RSVP-Erfahrung gelernt:
 - RSVP bietet keine effiziente Unterstützung von Unicast-Reservierungen
 - Einsatz in sehr unterschiedlichen Umgebungen, war bei Definition noch nicht vorhersehbar
- Weitere Annahmen:
 - **Pfad-gekoppelte Signalisierung:**
Signalisierungsnachrichten folgen Datenpfad
 - Normales Routing (kein QoS Routing oder Load Balancing etc.)
 - Keine Multicast(!)-Unterstützung



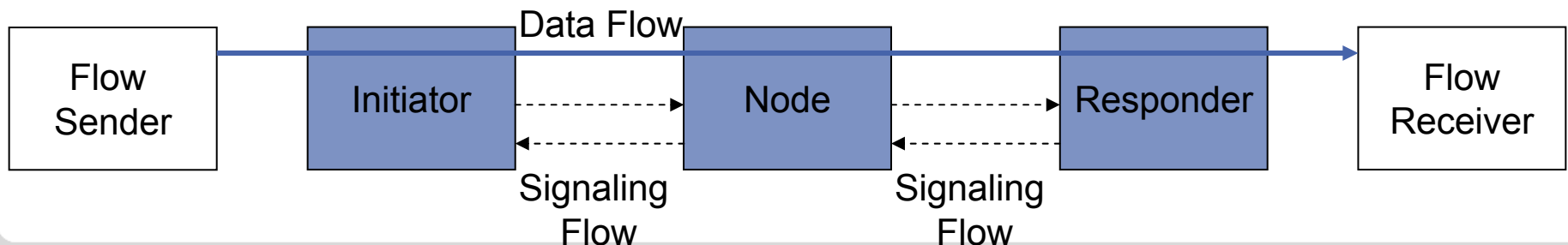
NSIS: Aufteilung in zwei Schichten



NTLP: GIST Eigenschaften

General Internet Signaling Transport (GIST) [GIST09]

- Gemeinsamer Signalisierungstransportdienst für verschiedene Signalisierungsanwendungen
- Einfacher Nachricht-für-Nachricht-Übertragungsdienst (Inhalt ist transparent für GIST)
- Aufspüren und Management von Routen für Signalisierungsnachrichten (Installation von NSIS-Routing-Information)
- Etablierung von Signalisierungsnachrichtenassoziationen (Soft-State)
- Einfacher Schutz gegen DoS beim Aufsetzen einer Assoziation



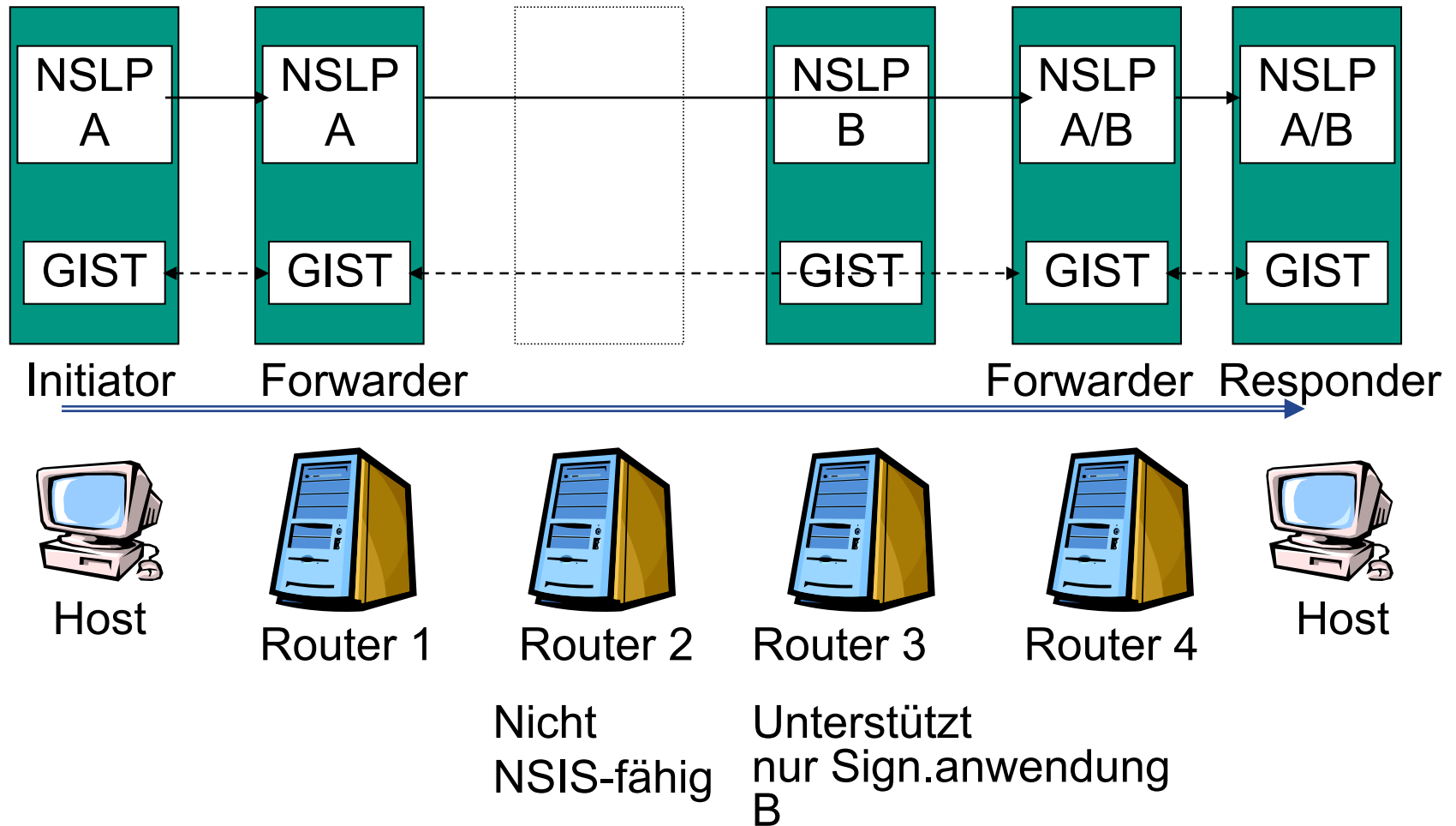
GIST-Merkmale

- Zwei Betriebsmodi (nicht sichtbar für die Anwendung)
 - Datagramm-Modus (UDP), Aufspüren der aktuellen Signalisierungsroute
 - Verbindungsmodus (TCP), längerfristige Beziehungen, z.B. Festnetzbetreiber
- Session Identifier
 - eindeutige ID (128 Bit), um Zustand zu kontrollieren
 - kann mehrere Datenströme umfassen, z.B. während Handover
- Message Routing Method (MRM) gibt an, wie das Signalisierungspaket geroutet werden soll, bzw. anhand welcher Kopffelder das Datenpaket geroutet wird

NTLP-Funktionen

- Datagramm-Modus (**D-Mode**) wird eingesetzt für
 - Aufspüren einer Signalisierungsrouten
 - Etablierung von Signalisierungsassoziationen
- Verbindungsorientierter Modus (**C-Mode**) wird verwendet wenn:
 - Fragmentierung erforderlich ist
 - Zuverlässiger Transport gefordert wird
 - Sicherheit erforderlich ist
- **Message Routing State Table** (pro Flow):
[Message Routing Information (MRI), Signaling Application Identification (NSLPID), Session Identification (SID)]
→ [Upstream Peer, Downstream Peer]
- **Messaging Association** (pro Partner):
bidirektional, Wiederverwendung


NTLP Szenario



NTLP-Nachrichten

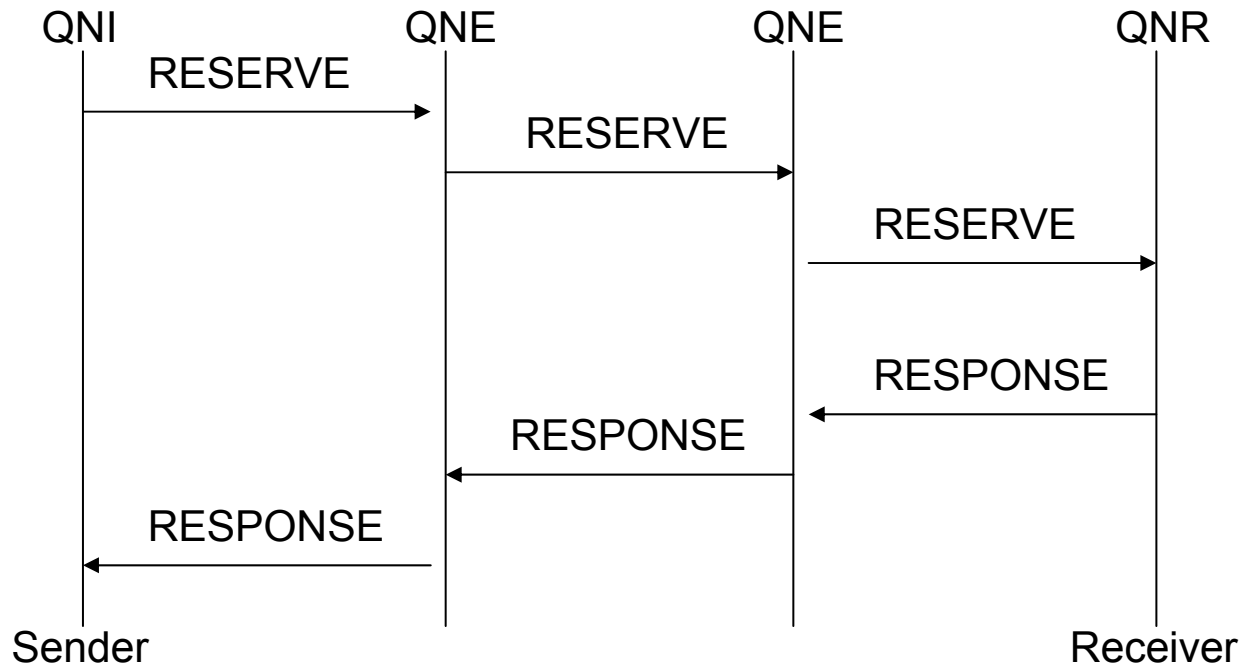
- Router Alert Option wird verwendet, d.h. NSIS-fähige Knoten erkennen und verarbeiten NSIS-Nachrichten → praktische Probleme mit RAO-Nutzung
- Gemeinsamer Kopf und TLV (Type-Length-Value) Kodierung
- Query/Response/(Confirm)
 - Auffinden einer Route
 - Mit optionalen Cookies beim Assoziationsaufbau
 - Zum Auffrischen von Zuständen
 - Aushandlung von Transportprotokollen
- MA Hello
 - Zum Aufrechterhalten von Messaging Associations

QoS NSLP

- Ähnliche Funktionalität wie RSVP (aber unterstützt nur Unicast)  [QoS-NSLP10]
- Unabhängigkeit von spezifischen QoS-Modellen wie IntServ oder DiffServ
- Nachrichten:
 - **RESERVE**: erzeugt, verändert oder löscht einen Reservierungszustand
 - **QUERY**: versuchsweise Anfrage (Probing)
 - **RESPONSE**: Antwort auf RESERVE oder QUERY

QoS NSLP Beispiel

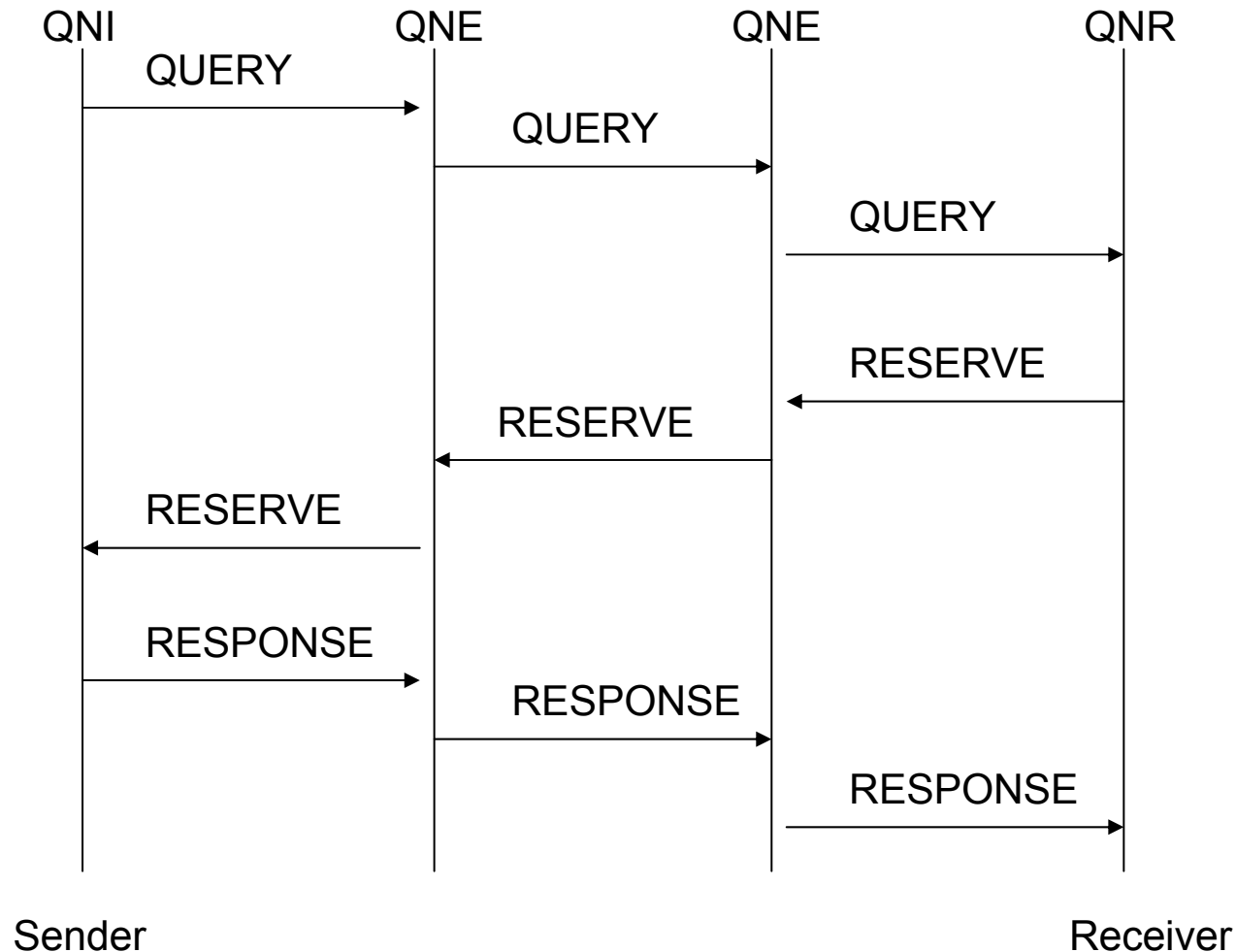
■ Sender-initiierte Reservierung:



QNI: QoS NSLP Initiator
QNE: QoS NSLP Entity
QNR: QoS NSLP Responder

QoS NSLP – Bsp. für empf.-initiierte Reservierung

■ Empfänger-initiierte Reservierung:



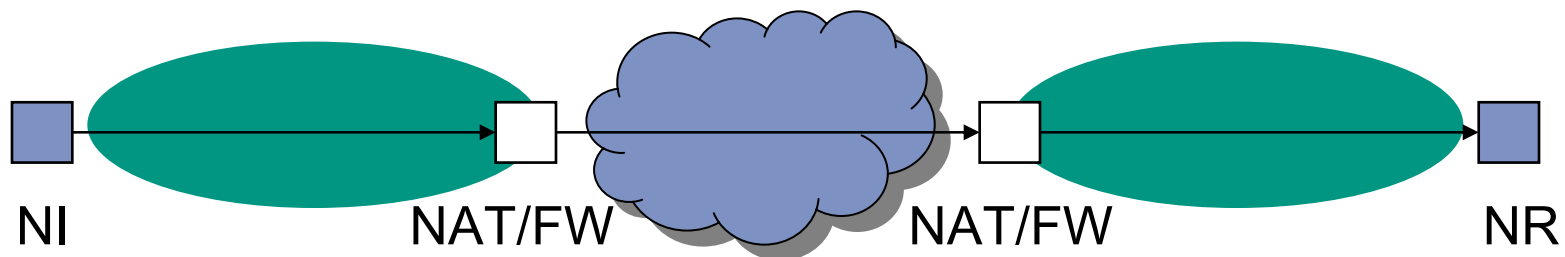
QoS-NSLP QSPEC Template

- Allgemeine Darstellung von QoS Parametern
- Verkehrsbeschreibung und einzuhaltende Grenzen
- Verkehrsbeschreibung durch **Traffic Model**
 - rate (r)
 - bucket size (b)
 - peak rate (p)
 - minimum policed unit (m)
- **Einzuhaltende Grenzen:**
<Path Latency>, <Path Jitter>, <Path PLR> und <Path PER>
 - PLR: Packet Loss Ratio
 - PER: Packet Error Ratio
- Einhaltung d. Grenzen kann durch beliebiges QoS Model realisiert werden



NAT/FW NSLP

- Firewalls und NAT-Gateways erschweren Kommunikation für zahlreiche Anwendungen
- Aufgabe NAT/FW NSLP: Dynamische Konfiguration von NATs und/oder Firewalls entlang des Datenpfads
 - Etablierung eines Bindings
 - Gezieltes Öffnen der Firewalls



NSIS-ka: Implementierung d. Uni Karlsruhe

Implementierte Protokolle:

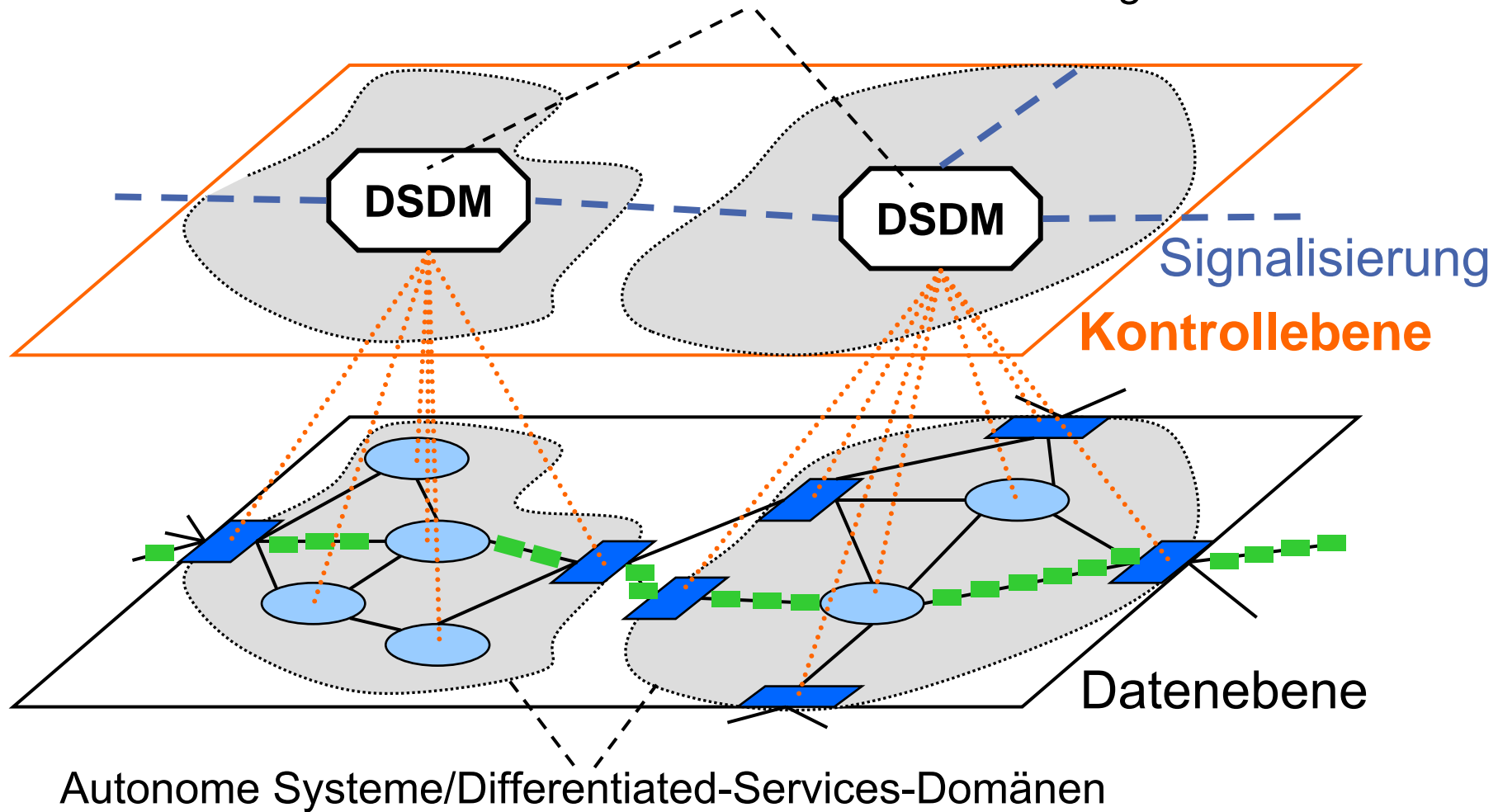
- GIST (UDP, TCP, TLS/TCP, SCTP)
- QoS NSLP + QSPEC Template
- NAT/FW NSLP
- C++-basiert
- Multi-threaded
- Linux
- Frei verfügbar <http://nsis-ka.org/>

4.5.5 Arbeiten am ITM

- DiffServ Management
- ScaleQoS: technikübergreifende QoS-Steuerung für heterogene Netze (Fixed-Mobile-Converged Networks)
- QoS in SpoVNets (Spontane Virtuelle Netze)
 - QoS vor allem auch für Gruppenkommunikation
- NSIS-Erweiterungen und -Implementierung
- QoS in Sensornetzen

DS Domänen-Manager – Architektur

Differentiated-Services-Domänen-Manager

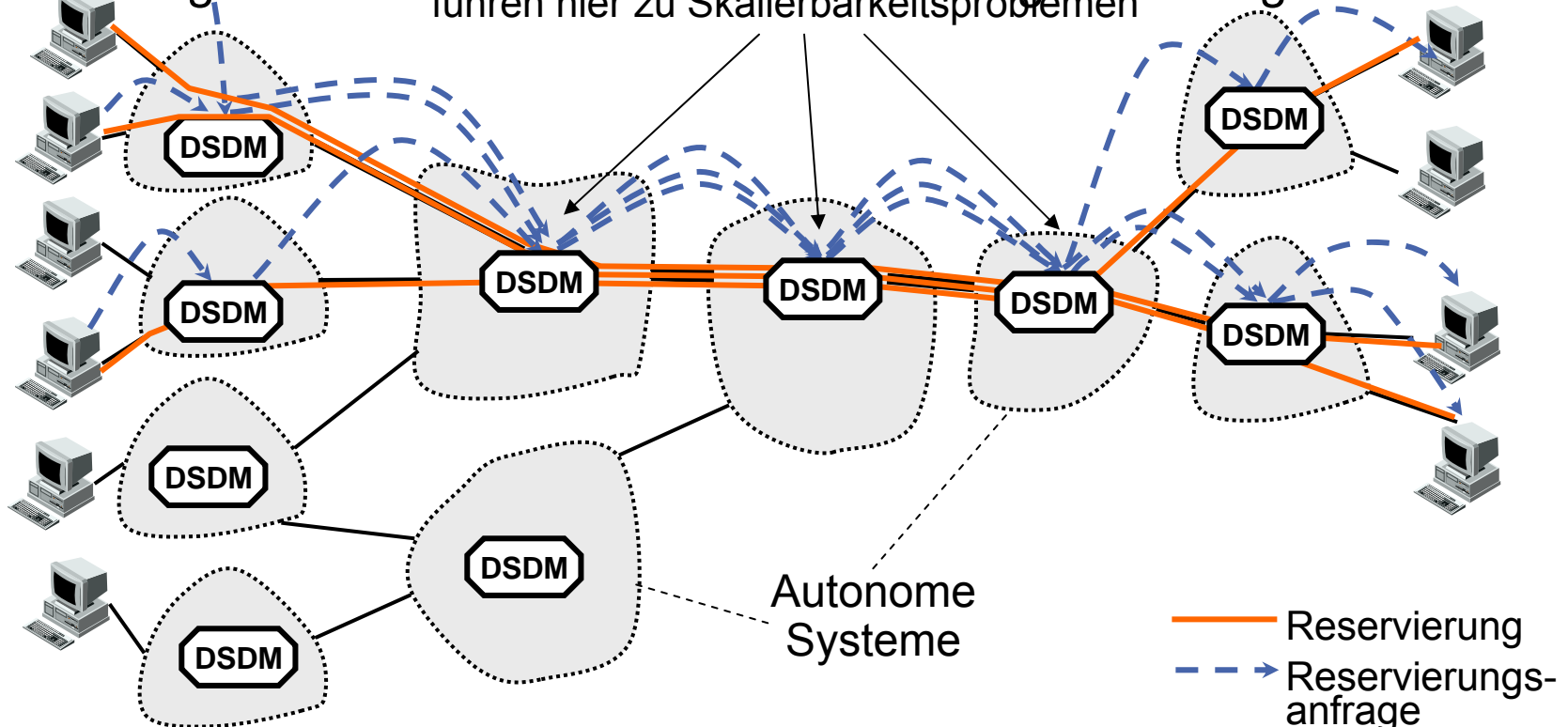


Skalierbarkeitsprobleme in der Kontrollebene

- Reservierung für jeden einzelnen Datenstrom (z.B. Bandbreite)

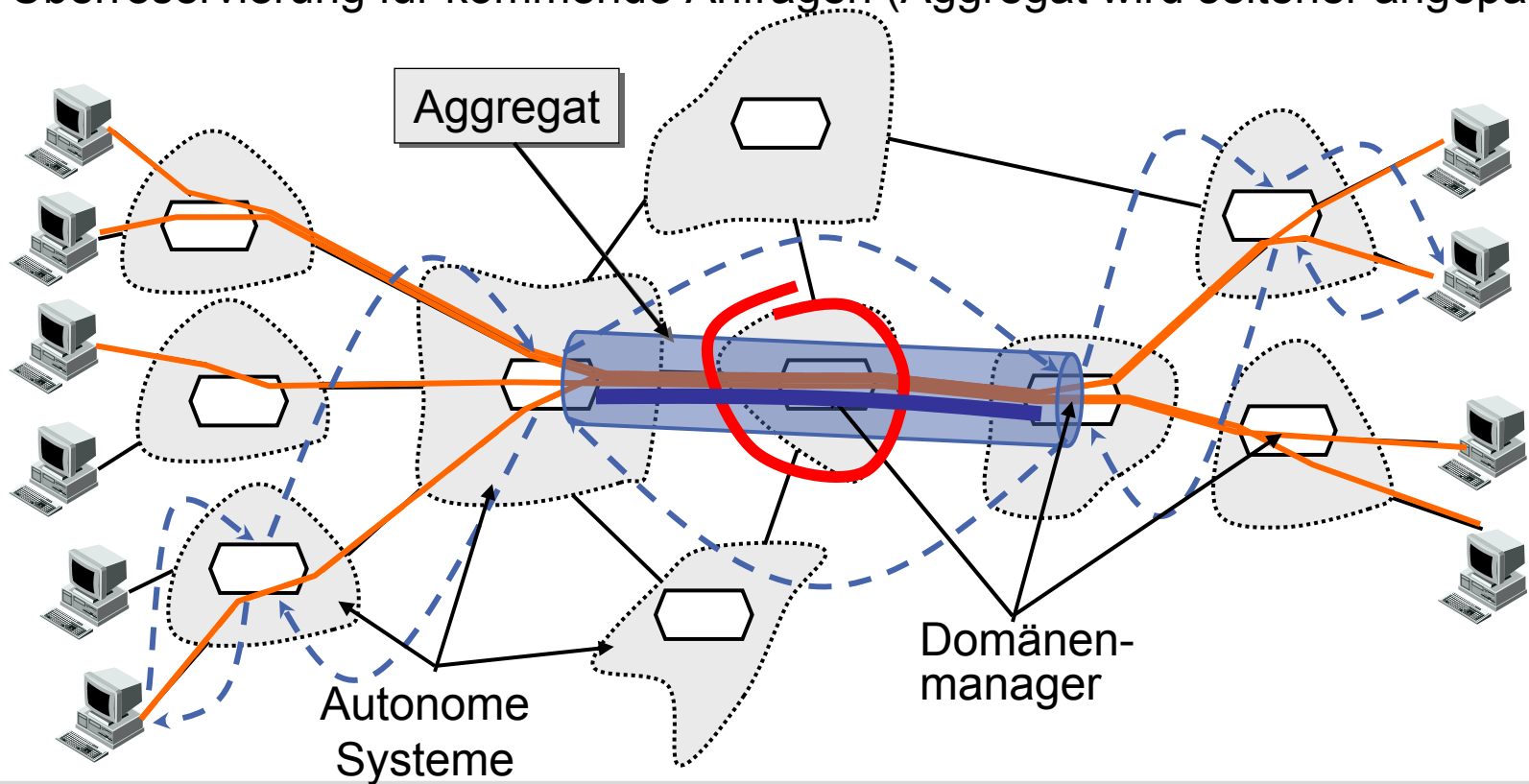
→ Skalierbarkeitsprobleme in Kernnetzbereichen

- bezüglich Anzahl der zu verwaltenden Reservierungszustände
 - bezüglich Anzahl der zu verarbeitenden Signalisierungsnachrichten
- zusammentreffende Anfragen/Zustände
führen hier zu Skalierbarkeitsproblemen



Das DARIS-Konzept – Dynamische Aggregation

- **Dynamic Aggregation of Reservations for Internet Services**
 - am Institut entwickeltes Konzept, um Skalierbarkeit zu erreichen
 - DSDM fasst mehrere Reservierungen entlang gleicher Teilpfade zusammen und erzeugt Aggregat (Reservierungsbündel)
 - Zustandsreduktion im mittleren AS
 - Überreservierung für kommende Anfragen (Aggregat wird seltener angepasst)



4.6 Einsatz von Dienstgütemechanismen


- Einige Betreiber (u.a. Sprint) stehen auf dem Standpunkt, dass Sie **keine Dienstgüteunterstützung** benötigen:
 - Sorgfältig dimensioniertes Netz garantiert niedrige Auslastung („Overprovisioning“) und max. Verzögerung
 - Netzdurchleitung ohne Paketverluste und mit Delay-Garantie
- Heutige Produkte verfügen standardmäßig über DiffServ-Mechanismen, größere Betreiber nutzen diese auch
- Einsatz v. DiffServ-Mechanismen am Rand in Zugangsnetzen spart ggf. Kosten

VoIP als treibende Kraft?

- Telecom Italia führt öffentlichen Telefonverkehr seit Ende 2002 über ihr IP-Netzwerk
- Einsatz von DiffServ-Mechanismen, um Datenverkehr von echtzeit-kritischem Sprachverkehr zu trennen
- EF für Sprachverkehr
- Kosten-Einsparungen durch integrierte Lösung auf Basis derselben Plattform
- Details:

http://newsroom.cisco.com/dlls/prod_100902.html

DiffServ Service Classes (1)

- Vorschlag zwölf verschiedene DiffServ-basierte Dienstklassen bereitzustellen (inklusive Konfigurationshinweisen):
 [RFC4594]
- Zwei Dienstklassen zur **Netzsteuerung**
 - Network Control: Routing und Netzwerksteuerung
→ CS 6
 - OAM (Operations, Administration and Management):
Netzwerkkonfiguration und -Management →
CS 2 (SR-BS)
- Zehn **Nutzerdienstklassen** (s. nächste Folie)

DiffServ Service Classes (2)

■ Nutzerdienstklassen:

- **Telefon** → EF m. Single Rate, Burst Size Token Bucket Control (SR-BS)
- **Signalisierung** → CS 5 (SR-BS)
- **Multimediakonferenz** → AF41,42,43 mit Two-Rate Three Color Marker
- **Echtzeit-Interaktiv** (interaktiv, variable Rate, niedriger Jitter, niedriger Verlust, sehr niedrige Verzögerung) → CS 4 (SR-BS)
- **Multimedia Streaming** (Variable Rate) → AF31,32,33 mit Two-Rate TCM
- **Broadcast Video** → CS 3 (SR-BS)
- **Niedrige Verzögerung** → AF21,22,23 mit Single-Rate TCM
- **Hoher Durchsatz** → AF11,12,13 mit Two-Rate TCM
- **Standard** (Best-Effort) → Default-PHB
- **Niedrige Priorität** → CS 1 [RFC 3662]

4.8 Übungen (1)

- 4.0 Welches sind die grundlegenden Mechanismen, die man zur Bereitstellung einer Dienstgüteunterstützung benötigt?
- 4.1 Welche Funktion hat ein Scheduler?
- 4.2 Was impliziert das Conservation Law?
- 4.3 Erläutern Sie den Begriff Schedulable Region.
- 4.4 Welche Vorteile hat WFQ gegenüber gewichtetem Round-Robin?
- 4.5 Berechnen Sie die Max-min Fair Share der Datenströme A, B, C, D und E mit Anforderungen der Größe 2, 3, 4, 4, 5 bei einer Kapazität von 15.
- 4.6 An einem Scheduler mit Fair Queueing kommen Dateneinheiten der Länge 100 bzw. 200 Bit der Datenströme A und B an. Die Datenrate der abgehenden Leitung betrage 100 Bit/s. Zu welchem Zeitpunkt sind die beiden Dateneinheiten bedient? Welche Rundenzahl ist erreicht nach Ende der Bedienung der Dateneinheiten?

4.7 Übungen (2)

- 4.7 Was versteht man unter elastischen Anwendungen?
- 4.8 Was ist bei der Dienstgüteaushandlung für Multicast zu beachten?
- 4.10 Erläutern Sie die grundlegenden Konzepte von Differentiated Services.
- 4.11 Welche prinzipiellen Unterschiede existieren zwischen DiffServ und IntServ?
- 4.12 Was sind die Unterschiede zwischen einem PHB und einem PDB?
- 4.13 Wie entsteht das NRS-Problem?

4.8 Literatur (1)

■ Bücher zum Thema QoS

- G. Armitage: „Quality of Service in IP Networks“, Macmillan Technical Publishing, 2000
- K. Kilkki: „Differentiated Services for the Internet“, Macmillan Technical Publishing, 1999
- B. Peterson, B. Davie: Computer Networks, Morgan Kaufmann, 2003, 3rd Ed., Kap. 6.1 (Ressourcen), 6.2 (Scheduling/Queueing), 6.5 (QoS)
- J. Kurose, K. Ross: Computer Networking, Pearson/Addison Wesley, 3rd Edition, 2005, Kap. 7.6–7.9 (QoS, Scheduling, DiffServ/IntServ, RSVP)
- S. Keshav: „An Engineering Approach to Computer Networking“, Addison Wesley, Kap. 9 zu Scheduling
- J. Liu: „Real-time systems“, Prentice Hall, 2000, zu Scheduling allgemein

Literatur (2)

- Z. Wang: „QoS – Architectures and Mechanisms for Quality of Service“, Morgan Kaufmann, 2001

■ Papers

- A. Demers, S. Keshav, S. Shenker: „Analysis and simulation of a fair queueing algorithm“, Proceedings of the ACM SIGCOMM 1989,
<http://portal.acm.org/citation.cfm?id=75248&coll=Portal&dl=ACM&CFID=45518805&CFTOKEN=19583060> (zugreifbar innerhalb der Universität)

Literatur (3)

- [FJa93] Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, V.1, N.4, August 1993, pp. 397-413. Also available from <http://ftp.ee.lbl.gov/floyd/red.html>
- [JGKT03] Ping Ji, Zihui Ge, Jim Kurose, Don Towsley, A Comparison of Hard-state and Soft-state Signaling Protocols, Proceedings of ACM SIGCOMM 2003, <http://www.acm.org/sigs/sigcomm/sigcomm2003/papers/p251-ji.pdf>
- [GIST09] H. Schulzrinne, R. Hancock, GIST: General Internet Signalling Transport, Juni 2009, IETF Internet-Draft, <http://tools.ietf.org/html/draft-ietf-nsis-ntlp-20>
- [LAJS03] Long Le, Jay Aikat, Kevin Jeffay, F. Donelson Smith, The Effects of Active Queue Management on Web Performance, Proceedings of ACM SIGCOMM 2003, <http://www.acm.org/sigs/sigcomm/sigcomm2003/papers/p265-le.pdf>
- [McJV96] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast", SIGCOMM '96, August 1996, pp. 117-130.

Literatur (4)

- [QoS-NSLP10] J. Manner, G. Karagiannis, A. McDonald. NSLP for Quality-of-Service Signaling, IETF Internet-Draft, Januar 2010
<http://tools.ietf.org/html/draft-ietf-nsis-qos-nslp-18>
- [QSPEC10] G. Ash, A. Bader, C. Kappler, D. Oran. QoS NSLP QSPEC Template, IETF Internet-Draft, Januar 2010
<http://www.ietf.org/html/draft-ietf-nsis-qspec-24>
- [RFC 791] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349. URL: <http://www.ietf.org/rfc/rfc791.txt>
- [RFC 2205] R. Braden, L. Zhang, S. Berson, S. Herzog und S. Jamin. Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936, 4495. URL: <http://www.ietf.org/rfc/rfc2205.txt>
- [RFC 2208] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib und L. Zhang. Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment. RFC 2208 (Informational), September 1997. URL: <http://www.ietf.org/rfc/rfc2208.txt>

Literatur (5)

- [RFC 2210] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC 2210 (Proposed Standard), September 1997. URL: <http://www.ietf.org/rfc/rfc2210.txt>
- [RFC 2215] S. Shenker, J. Wroclawski. General Characterization Parameters for Integrated Service Network Elements. RFC 2215 (Proposed Standard), September 1997. URL: <http://www.ietf.org/rfc/rfc2215.txt>
- [RFC 2212] S. Shenker, C. Partridge und R. Guerin. Specification of Guaranteed Quality of Service. RFC 2212 (Proposed Standard), September 1997. URL: <http://www.ietf.org/rfc/rfc2212.txt>
- [RFC 2474] K. Nichols, S. Blake, F. Baker und D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (Proposed Standard), Dezember 1998. Updated by RFCs 3168, 3260. URL: <http://www.ietf.org/rfc/rfc2474.txt>
- [RFC 2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang und W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), Dezember 1998. Updated by RFC 3260. URL: <http://www.ietf.org/rfc/rfc2475.txt>
- [RFC 2598] V. Jacobson, K. Nichols und K. Poduri. An Expedited Forwarding PHB. RFC 2598 (Proposed Standard), Juni 1999. Obsoleted by RFC 3246. URL: <http://www.ietf.org/rfc/rfc2598.txt>

Literatur (6)

- [RFC 2638] K. Nichols, V. Jacobson und L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. RFC 2638 (Informational), Juli 1999. URL: <http://www.ietf.org/rfc/rfc2638.txt>
- [RFC 3086] K. Nichols und B. Carpenter. Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification. RFC 3086 (Informational), April 2001. URL: <http://www.ietf.org/rfc/rfc3086.txt>
- [RFC 3246] B. Davie, A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu und D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). RFC 3246 (Proposed Standard), Marz 2002. URL: <http://www.ietf.org/rfc/rfc3246.txt>
- [RFC 3247] A. Charny, J. Bennet, K. Benson, J. Boudec, A. Chiu, W. Courtney, S. Davari, V. Firoiu, C. Kalmanek und K. Ramakrishnan. Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior). RFC 3247 (Informational), März 2002. URL: <http://www.ietf.org/rfc/rfc3247.txt>
- [RFC 3662] R. Bless, K. Nichols und K. Wehrle. A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services. RFC 3662 (Informational), Dezember 2003. URL: <http://www.ietf.org/rfc/rfc3662.txt>

Literatur (7)

- [RFC 3726] M. Brunner. Requirements for Signaling Protocols. RFC 3726 (Informational), April 2004. URL: <http://www.ietf.org/rfc/rfc3726.txt>
- [RFC 3754] R. Bless und K. Wehrle. IP Multicast in Differentiated Services (DS) Networks. RFC 3754 (Informational), April 2004. URL: <http://www.ietf.org/rfc/rfc3754.txt>
- [RFC 4594] J. Babiarez, K. Chan und F. Baker. Configuration Guidelines for DiffServ Service Classes. RFC 4594 (Informational), August 2006. URL: <http://www.ietf.org/rfc/rfc4594.txt>
- [ZhLW03] Z. Heying, L. Baohong, D. Wenhua, Design of a Robust Active Queue Management Algorithm Based on Feedback Compensation, Proceedings of ACM SIGCOMM 2003, <http://www.acm.org/sigs/sigcomm/sigcomm2003/papers/p0802-zhang.pdf>