# Dienstgüte-Unterstützung für zukünftige Netze

Tobias B. Hlavka
Betreuer: Dr. Heiko Niedermayer
Seminar Future Internet SS2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
E-Mail: tobias.hlavka@in.tum.de

## KURZFASSUNG

Quality of Service (QoS) ist als Merkmal für das Internet und für das zukünftige Internet von großer Bedeutung. Quality of Service umfasst die Merkmale Bandbreite, Latenz, Jitter und Paketverlust. Es ist momentan in Form des ToS-Headers in IPv4 vorgesehen. Die Arbeit zeigt, wie Differentiated Services nach nach Umdefinierung dieses Headers funktioniert. So ermöglicht es eine genauere Aufteilung in Verkehrsklassen, die in Behaviour Aggregates zusammengefasst werden und einem fest definierten Per-Hop-Behaviour unterliegen. Neben QoS kann für eine gute Netzperformance auch eine Überdimensionierung (Overprovisioning) stattfinden, was zunächst als einfacher Umsetzbar erscheint. In jedem Fall muss entschieden werden, wann eine Denial-of-Service-Situation vorliegt und wie und ob QoS-Maßnahmen in den Entwurf für ein Future Internet aufgenommen werden müssen.

## Schlüsselworte

Future Internet, Quality of Service (QoS), Dienstgüte, DiffServ, Differentiated Services, Overprovisioning

## 1. EINLEITUNG

Keshav erkannte bereits 1997:

> „The Holy Grail of computer networking is to design a network that has the flexibility and low cost of the Internet, yet offers the end-to-end quality-of-service guarantees of the telephone network.“ [1] [9]

Diese Feststellung zeigt, dass der Gedanke eines internetartigen Netzwerks mit Unterstützung für Quality-of-Service (QoS) keineswegs neu ist. Aus wirtschaftlichen Gründen fragt sich die Branche seit Jahren, ob es möglich ist, ein höher priorisiertes „Netz im Netz“ zu schaffen - natürlich gegen Entgelt. Oder ist dafür nicht doch eine ganz neue Netzstruktur abseits des in die Jahre gekommenen WWW notwendig? Immer neue Anwendungen für das Internet lassen diese Idee plausibel erscheinen. Es entstehen Consumer-Dienste, die nach hoher Bandbreite verlangen (z.B. YouTube), aber auch kommerzielle Anwendungen wie beispielsweise Fern-Operationen durch Spezialärzte und über das Internet ferngesteuerte Roboter, welche nach äußerst ausfallsicheren Leitungen mit geringer Latenz verlangen. Andererseits ist es fraglich, ob eine solche Dienstgüteunterstützung - und damit die Aufteilung der Daten in verschiedene Klassen - überhaupt wünschenswert ist. Es darf bezweifelt werden, dass sich das WWW zu dem entwickelt hätte, was es heute ist, wenn eine solche Klassifizierung von Anfang an nach monetären Aspekten erfolgt wäre. Dies soll neben den Fragen, ob Dienstgüteunterstützung für ein „Future Internet“ überhaupt benötigt wird und wie diese QoS-Maßnahmen technisch realisiert werden könnten, im Folgenden geklärt werden.

Der zweite Abschnitt soll zeigen, um was es sich bei QoS überhaupt handelt. Dazu wird zunächst eine Definition der QoS-Parameter gegeben. Es wird erklärt, wie QoS heute in den IP-Headern verankert ist.

Der dritte Abschnitt beschäftigt sich mit der konkreten technischen Maßnahme DiffServ. Es wird eine Übersicht über die Grundsätze von DiffServ, dessen Architektur und den DiffServ Codepoint gegeben. Danach werden die verschiedenen Per-Hop-Behaviours erklärt und gegenübergestellt. Schließlich wird aufgezeigt, in welchen Gebieten DiffServ als Methode geeignet ist, QoS zu implementieren, es wird die Alternative IntServ betrachtet und eine Bewertung beider Techniken vorgenommen.

Im vierten Abschnitt wird betrachtet, inwiefern Overprovisioning QoS überflüssig machen könnte.

Welche Art von Attacken auf ein QoS-unterstüzendes Netzwerk abzielen könnten, wird in Sektion fünf aufgeklärt.

In Abschnitt sechs wird geklärt, inwiefern QoS im Future Internet von Relavanz ist.

## 2. QUALITY OF SERVICE ALS ANFORDERUNG FÜR ZUKÜNFTIGE NETZE

QoS ist eine der wesentlichen Gesichtspunkte bei der Gestaltung zukünftiger Netze. Dieser Abschnitt definiert zunächst QoS und zeigt die Merkmale auf, auf die QoS-Strategien angewendet werden können. Danach wird die momentane Situation von Dienstgüteunterstützung für das jetzige Internet aufgezeigt.

### 2.1 Was bedeutet QoS

Generell bedeutet QoS, ein bestimmtes Service Level Agreement (SLA) zu erfüllen. Dieses SLA bezieht sich auf die Qualität der Datenverbindung und kann zwischen zwei Providern oder zwischen Provider und Endkunde abgeschlossen

[1] etwa: „Der heilige Gral der Computernetze ist es, ein Netzwerk zu entwerfen, das so flexibel und preiswert wie das Internet ist, jedoch zugleich die vom Telefonnetz bekannten Dienstgüte-Garantien für Punkt-zu-Punkt-Verbindungen bietet.“ (Übers. d. Autors)


doi: 10.2313/NET-2011-07-2_03

werden. Im Allgemeinen werden dort folgende Parameter mithilfe von Kennzahlen für die jeweilige Qualitätsstufe fixiert:

1. Bandbreite: Beschreibt eine Mindestdatenrate, die für eine bestimmte Klasse von Datenverkehr garantiert werden muss
2. Latenz: Höchste Verzögerung, mit der ein Datenpaket einer bestimmten Klasse von Datenverkehr ausgeliefert wird
3. Jitter: Größte Schwankungsbreite der Latenz einer bestimmten Klasse von Datenverkehr
4. Paketverlust: Das Verhältnis von verlorenen Paketen zu gesendeten Pakete einer bestimmten Klasse von Datenverkehr

Diese Punkte lassen sich auf eine anschauliche Analogie zum Straßenverkehr übertragen, womit das Beispiel von Bricklin [3] noch erweitert werden soll: Im Falle von hohem Verkehrsaufkommen ist es zuerst einmal notwendig, Staus zu vermeiden und einen befahrbaren Weg für hoch priorisierten Verkehr, wie Feuerwehr, Notarzt und Polizei freizuhalten. Dies passiert entweder mit speziell reservierten Spuren für solchen Notfallverkehr, oder eben einer ausreichenden Anzahl an Fahrspuren (1). Verspätungen (2) sollen verhindert werden. Das wird in der realen Welt durch geeignete Lichtsignale, insbesondere Blaulicht, erreicht. Ebenso will man größere Unregelmäßigkeiten im Straßenverkehr vermeiden, man sollte für die gleiche Strecke zu unterschiedlichen Zeitpunkten gleich lang unterwegs sein (3). Die Unfallrate (4) muss natürlich niedrig gehalten werden, sowohl für allgemeinen Verkehr, aber auch für den priorisierten Blaulichtverkehr. Dies wird dadurch erreicht, dass alle anderen Verkehrsteilnehmer darauf konditioniert wurden, Blaulichtverkehr zu beachten. Diese Beachtung kann beispielsweise in Form von Bremsen und an den Rand der Fahrbahn fahren erfolgen, oder darin bestehen eine Abfahrt zu nehmen und die Straße zu verlassen, unter Umständen mit der Prämisse, das Ziel nicht oder nur verspätet zu erreichen. Es deutet sich hier schon an, dass alle diese Dienstgüteunterstützungsmaßnahmen überhaupt nur dann notwendig sind, wenn die vorhandene Straßen- bzw. Leitungskapzität nicht ausreicht. Eine solche QoS-Sicherung ist bei bestimmten Netzwerkanwendungen unabdingbar. Hier sind besonders zeitkritische Anwendungen zu nennen, die sich von Spezialgebieten wie Business-Prozessen bis zu Consumer-Themen wie Voice-over-IP erstrecken. Hohe Latenzen würden diese Verfahren ad absurdum führen. Andererseits muss auch beachtet werden, dass für einen großen Teil von Netzwerkanwendungen eine Einführung von QoS nicht notwendig ist. Generell gesagt ist das für alle diejenigen Anwendungen der Fall, deren Funktion nicht zeitlich vom Eintreffen bestimmter Pakete zu genau definierten Zeitpunkten im Kommunikationsprozess abhängig ist. Das bedeutet konkreter, dass Video-on-Demand-Portale wie YouTube, oder auch asynchrone Kommunikationsmittel wie E-Mail nicht unmittelbar von QoS profitieren, da sowohl Zeit- als auch Datenpuffer vorhanden sind.

## 2.2 Status Quo - QoS heute

Ansätze, um QoS in den Netzwerkverkehr zu integrieren, gab es bereits mit IPv4. IPv4 sieht in dieser Hinsicht das so genannten Type-of-Service (ToS)-Byte im Header vor. Dieses Byte ist wie in Tabelle 1 dargestellt aufgebaut.

**Tabelle 1: Aufbau des IPv4 ToS-Byte [4]**

| Bit Nr. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Bedeutung | Priorität | | | Type of Service | | | | Null |

Somit ist es bereits mit IPv4 möglich, verschiedene Prioritäten wie „Immediate“ oder „Routine“ in Bit 0 bis 2 festzulegen, die im Wesentlichen an den Netzgrenzen berücksichtigt werden. Je kleiner die Priorität, desto eher darf ein Paket bei hoher Auslastung bzw. Überlastung verworfen werden um Freiraum für höher priorisierte Pakete zu schaffen. Eine weitere Klassifizierung erfolgt anhand der ToS-Bits 3 bis 6. Hier können grobe Einteilungen wie zum Beispiel „minimize monetary cost“ oder „maximize throughput“ festgelegt werden. Es muss hier also eine Abwägung stattfinden zwischen Durchsatz, Verzögerung, Zuverlässigkeit und Kosten [2]. Das letzte Bit blieb unbenutzt und muss, um ToS-konform zu sein, null sein. Für detaillierte Belegungsmöglichkeiten und resultierende Bedeutungen siehe [4]. An dieser Stelle reicht es aus, zu verstehen, dass bereits ein Klassifizierungsbyte im Header vorgesehen ist und dass dieses auch bereits mit konkreten Bedeutungen belegt ist. Das gilt es zu beachten, wenn versucht wird, für IPv4 eine neue Struktur für QoS zu schaffen.

Warum hat nun dieses ToS-Byte nicht ausgereicht um eine skalierbare QoS-Infrastruktur zu schaffen? Wieso wird ein neuer Ansatz benötigt? Dafür gibt es nach [4] mehrere Gründe:

Die Prioritäten- und Type-of-Service-Bits sind nicht flexibel genug. Sie erlauben lediglich eine Angabe der Priorität und des ToS relativ zu anderen Klassen. Absolute Angaben sind nicht möglich. Dazu kommt, dass sich keine Festlegung treffen lässt, wie Traffic in der gleichen Klasse, jedoch mit unterschiedlichem Inhalt, im Falle einer hohen Last behandelt wird. Werden beispielsweise HTTP und SSH in die gleiche Klasse eingeordnet, welcher Service wird dann zuerst fallengelassen? Es sollte möglich sein, hier eine Unterscheidung treffen zu können, damit beispielsweise wichtiger Geschäftsverkehr (HTTP) höherwertiger behandelt wird als die SSH-Session eines internen Benutzers.

Darüber hinaus gibt es zu wenige Prioritätenklassen. Da nur drei Bit dafür reserviert sind, ergeben sich $2^3 = 8$ Prioritätenklassen, wovon zwei aber schon für interne Router-Nachrichten reserviert sind, die zugleich als höchstpriorisiert behandelt werden, damit auch im High-Traffic-Fall auf jeden Fall die Netzwerkknoten untereinander kommunizieren können.

Schließlich halten sich die Hersteller in ihren Implementierungen nicht an die Bit-Definitionen im ToS-Feld, sodass hier sehr uneinheitliche Implementierungen entstanden sind. Darüber hinaus wurden die Bits in RFC 1349 [2] neu definiert, was zu zusätzlichen Überschneidungen geführt hat. Folglich ist ein Clean-Slate-Ansatz nötig geworden.

Es besteht zusätzlich das Problem, dass mit den gegebenen Mitteln QoS nur auf flow-Basis definiert werden kann. Ein flow ist ein einzelner Datenstrom. Er besteht aus dem 5-Tupel (Quell-IP-Adresse; Quell-Portnummer; Ziel-IP-


doi: 10.2313/NET-2011-07-2_03

Adresse; Ziel-Portnummer; Protokoll [UDP / TCP]) [4]. Diese sehr feingranulare Einteilung ermöglicht natürlich sehr exakte QoS-Vorgaben auf Paketbasis. Dies scheint durchaus implementierbar in kleineren Netzverbänden und an Routern mit mäßigem Traffic, denn zur Berücksichtigung der QoS-Richtlinen müssen beim Eintreffen des Pakets verschiedene Tests und Berechnungen durchgeführt werden. Dies wiederum kostet Zeit, was die Latenz bei höherem Datenaufkommen empfindlich steigern kann. Dies würde genau dem Gedanken des Verfügbarmachens von QoS widersprechen! Man mag sagen, an den Blättern des Netzes ließe sich auch das noch durch gesteigerte Rechenkraft ausgleichen. Jedoch spätestens bei Betrachtung der zentralen Backbones des Internets mit einer Anzahl von flows in vielfach höherer Größenordnung ist diese Methode nicht mehr praktikabel. Um also eine angemessene Skalierbarkeit der QoS-Maßnahmen herzustellen, müssen die flows in irgendeiner Form zusammengefasst und somit gleich zu behandelnder Traffic in gröber eingeteilten Klassen aggregiert werden. So kann eine deutliche Senkung des Overheads in den Netzknoten erreicht werden und der ganze Prozess des QoS kostet nicht übermäßig unmittelbare Rechenleistung.

# 3. FALLBEISPIEL DIFFERENTIATED SERVICES

In diesem Abschnitt wird zunächst DiffServ als Technik zur Umsetzung von QoS-Maßnahmen definiert und erklärt. Danach wird geklärt, wo es sich einsetzen lässt und wie dies zu Future Internet Ansätzen passt. Schließlich erfolgt noch eine kurze Darstellung von Alternativen zu DiffServ und eine vergleichende Bewertung.

## 3.1 Definition DiffServ / DSCP

Differentiated Services (DiffServ) wurde von der IETF[2] entworfen, wobei das Hauptaugenmerk auf Skalierbarkeit gelegt wurde. Man hat also den konsequenten Schritt gemacht, statt auf feingranularer flow-Basis, die QoS-Maßnahmen auf einer Aggregat-Basis anzuwenden. Es werden daher gleichartige Pakete zu gemeinsamen Klassen von Netzwerkverkehr zusammengefasst. So kann das Netz an sich in den Knoten recht simpel und kostengünstig gehalten werden, während die komplexeren Berechnungen und Klassifikationen an den Netzgrenzen und -blättern vorgenommen werden. Der strukturelle Aufbau des Differentiated Services Codepoints (DSCP), welcher Bit 0 bis 5 im IPv4-ToS-Byte umfasst, ist in Tabelle 2 dargestellt.

**Tabelle 2: Das DiffServ Codepoint Field (nach [4])**

| Bit Nr. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| DS-Feld | Class Selector CP | | | | | | ungenutzt | |
| | Differentiated Services Codepoint | | | | | | | |

Das ursprüngliche Type-of-Service-Byte wurde vollständig umdefiniert. Die ToS-Prioritäten-Bits wurden mit den weiteren Bits zusammengelegt. So festgelegt in [10], gibt es nun $2^6$ Möglichkeiten, Pakete zu klassifizieren. Dies stellt eine großen Sprung zum alten ToS-Byte dar. Wichtig bei der Betrachtung dieser Neudefinition ist, dass die für DSCP definierten Bit-Sequenzen in den ersten drei Bits deckungs- und bedeutungsgleich mit der alten Definition sind. Das heißt, dass diese Technik soweit abwärtskompatibel ist, dass diejenigen Router, welche ToS-Markierungen unterstützen, weitergenutzt werden können und mit den neuen DSCP-fähigen Routern zusammenarbeiten. Das in der Tabelle gezeigte Sextett bestimmt das sogenannte Per-Hop-Behaviour, welchem die Pakete an den Netzknoten unterliegen.
Analog zur IPv4-Umdefinierung ist QoS mit Hilfe von DiffServ auch in IPv6 möglich. Dort wird die Markierung im Traffic-Class-Byte vorgenommen [10]. Die generelle Funktionsweise von DiffServ ist unabhängig davon, ob IPv4 oder IPv6 eingesetzt wird.

[2]Internet Engineering Taskforce

## 3.2 Aufbau der Netzstruktur mit DiffServ

Wie in Abbildung 1 dargestellt, besteht eine DiffServ-Domäne für gewöhnlich aus einer einzelnen Administrationseinheit. Innerhalb derer bestehen fest definierte Service Level Agreements, deren Einhaltung als sicher gilt. Besonders zu beachten sind hier die Eingangs- und Ausgangsknoten.

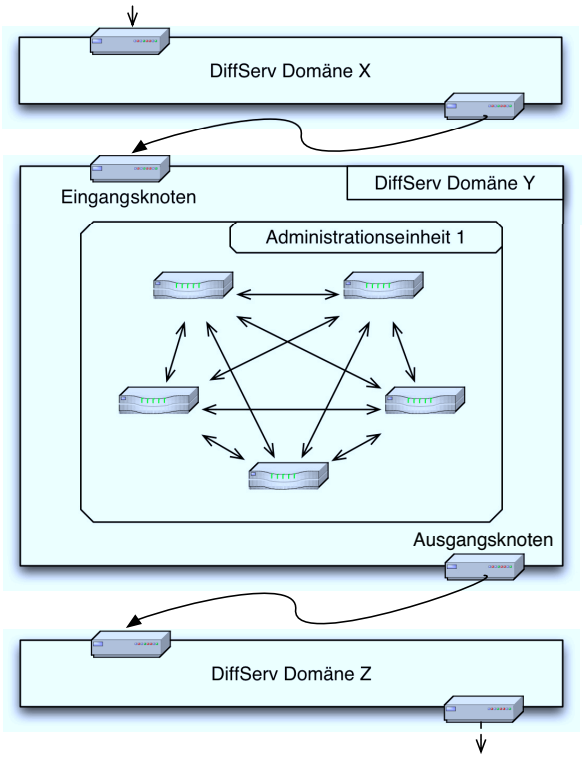


**Abbildung 1: Die Architektur von DiffServ**

### *3.2.1 Knoten innerhalb einer DS-Domäne*

Die Knoten innerhalb einer Domäne können haben hauptsächlich zwei Aufgaben. Die erste Aufgabe ist das Klassifizieren und Auslesen der Codepoints. Dies beschränkt sich innerhalb einer Administrationseinheit auf das Auslesen des DSCP, denn hier herrschen übereinstimmende SLAs zwischen den Routern. Die zweite Aufgabe ist das Weiterleiten der Pakete innerhalb der DS-Domäne anhand der zuvor ausgelesenen Kriterien passend zu den vereinbarten Per-Hop-Behaviours (siehe Abschnitt 3.3).


doi: 10.2313/NET-2011-07-2_03

### 3.2.2 *Ein- und Ausgansknoten außerhalb einer DS-Domäne*

Diese Knoten werden auch als Grenzknoten (Boundary Nodes) bezeichnet. Sie verbinden entweder zwei DiffServ-Domänen miteinander oder aber eine DiffServ-Domäne mit einem anderen Netz, das kein DiffServ unterstützt. Diese Knoten haben neben der Klassifizierung und Weiterleitung noch die Aufgaben der Markierung und Überwachung. Ein Markieren muss immer dann stattfinden, wenn Pakete aus nicht-DiffServ-Netzen eintreffen. Dabei wird je nach geltenden SLAs eine DiffServ Codepoint zugeordnet, der dann das Verhalten definiert, welches innerhalb der Domäne auf das Paket oder das Behaviour Aggregate angewandt wird. Die Überwachung spiegelt sich in der Form wider, dass über Traffic-Shaping-Maßnahmen sichergestellt werden muss, dass die definierten QoS-Anforderungen erfüllt werden können. Abbildung 2 zeigt den typischen Aufbau dieses DiffServ Traffic Conditioner Block (TCB) [4], in dem die genannten Maßnahmen umgesetzt werden.

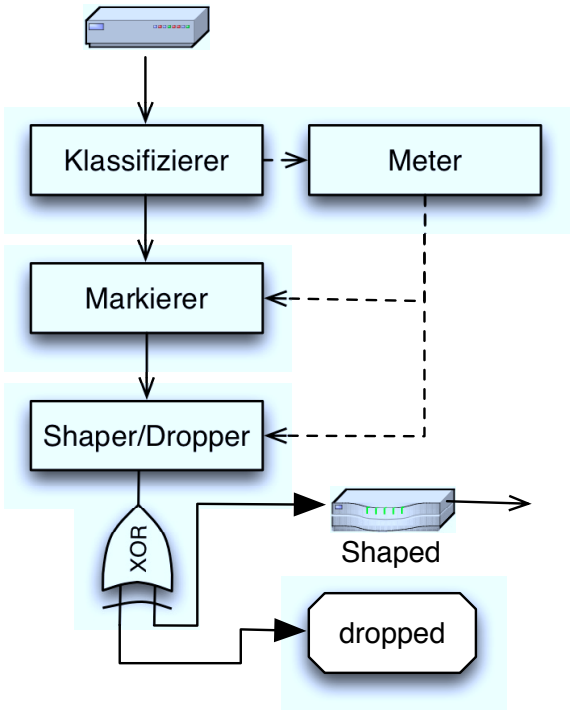


**Abbildung 2: Der DiffServ Traffic Conditioner Block**

Der Klassifizierer liest hier die DSCPs aus und stellt gegebenenfalls Behaviour Aggregates zusammen. Falls noch kein DSCP festgelegt ist, wird dies durch den Markierer nach der vom Klassifizierer festgestellten Klasse angelegt. Das Meter-Modul fungiert hier als eine Kontrollinstanz, die überprüft, ob der markierte eintreffende Datenstrom auch zu der Klasse gehört, die er im DSCP vorgibt zu sein. Passen Markierung und Pakettyp nicht zusammen, wird es im Shaper entweder einer anderen Klasse zugeordnet (das heißt der DSCP wird geändert) und dann in die DS-Domäne geroutet oder aber es wird verworfen und geht somit verloren. Dadurch wird sichergestellt, dass alle Pakete, die in die Domäne gelangen die dort geltenden SLAs erfüllen. Dies zeigt die enorme Wichtigkeit der Eingangs- und Ausgangsknoten.
Wie der Traffic innerhalb einer DS-Domäne behandelt wird, wird im folgenden Abschnitt dargelegt.

## 3.3 Per Hop Behaviours (PHBs)

Als Hop (Etappe) bezeichnet man den Weg zwischen zwei Netzknoten. Folglich versteht man unter Per-Hop-Behaviour (PHB) verschiedene Verhaltensweisen zur Behandlung und Weiterleitung einzelner Pakete oder Paket-Bündeln. Zu diesen Paketbündeln[3] werden solche Pakete zusammengefasst, welche die gleiche Bitsequenz im DSCP tragen. Im RFC 2474 wird PHB definiert als

> „[...] a description of the externally observable forwarding treatment applied at a differentiated services-compliant node to a behavior aggregate.“ [4][10]

Es muss also konkret entschieden werden, welcher CoS[5] ein Paket zugeordnet wird, insbesondere unter den Gesichtspunkten scheduling, queuing, policing und shaping - relativ zum zwischen den Endpunkten vereinbarten SLA. Die folgenden vier PHBs sind in verschiedenen RFCs beschrieben:

### 3.3.1 *Default PHB*

Dieses PHB wird definiert in RFC 2474 [10]. Es muss verpflichtend von allen DS-kompatiblen Knoten implementiert werden. Inhaltlich handelt es sich hierbei um die herkömmliche Best-Effort-Qualitätsstufe, bei der eben lediglich *versucht* wird, so viele Pakete wie gerade möglich so schnell wie gerade möglich weiterzuleiten. Gleichzeitig handelt es sich hier um eine Art Rückfallklasse, der alle Pakete zugeordnet werden, die keine anderweitige PHB-Markierung mit sich tragen. Dadurch wird eine gewisse Abwärtskompatibilität zu Netzteilnehmern erhalten, welche keine DiffServ-Mechanismen implementieren, wenngleich sie unter Umständen eine sehr schlechte Behandlung erhalten. Wenn man davon ausgeht, dass Default-PHB-Traffic zunächst gepuffert wird und dann nur versendet wird, wenn gerade keine höhere PHB-Klasse Bandbreite beansprucht, kann der Puffer schnell überlaufen und viele Pakete verloren gehen. Das DSCP-Bitmuster ist ’000000’, was wie eingangs erwähnt im Einklang zu ’000’ für den Prioritäten-Teil des ToS-Bytes steht. Es steht dem implementierenden Netzwerkknoten frei, die mit ’000000’ markierten Pakete mit einer höheren Priorität zu versehen, sofern das innerhalb einer Netzwerkdomäne notwendig ist. Dies könnte beispielsweise der Fall sein, wenn innerhalb der Domäne abweichende SLAs zwischen den Routern gelten.

### 3.3.2 *Class-Selector PHB*

Dieses PHB wird ebenfalls definiert in RFC 2474 [10]. Es existiert, weil eine - zumindest teilweise - Rückwärtskompatibilität zum vorhandenen ToS erhalten werden soll. Es

[3] Behaviour Aggregates, kurz: BA
[4] etwa: „[...] Eine Beschreibung der von außen beobachtbaren Weiterleitungsart, mit der ein Paketbündel an einem DiffServ-kompatiblen Knoten behandelt wird.“ (Übers. d. Autors)
[5] Class of Service, etwa: „Serviceklasse“ (Übers. d. Autors)


doi: 10.2313/NET-2011-07-2_03

ist erlaubt, dass mehrere Class-Selector Codepoints auf die gleiche Klasse zeigen; es muss aber auch klar definierte Unterscheidungen geben, ein Mapping auf eine einzige Klasse ist nicht zulässig. Das Bitmuster lautet hier 'xxx000' mit $x \in \{0, 1\}$. So ergibt sich eine isomorphe Abbildung der ToS-Prioritäten-Bits. Es werden auch dessen Regeln zur Behandlung der Pakete angewandt, das heißt, höhere 'xxx'-Werte bedeuten eine höhere Priorität beim Weiterleiten und Erhalten der so markierten Pakete. PHBs mit unterschiedlichen Class Selector Codepoints sollen unabhängig voneinander weitergeleitet werden. So ist es möglich, dass DiffServ-kompatible Router parallel zu Routern, die IP-Prioritäten berücksichtigen, existieren können.

### 3.3.3 Expedited Forwarding PHB

Dieses PHB wird definiert in RFC 2598 [8]. Expedited Forwarding[6] wird immer dann eingesetzt, wenn Datenverkehr mit geringem Paketverlust, geringer Latenz, niedrigem Jitter und mit einer garantierten Bandbreite erfolgen soll. Typische Anwendungsfälle für dieses PHB wären VoIP, Videoanwendung oder auch die Kommunikation zwischen ERP[7]-Systemen, die für die just-in-time-Abwicklung von Geschäftsprozessen auf eine Verbindung mit den genannten Eigenschaften angewiesen sind.
Die für EF vorgesehene Bitsequenz ist '101110', was der IP-Priorität '101' entspricht, jedoch mit einer weiteren Spezifizierung in den letzten drei Bits. Es gilt zu beachten, dass EF PHB sparsam eingesetzt werden muss. Schließlich ist es nicht zielführend, zu viel (oder gar allen) Traffic als EF zu betrachten, denn somit wären sämtliche Klassifizierungen egalisiert und im Falle einer hohen Netzlast käme es wiederum zu unerwünschtem Verhalten.

### 3.3.4 Assured Forwarding PHB

Dieses PHB wird definiert in RFC 2597 [7]. Mit Assured Forwarding[8] (AF) ist es möglich, verschiedene Klassen von Behaviour Aggregates zu schaffen. „The Internet Society" [7] schlägt hier eine „olympische Einteilung" in Gold-, Silber- und Bronzetraffic vor. Cisco Systems empfiehlt konkret eine Bandbreitenaufteilung von 50%/30%/20% [4]. Es gibt insgesamt vier AFx-Klassen ($x \in \{1, 2, 3, 4\}$), für die jeweils eine bestimmte Größe an Pufferspeicher und eine gewisse Bandbreite vorgehalten werden. Die konkrete Behandlung dieser Klassen hängt von den für die betroffenen Knoten ausgehandelten SLAs ab. Zusätzlich kann man jeder AFxy-Klasse noch drei Stufen $y \in \{1, 2, 3\}$ zuweisen, die innerhalb der einzelnen Klassen bestimmen, welche Pakete im Fall von hoher Auslastung bzw. Überlastung zuerst verworfen werden sollen. Dies macht es möglich, einzelne flows innerhalb des Behaviour Aggregates unterschiedlich zu behandeln, beispielsweise wenn ein einziger flow die gesamte für die AFx-Klase reservierte Bandbreite für sich beansprucht. So lassen sich alle AF-Klassen darstellen aus Sextetten der Form „pqrab0", wobei „pqr" die Binärdarstellung für das obige x und „ab" die Binärdarstellung für die Drop-Priorität y darstellt. Es ergeben sich damit die in Tabelle 3 gezeigten möglichen Kodierungen.

[6] etwa: „beschleunigte Weiterleitung" (Übers. d. Autors)
[7] Enterprise Ressource Planning
[8] etwa: „garantierte Weiterleitung" (Übers. d. Autors)

**Tabelle 3: DiffServ AF Codepoints Fields (nach [7], [5])**

| *AFxy* | x = 1 | x = 2 | x = 3 | x = 4 |
|---|---|---|---|---|
| y = 1 | 001010 | 010010 | 011010 | 100010 |
| y = 2 | 001100 | 010100 | 011100 | 100100 |
| y = 3 | 001110 | 010110 | 011110 | 100110 |

Die konkreten technischen Implementierungsöglichkeiten der einzelnen PHBs gehen über den Rahmen dieser Arbeit hinaus und können gesondert nachgelesen werden.

## 3.4 Eignungsgebiete Differentiated Services

Es zeigt sich, dass DiffServ geeignet ist um Datenströme in BAs zusammenzufassen und diese dann nach einem geregelten Schema zu strukturieren. Da eine Umdefinierung der PHBs und somit der Prioritäten gestattet ist, eignet sich diese Technik hauptsächlich um Traffic innerhalb einer geregelten DiffServ-Domäne zu steuern. Diese DiffServ-Domäne kann ihrerseits wieder aus verschiedenen Administrationseinheiten bestehen. Grundvoraussetzung ist hier, dass alle Router innerhalb dieser einen DiffServ-Domäne auch tatsächlich eine deckungsgleiche Menge an DiffServ-Funktionalitäten erfüllen. Viel entscheidender ist aber, dass das zwischen den Knoten bestehenden SLA innerhalb der Domäne klar definiert ist und von allen Knoten strikt eingehalten wird.
Somit empfiehlt sich der Einsatz von DiffServ insbesondere für überschaubare Netze, deren Teilnehmer wohlbekannt sind, denen vertraut werden kann und mit denen verbindliche SLAs ausgehandelt werden können.

## 3.5 Alternativen zu DiffServ

Der größte Gegenentwurf zu DiffServ ist IntServ (Integrated Services). Dieses System bearbeitet als wesentlichen Unterschied zu DiffServ keine zusammengefassten Klassen von Datenverkehr, sondern legt ein QoS für jede einzelne Verbindung fest [14]. Das Gegenstück zu den DiffServ BAs und PHBs stellt hier das sog. Resource Reservation Protocol (RSVP) dar. So kann jeder Router auf flow-Basis angefragte Leitungskapazitäten reservieren und muss diese auch für die Verbindungsdauer freihalten und garantieren. Diese Garantie wird so lange aufrecht erhalten, wie die Reservierung erneuert wird. Trifft nach einem definierten Timeout keine neue Reservierung ein, so werden die Ressourcen wieder freigegeben [15]. Zur Markierung des Traffics wird ebenfalls das ToS-Feld im IPv4-Header benutzt. Die zwei grundlegenden Dienstkategorien sind „Guaranteed QoS" (macht quantitative Zusagen) und „Controlled Load" (macht lediglich vage qualitative Zusagen, weitestgehend Best-Effort). Details sind in [14] spezifiziert.
Es hat sich aber als nicht global einführbar erwiesen, da ein inkrementelles Deployment nicht möglich ist. Um IntServ einzuhalten, muss explizit jeder Router diese Technik unterstützen, es gibt keine Fallback-Lösung. Darüber hinaus ist das Queue-Management auf flow-Basis im kleinen, ähnlich wie beim IPv4-ToS, noch handhabbar, in den großen Backbones mit extrem vielen flows ist das Queue-Management jedoch kaum noch möglich [15].


doi: 10.2313/NET-2011-07-2_03

## 3.6 Bewertung von DiffServ

Wie in Abschnitt 3.4 erwähnt, sind die SLAs ein essentieller Teil von Netzen, die DiffServ implementieren. Werden diese SLAs nicht eingehalten, kommt es zu einer großen Last an den Ein- und Ausgangsknoten einer Domäne; der Traffic muss dauerhaft geshaped werden. Eine große Gefahr stellen hier kompromittierte Knoten dar, wie in Abschnitt 5 dargelegt wird. Der wunde Punkt sind also insbesondere die Eingangs- und Ausgangsknoten einer DiffServ-Domäne und die Übergänge zwischen diesen. Hier gilt es entweder zwischen den Domänen SLAs auszuhandeln und festzulegen, oder aber eine Re-evaluierung des Traffics beim Passieren der Domänengrenzen vorzunehmen. Dies ist zeitintensiv und rechenaufwändig. Außerdem ist das end-to-end-Verhalten solcher Trafficströme, die mehrere Domänen passieren, nicht vorhersehbar. Zum einen kann es sein, dass manche Netzbereiche gar kein DiffServ unterstützen - dann ist ein QoS nicht mehr möglich. Zum anderen könnten aber verschiedene Netze den Traffic völlig unterschiedlich behandeln, obwohl sie beide DiffServ unterstützen. Denn für Onlinespieler ist natürlicherweise Spiel-Traffic wesentlich wichtiger als eine Aktienorder - und so könnte diese beim Passieren eines solchen Subnetzes durch die Router empfindlich verlangsamt werden oder im Extremfall bei hoher Netzlast verloren gehen, wenn Internetverkehr des einen Interessengebietes den Bereich des anderen durchläuft.

Weiterhin führen die diversen Aggregationsverhalten zu unvorhersehbaren Effekten bei Ende-zu-Ende-Verbindungen. Es scheint unrealistisch, dass hier Absprachen zwischen den Providern getroffen werden können, die zu einer Standardisierung führen. Diese Koordination müsste weltweit stattfinden, sodass schließlich jegliche Differenzierung zwischen den Providern verloren ginge. Eine Lösung dafür könnten Interconnection-Entgelte sein, die in dieser Form ja schon im Telefon und Mobilfunkbereich existieren. Es bleibt jedoch die Frage, wie viel Best-Effort-Kapazität man mit DiffServ vorhalten könnte, mit der sich am wenigsten Profit erwirtschaften lässt, oder ob die Provider dann beginnen, bei hoher Last keine Teilnehmer mehr zuzulassen - wieder analog zum Telefonnetz, oder ob dann die Bandbreite gleichmäßig für alle auf ein absolutes Minimum reduziert werden würde.

Aus technischer Sicht sind alle Aussagen, die in den vorigen Abschnitten aus RFCs angeführt wurden, ausschließlich als Empfehlungen zu sehen. Dies eröffnet bei der Implementierung der Richtlinien ausgesprochen große Spielräume. Unter diesem Aspekt ist klar, dass eine hohe Marktdurchdringung ohne konkrete Aussagen und Vorschriften bezüglich der Implementierung (konkrete Werte für Jitter, prozentual zugelassener Paketverlust) nicht zu erreichen ist. Zudem war noch nicht einmal die feste Behandlung der PHBs und Codepoints festgelegt. In dieser Hinsicht wurden einige Fortschritte erzielt, indem letztlich durch das EU-Projekt MUSE sogar klare Aussagen zu Puffergrößen [6] und Längen der Warteschlangen [1] gemacht wurden.

Von dem Ziel, das Internet im Kern simpel und relativ „dumm“ zu halten, muss bei der flächendeckenden Implementierung von DiffServ allerdings Abstand genommen werden. Jede weitere Berechnung, Aggregierung und (Neu-)Zuordnung von Daten ist unweigerlich mit einer gewissen erforderlichen Rechenleistung verbunden, sodass die Knoten sowohl intelligenter, als auch teurer und fehleranfälliger werden würden.

# 4. OVERPROVISIONING VS. QOS

Wie aus den vorhergehenden Abschnitten und insbesondere dem anfangs erwähnten Straßenverkehrszenario hervorgeht, ist QoS im Grunde nur dann notwendig, wenn das Best-Effort-Verfahren an seine Grenzen stößt. Dies ist genau dann der Fall, wenn die Straße nicht breit genug ist, wenn also auf das Netzwerk bezogen nicht genügend Bandbreite zur Verfügung steht. Erst dieser Engpass macht es überhaupt erforderlich, QoS-Maßnahmen zu ergreifen.

Nun kann argumentiert werden, dass eine großzügige Überdimensionierung der zur Verfügung gestellten Bandbreite jegliche Probleme bezüglich der Parameter, die in Sektion 2.1 erläutert wurden, vergleichsweise kostengünstig gelöst werden kann. Dies wirft jedoch zwei Probleme auf. Zum einen ist diese kostengünstige Erweiterung recht beschränkt, denn mit der Entstehung vieler neuer Anwendungen, besonders im Web 2.0, steigt der Bandbreitenbedarf rasant und kommt demnach auch schnell an seine physikalischen Grenzen. Müssen jedoch erst aufwändig neue Leitungen verlegt werden, explodieren die Kosten. Das zeigt also, dass Overprovisioning sinnvoll zunächst nur bei komplett neu zu entwerfenden und zu errichtenden Netzen möglich ist.

Dies führt jedoch zum zweiten Aspekt, der mit einer Überdimensionierung einhergehen würde: *Wie stark* kann man hier überdimensionieren? Es ist stets nur möglich, so hoch zu dimensionieren, wie es der aktuelle Stand der Technik und die Vorstellungskraft der Techniker erlaubt. Wer weiß schon, welche Anwendungen in den nächsten 5, 10 oder 20 Jahren existieren werden und wie viel Bandbreite sie beanspruchen? Schon jetzt werden große Bandbreiten für Consumer-Anwendungen wie hochaufgelöstes Fernsehen über IPTV benötigt, auch transferieren Firmen gigabyteweise Daten in die Cloud und zurück - was vor einigen Jahren noch völlig undenkbar gewesen wäre.

Dazu kommt der wirtschaftliche Aspekt. Die Überdimensionierung kostet natürlich zusätzlich Geld, was ohne unmittelbare Notwendigkeit aufgebracht werden muss. Dies widerspricht dem ökonomischen Prinzip.

Weiterhin können Netzprovider den Kunden „Best-Effort“ nicht als etwas Besonderes verkaufen. Der Ausbau der Infrastruktur kann nur bedingt auf die Endkundenpreise umgelegt werden, denn der Markt ist hart umkämpft. Setzt man hingegen auf QoS, ließen sich neue Premiumdienste schaffen. Sowohl Privat- als auch Geschäftskunden könnte man hier bevorzugte Behandlung gegen Aufpreis verschaffen und hätte so neue Geschäftsfelder im ansonsten innovationsarmen Providergeschäft erschlossen. Es muss jedoch berücksichtigt werden, dass eine gewisse Bandbreite immer für Notfälle freigehalten wird, beispielsweise für staatliche Belange. Inwiefern eine solche „Klassengesellschaft“ im Future Internet wünschenswert und mit den Grundsätzen des heutigen Internets vereinbar ist, soll im letzten Abschnitt diskutiert werden.

Insgesamt ist also zu sagen, dass Overprovisioning als die zunächst einfachere, aber auch „brachialere“ Lösung erscheint, während das Implementieren von QoS-Lösungen technisch anspruchsvoller und eleganter ist, auf vorhandener Infrastruktur aufsetzen kann, aber die Komplexität des Netzes deutlich erhöht.


doi: 10.2313/NET-2011-07-2_03

## 5. DENIAL OF SERVICE ALS GEFAHR FÜR QOS

Obwohl Best-Effort-Netzwerke im Allgemeinen einen stabilen und zuverlässigen Dienst bieten, kann es sein, dass sie im Fall von hoher Auslastung lediglich schlechte oder sogar gar keine Verbindungen mehr ermöglichen [11]. Die Relevanz von DoS-Attacken ist nur gegeben, weil es im Moment in Festnetzen keine Gebühren auf Bit/Byte-Basis gibt [12]. Müsste hier etwa paketweise bezahlt werden, was in Zeiten von Flatrates undenkbar ist, wäre die Gefahr eines Floodings mit Paketen weit geringer.
Um einen Dienstausfall zu vermeiden, kann man versuchen, die Auslastung statistisch vorherzusagen. Dies ist aber nur sehr schwer möglich, da die Netzlast sprunghaft ansteigen kann, wie beispielsweise bei den jüngsten Naturkatastrophen in Japan, wo die Bevölkerung der ganzen Welt nach Informationen sucht. Ein solcher Anstieg ist kaum von einer DoS-Attacke zu unterscheiden, denn solche Attacken können auch verteilt von Botnetzen initiiert werden. Es gibt hier laut Shalunov [11] viele ungeklärte Fragen, die den Umgang mit kompromittierten Routern oder Hosts betreffen, die etwa eine ganze Trafficklasse für sich beanspruchen oder sämtliche verfügbare Bandbreite reservieren. Möchte man dies verhindern, so muss der Anteil, der für den höherwertigen (nicht Best-Effort)-Traffic verwendet wird, anpassbar sein. Denn nur so kann flexibel auf hohe Last reagiert werden, wenn sichergestellt ist, dass es sich um keinen DoS-Angriff handelt. So werden dann Engpässe in der Best-Effort oder QoS-Klasse vermieden, ohne dass Pakete verloren gehen und die Leerkapazität der einen oder anderen Klasse wird sinnvoll genutzt.
Hier besteht natürlich die Möglichkeit, sich gewisse Qualitätsmerkmale bzw. deren Nichteinhaltung versichern zu lassen, oder aber die Provider müssten hier mit Geld-zurück-Garantien oder ähnlichen Zusicherungen punkten [13]. Das zieht wiederum das Problem der Nachweisbarkeit nach sich. Denn schließlich können die QoS-Zusicherungen nur im Falle extrem hoher Last getestet werden - und eine künstliche Überlastung kann auf keinen Fall im Interesse des Providers liegen. Deswegen ist es sehr schwer, hier Garantien zu geben, die auch mess- und spürbar sind.

## 6. QOS IM (FUTURE) INTERNET

Kann man nun Quality of Service vorbehaltlos in einem jetzigen oder einem zukünftigen Internet einführen? Das Future Internet kann einerseits als radikaler Neuanfang gedeutet werden oder aber in Form von inkrementellen Veränderungen aus dem nun vorhandenen Internet hervorgehen. Da ein radikaler Schnitt nicht praktikabel ist, müssten vorerst zwei „Internets“ parallel existieren. Lässt man aber diese erst parallel entstehen, könnten daraus untereinander nicht kompatible Netze hervorgehen und bestehen bleiben, sodass eine große Fragmentierung entsteht. Es ist außerdem nicht klar, wie hier der Traffic dann zwischen den beiden Netzen fließen sollte. Es soll ja kein unklassifizierter Verkehr aus dem alten Internet in das neue QoS-fähige Future Internet fließen. Um eine völlige Isolierung zu verhindern, und auch im Hinblick darauf, dass im Future Internet mit QoS ein neue Form der Abrechnung eingeführt werden muss, müssen providerübergreifende Vereinbarungen getroffen werden. Dies scheint schwer umsetzbar.
Denkt man hier an einen radikalen Neubeginn, könnte ein Peer-to-Peer-basierter Ansatz die Lösung sein. Wenn man den kompletten Aufbau des Internets dezentralisiert und nur noch Router teilnehmen lässt, die per Implementierung einen vereinheitlichten Standard für QoS-Maßnahmen umsetzen, können die Provider überflüssig werden. Durch den P2P-Ansatz entstünde dann ein riesiges, providerloses Netz unter einer einzigen DS-Domäne, womit grundsätzlich die Grenzknotenproblematik, insbesondere von DiffServ, gelöst wäre.
Geht man davon aus, dass QoS-Maßnahmen weitgehend selbstlos eingesetzt werden würden, so wie das in Grundsätzen der Idee des jetzigen Internets entspricht, ist es natürlich sinnvoll eine bestimmte Servicequalität garantieren zu wollen. Jedoch zeigt die Erfahrung, dass Premiumdienste oft mit einer Premiumbezahlung einhergehen und somit das Internet für den normalen Best-Effort-User unbenutzbar werden würde. Wenn mit höherwertigen Dienstleistungen mehr Geld verdient werden kann, wird die Standard-Leistung bald gekürzt werden bis hin zur Unbenutzbarkeit. Damit wird ein wichtiger Teil der modernen Kommunikation von vielen Menschen ferngehalten, was nicht mit dem Grundgedanken vereinbar ist, das Internet möglichst günstig allen Menschen zugänglich zu machen. Die Premiumservices, die noch Performance bieten würden, wären sehr teuer. Ein großer Teil von Bildungsmöglichkeiten in der modernen Gesellschaft würde eingeschränkt und den aufstrebenden Gesellschaftsteilen in anderen Ländern würde im wahrsten Sinne des Wortes der Anschluss gekappt, da sie sich keine hochwertige Dienstgüte leisten könnten.
Schafft man dagegen ein neues Internet mitsamt neuer physikalischer Beschaffenheit, ohne Rücksicht auf Kompatibilität zu vorhandenen Technologien, so scheint die Einführung von QoS realistischer. Zwar tritt auch hier das Problem auf, dass eine im QoS-Sinne „bessere“ Leitung auch teurer ist, jedoch gäbe es die Möglichkeit, Regionen mit schlechteren Leitungen aus dem QoS-unterstützten Gebiet auszunehmen. Dies trifft insbesondere auch auf Mobilfunkverbindungen zu, für die aufgrund physikalischer Gegebenheiten bestimmte Qualitätsmerkmale nicht garantiert werden können (z.B. Latenz).

## 7. ZUSAMMENFASSUNG UND AUSBLICK

Zusammenfassend kann gesagt werden, dass QoS für das zukünftige Internet eine sinnvolle Erweiterung darstellen kann. Mit DiffServ existiert eine gut erforschte Implementierungsmöglichkeit, es lassen sich viele Regelungen treffen, die jedoch an ein schwer zu etablierendes SLA gebunden sind. Insgesamt ist es jedoch schwierig, QoS für das zukünftige Internet zu etablieren. Zuerst gibt es einige infrastrukturelle Probleme. Beispielsweise ist es einfach nicht möglich, übergreifend für mobile und drahtgebundene Netze gleichermaßen hochwertige QoS-Kriterien anzusetzen. Latenz und Bandbreite der flächendeckend verfügbaren Mobilfunknetze lassen sich in keinster Weise mit der durch die Festnetzanbieter etablierten Netzstruktur vergleichen. Eine Festlegung auf einen geringen netzübergreifenden Standard ist auch keine Lösung, denn dieser ist oftmals sogar deutlich unter dem Niveau der Best-Effort-Lösungen in den Festnetzen anzusiedeln. Zudem haben sich die technischen Möglichkeiten wie DiffServ oder IntServ mangels früher Standardisierung und zu aufwändiger Implementierung nicht etabliert.


doi: 10.2313/NET-2011-07-2_03

Andererseits ist das Bedürfnis nach QoS in der Realität, außer in Zugangsnetzen, noch nicht vorhanden, da die Best-Effort-Leistungen der heutigen Netze als ausreichend gedeutet werden müssen. In Kombination mit Audio- und Videocodecs oder passenden Technologien für andere Einsatzfelder, die ihre Qualität an die vorhandene Bandbreite anpassen, ist auch mit dieser Methode eine absolut akzeptable Netzleistung zu erreichen.
Darüber hinaus müssen aber auch Probleme moralischer Art gelöst werden. Insbesondere das Thema Netzneutralität spielt hier eine zentrale Rolle. Wenn es schon verschiedene Klassen von Datenverkehr gibt, wer bestimmt, welche Arten von Traffic zu welcher Klasse gehören? Viel komplexer erscheint hier auch, dass der Traffic dann nicht nur nach Art klassifiziert werden könnte, sondern auch nach Organisation/Firma. Wer erhält dann ein Vorrecht auf priorisierten Internetverkehr? Wer wird im Extremfall ganz von den Backbones abgetrennt? Kann man anderen Organisationen eine großen nutzbaren Anteil von Internetkanälen „wegkaufen“?
Solange diese Probleme nicht geklärt sind, ist es zumindest zweifelhaft, dass sich QoS flächendeckend etablieren kann und muss. Der Erfindergeist der Ingenieure hat gezeigt, dass es zunehmend möglich wird, immer mehr Datenverkehr über vorhandene oder günstige Leitungen zu transportieren. Durch Beseitigung der Flaschenhälse des Internets könnte so ein großzügiges Overprovisioning und die Vermeidung der Worst-Case-Vollauslastung am Ende doch die Lösung aller QoS-Überlegungen sein.

# 8. LITERATUR


[1] PlaNetS QoS Solution. http://www.medea-planets.eu/QoSsolution.php?page=solution (13.03.2011, 15.30 Uhr).

[2] P. Almquist. Type of Service in the Internet Protocol Suite. RFC 1349, 1992.

[3] D. Bricklin. Why We Don’t Need QOS: Trains, Cars, and Internet Quality of Service. http://www.bricklin.com/qos.htm (09.03.11, 22.45 Uhr).

[4] Cisco Systems. *DiffServ – The Scalable End-to-End QoS Model*, 8 2005.

[5] Cisco Systems, USA. *Overview of DiffServ for QoS*, Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 edition, 4 2007.

[6] A. Foglar. Die QoS Lösung des MUSE Projekts - Europa definiert das Breitbandnetz der nächsten Generation (The QoS Solution of the MUSE Project). *it - Information Technology*, 48(5):282–, 2006.

[7] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597, 1999.

[8] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598, 1999.

[9] S. Keshav. *An engineering approach to computer networking: ATM networks, the Internet, and the telephone network*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[10] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474, 1998.

[11] S. Shalunov and B. Teitelbaum. Quality of Service and denial of service. In *Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?*, RIPQoS ’03, pages 137–140, New York, NY, USA, 2003. ACM.

[12] B. Teitelbaum and S. Shalunov. Why Premium IP Service Has Not Deployed (and Probably Never Will). Internet2 QoS WG informational doc, May 2002.

[13] B. Teitelbaum and S. Shalunov. What QoS research hasn’t understood about risk. In *Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?*, RIPQoS ’03, pages 148–150, New York, NY, USA, 2003. ACM.

[14] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC 2210, 1997.

[15] W. Zhao, D. Olshefski, and H. Schulzrinne. Internet Quality of Service: an Overview. Technical report, 2000.




doi: 10.2313/NET-2011-07-2_03