Abschlussbericht

Differentiated Services zum Abruf digitaler Dokumente aus elektronischen Bibliotheken – DiDelBi –

im Rahmen des Schwerpunktprogramms
Verteilte Verarbeitung und Vermittlung digitaler Dokumente
– V^3D^2 –

1. Überblick

1.1 Bearbeitungszeitraum

24 Monate: 1. Januar 2001 bis 31. Dezember 2002 (Phase II des SPPs).

1.2 Zielsetzung und Einordnung in das SPP V^3D^2

Ziel des Forschungsvorhabens war die Konzipierung und Implementierung einer geeigneten Kommunikationsunterstützung zum Abruf digitaler Dokumente. Dabei sollte – je nach Typ der Dokumente – die Bereitstellung einer kalkulierbaren Dienstgüte ermöglicht werden. Aus Sicht des Schwerpunktprogramms stellt dieses Forschungsvorhaben die Basiskommunikationsinfrastruktur bereit, die von anderen Projekten benötigt wird.

Im Rahmen des Vorhabens wurden die so genannten Differentiated Services, auf denen derzeitige Bemühungen zur Bereitstellung von Dienstgüte im Internet fußen, analysiert und geeignet für den Einsatz in digitalen Bibliotheken erweitert. Hierzu gehörte die Anpassung von so genannten Bandwidth Brokern, die in jeder Differentiated Services Domäne vorhanden sind, eine Zugangskontrolle realisieren und die Bereitstellung der geforderten Ressourcen steuern. Dies musste im Umfeld digitaler Bibliotheken so gestaltet werden, dass auf dem Rückweg, also von der Bibliothek zum Kunden, die entsprechende Dienstgüte bereitgestellt werden kann. Hierbei fand insbesondere die besondere Problematik einer asymmetrischen Kommunikationsbeziehung zwischen Besucher und Bibliothek Berücksichtigung.

1.3 Gliederung

Das Projekt umfasste die folgenden Arbeitsschwerpunkte:

- Dienstgüteanforderungen – Simulation neuer Dienste in Diffserv
- Dienstgütemanagement – Bandwidth Broker
- Integration von Dienstgüte – Nutzung im Umfeld digitaler Bibliotheken

- Prototyp – Evaluierung der entwickelten Konzepte

Die Analyse der Interaktion mit einer digitalen Bibliothek lieferte Anforderungen an die bereitzustellende Dienstgüte, so dass nach entsprechender Klassifikation eine Abbildung auf existierende Dienste bzw. die Definition neuer Dienste im Rahmen der Diffserv-Architektur erfolgen konnte, die auch simulativ untersucht wurden. Daneben wurde die allgemein als Bandwidth Broker (BB) bezeichnete Managementinstanz der Differentiated Services (Diffserv, DS) Dienstgütearchitektur daraufhin untersucht, wie sich asymmetrische Kommunikationsbeziehungen einbeziehen lassen, wobei besonderer Wert auf eine Unterstützung von Diensten nicht nur innerhalb einer Domäne sondern internetweit von Ende zu Ende zu ermöglichen und dabei weiterhin skalierbar zu bleiben. Ein geeigneter Ablauf zur Auswahl und Anforderung von Dienstgüte wurde entworfen, um die bereitstehende Dienstgüteinfrastruktur für digitale Bibliotheken nutzen zu können. Hierdurch ergaben sich neue Anforderungen an das Dienstgütemanagement, die dort entsprechend als Erweiterung berücksichtigt und zur praktischen Überprüfung der Konzepte prototypisch implementiert wurden. Insgesamt steht damit ein vollständiges Konzept zur dienstgüteunterstützten Kommunikation im Rahmen digitaler Bibliotheken bereit.

Die folgenden Abschnitte präsentieren die entwickelten Konzepte und gefundenen Ergebnisse im einzelnen.

2. Dienstgüteanforderungen

Im Rahmen von digitalen Bibliotheken lassen sich eine Reihe von Anforderungen an eine bereitzustellende Dienstgüte ableiten, die im nächsten Abschnitt 2.1 vorgestellt werden sollen. Basierend hierauf kann eine Abbildung auf Dienste, wie sie bereits im Rahmen der Differentiated Services Dienstgütearchitektur standardisiert wurden, stattfinden. Dort, wo geeignete Dienste fehlten, wurden im Rahmen des Forschungsvorhabens neue Dienste entworfen und simulativ evaluiert (Abschnitt 2.2).

2.1 Relevante Szenarien – Klassifizierung

- **Sprachkommunikation (VoIP)**: interaktiver CBR-Verkehr
  ⇒ möglichst kleine Verzögerung, konstante garantierte Rate

- **Recherche**: transaktionsorientiert, interaktiv
  ⇒ möglichst kleine Verzögerung von Paketbursts

- **Streaming**: konstante (mittlere) Rate, Schwankungen akzeptabel
  (abgeschwächte Interaktivität erlaubt größere Empfangspuffer)
  ⇒ garantierte Bandbreite über gewissen Zeitraum (Puffergröße) gemittelt

- **Abonnement/Datenabgleich**: Hintergrundverkehr niedriger Priorität mit gewissem (endlichen) Zeithorizont
  (soll anderweitig nicht benötigte Ressourcen effektiv ausschöpfen)
  ⇒ niedrigste Priorität aller Klassen, aber mit gesichertem Mindestanteil an der Gesamtbandbreite

- **Download**: je nach gewünschtem Verhalten wie Streaming, Low-Cost-Zugang oder Datenabgleich
  ⇒ Anforderungen siehe jeweils dort

- **Low-Cost-Zugang**: möglichst geringe Kosten
  ⇒ keine Dienstgüteunterstützung

Zur Erläuterung sei im Folgenden der virtuelle Besuch eines Bibliotheksnutzers näher betrachtet, wodurch sich eine Reihe von typischen Tätigkeiten identifizieren lassen. So steht am Anfang eines Bibliotheksbesuchs eine Recherche, die sich als *transaktionsorientiert* und *interaktiv* charakterisieren lässt. Wurde ein passendes Dokument gefunden, muss dies auf den Rechner des Nutzers transferiert werden. Je nach Medientyp kann dies ein Download oder Streaming sein. Auch wenn ein Download so schnell wie möglich abgeschlossen sein sollte, kann es je nach der Netzwerkanbindung des Nutzers wünschenswert sein, die Downloadrate einzuschränken, beispielsweise wenn parallel eine weitere Recherche stattfinden soll. Die Rate beim Streaming leitet sich direkt aus der Kodierung des Dokuments ab.

Der Push-Dienst bzw. das Abonnement stellt den zweiten Anwendungsfall dar. Hier verschickt die Bibliothek ohne explizite Anforderung durch den Nutzer regelmäßig Aktualisierungen von abonnierten Dokumenten oder neu hinzugekommene Dokumente, die gewisse vom Nutzer bestimmte Auswahlkriterien erfüllen. Als Hintergrundverkehr sind diese Dokumente kaum zeitkritisch, vielmehr darf eine u. U. im Vordergrund laufende Sitzung nicht beeinträchtigt werden. Trotzdem muss ein zuverlässiger Transport gewährleistet sein, eventuell mit einer definierten maximalen Übertragungszeit. Zwar nicht direkt mit einem virtuellen Bibliotheksbesuch verknüpft, im Rahmen digitaler Bibliotheken aber dennoch von Interesse ist der Datenabgleich zwischen replizierten Bibliotheken, der dieselben Dienstgüteanforderungen wie der Pushdienst besitzt.

Nicht direkt mit dem Abruf von Dokumenten verbunden, aber dennoch für einen Bibliotheksbesucher von Bedeutung ist die Möglichkeit, sich bei der Recherche von einem Mitarbeiter der Bibliothek unterstützen zu lassen. Hier bietet die fernmündliche Kommunikation auf Basis der von Besucher wie Mitarbeiter einsehbaren Rechereumgebung eine geeignete Basis. Sprachkommunikation (Voice over IP, VoIP) findet überwiegend mit konstanter Bitrate statt, so dass sich der generierte Verkehr als strenger CBR-Datenstrom (Constant Bit Rate) charakterisieren lässt.

Neben diesen durch Diffserv zu unterstützenden Anwendungen ist auch der nicht Dienstgüteunterstützte Transport als Best Effort dort sinnvoll, wo die Interaktion mit der Bibliothek zu kurz ist, als dass der Mehraufwand einer Reservierung einen Performancevorteil bringt oder einfach die Mehrkosten einer Reservierung unerwünscht sind (Low-Cost Zugang).

2.2 Dienstgüteklassen

Nachdem die zu unterstützenden Verkehrscharakteristika bekannt sind, können diese auf entsprechende Diffserv-Dienste abgebildet werden. Durch die IETF-Arbeitsgruppe „Diffserv“ wurden zwei Paketweiterleitungsverhalten (Per-Hop Behavior, PHB) standardisiert, das Expedited Forwarding (EF) PHB [DCBB$^+$02] und die Assured Forwarding (AF) PHB Group [HBWW99]. Mit Hilfe von EF kann eine konstante Datenrate bei minimaler Verzögerung und weitgehend ohne Paketverluste bereitgestellt werden. Mittels AF kann eine Mindestrate garantiert werden mit der Möglichkeit, darüber hinaus weitere, evtl. noch verfügbare Ressourcen nutzen zu können, wobei bezüglich der Verzögerung keine Angaben gemacht werden.

Da weder zur Unterstützung burstartigen Verkehrs noch für niederprioren Hintergrundverkehrs geeignete Weiterleitungsverhalten existieren, wurden hierfür zwei neue PHBs entwickelt: Das Quick Forwarding (QF) PHB für den Transaktionssupport und das Lower Effort (LE) PHB für Hintergrundverkehr. Ein PHB selbst stellt noch keinen Dienst dar. Die Auswahl bzw. Kombination von PHBs mit einer geeigneten Verkehrsbeeinflussung an den Domänengrenzen ist Gegenstand eines Per-Domain Behaviors (PDB) [NiCa01]. Es beschreibt das erhaltene Verhalten von Domänenrand zu Domänenrand und berücksichtigt hierbei explizit die Folgen von Aggregations- und Deaggregationseffekten im Inneren der Diffserv-Domäne. Die PHBs LE und QF [Küfn03] werden daher in Form von PDBs als Informational RFCs veröffentlicht

werden. Da bisher noch keine PDBs standardisiert wurden, wird das LE PDB [BlNW02] damit voraussichtlich das erste Diffserv Per-Domain Behavior werden.

Im Rahmen von digitalen Bibliotheken sind demnach zusammen mit dem bisherigen Best Effort „Dienst“ insbesondere folgende Dienste von Interesse:

- Verzögerungsbegrenzt mit minimalem Paketverlust bei konstanter Datenrate (Expedited Forwarding)
- Bursttransport mit minimaler Verzögerung von Transaktionen (Quick Forwarding)
- Garantierte Bandbreite ohne besondere Verzögerungsanforderungen (Assured Forwarding)
- Niederpriorer (unterhalb von Best Effort) Hintergrundverkehr, aber mit garantierter minimaler Rate (Limited Effort)
- Ohne Dienstgüteunterstützung, keine Garantien (Best Effort)

3. Dienstgütemanagement

Zentrales Designziel der im Rahmen des Forschungsvorhabens weiterentwickelten Managementinstanz (häufig auch als Bandwidth Broker bezeichnet) war, garantierte Ende-zu-Ende-Dienste zu ermöglichen, ohne dabei die durch Diffserv erreichte Skalierbarkeit des Datenpfads im Kontrollpfad wieder zunichte zu machen. Erreicht wurde dies durch das Prinzip der dynamischen Aggregation von Reservierungen, dass das Aggregationsprinzip vom Datenpfad auf die Kontrollebene überträgt. Im Folgenden soll zuerst das grundlegende Funktionsprinzip von Diffserv und das des Dienstgütemanagements erläutert werden unter besonderer Beachtung der Randbedingungen, die durch die Struktur des Internets und die Architektur von Diffserv vorgegebenen werden. Darauf aufbauend wird in das Prinzip der dynamischen Aggregation eingeführt sowie das zu seiner Umsetzung nötige Signalisierungsprotokoll vorgestellt.

3.1 Grundlagen des Dienstgütemanagements

Das heutige Internet gliedert sich in so genannte „Autonome Systeme“ (AS). Ein AS lässt sich definieren als „eine Menge von Routern, die unter derselben Administration stehen“. Nach außen hin stellt sich das AS als eine homogene Einheit dar, deren interner Aufbau und insbesondere deren technische Verwaltung wie beispielsweise das verwendete interne Routingprotokoll oder das zur Routerkonfiguration eingesetzte Netzwerkmanagement, verborgen bleibt. Die in Diffserv vorgesehene Untergliederung in einzelne Diffserv-Domänen lässt sich damit ideal auf das Internet mit seinen Autonomen Systemen abbilden, indem jedes AS eine Diffserv-Domäne darstellt. Bei großen Autonomen Systemen ist es auch möglich, dass sich ein AS in mehrere Domänen untergliedert, jedoch nicht umgekehrt.

Diffserv erzielt Dienstgüte, indem Pakete in unterschiedliche Verhaltensaggregate (Behavior Aggregates, BA) eingeteilt werden. Dazu besitzt jedes IP-Paket das DS-Feld (ehemals Type-of-Service-Feld), das den so genannten Diffserv Codepoint (DSCP) enthält. Über den Codepoint wird in jedem Router das entsprechende Weiterleitungsverhalten (Per-Hop Behavior, PHB) selektiert, so dass die Pakete bei ihrer Weiterleitung differenziert und so mehr oder weniger bevorzugt behandelt werden können. Jeder Diffserv-Router besitzt also einen Paketklassifizierer, den BA-Klassifizierer, der den DSCP auswertet, sowie aktive Warteschlangenmechanismen (weighted/gentle RED, EDF) und Bedienstrategien (WFQ, DRR, PQ, CBQ), um eine differenzierte Paketbehandlung zu ermöglichen. Die aggregierte Behandlung in Form

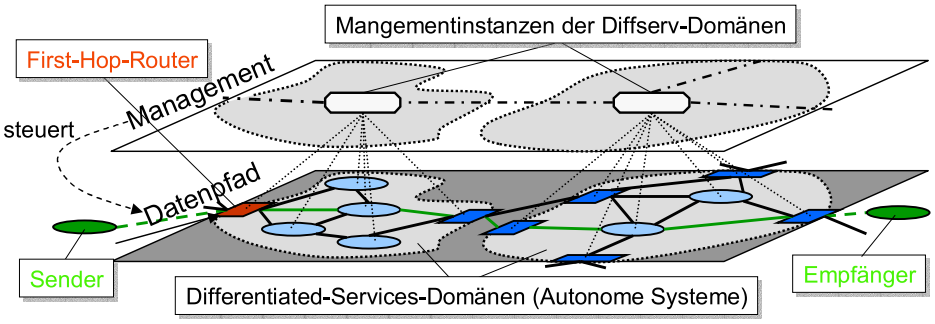


Abbildung 3.1 Zentrales Dienstgütemanagement mit einer (logischen) Managementinstanz je Diffserv-Domäne. Jede Instanz konfiguriert nur die Router der eigenen Domäne. Die Pfade der Signalisierungnachrichten und der Datenpakete sind entkoppelt (path-decoupled).

von BAs hat jedoch zur Folge, dass ein Datenstrom, der in ein BA klassifiziert wurde, obwohl nicht mehr genügend Ressourcen zu seiner Aufnahme vorhanden waren, die Qualität aller anderen Datenströme in diesem BA – und je nach Bedienstrategie darüber hinaus auch die Qualität weiterer BAs – beeinträchtigt. In Folge der Überlast kommt es zu wachsenden und schließlich überlaufenden Warteschlangen und damit zu Paketverlusten, so dass die eigentlich garantierte Dienstgüte nicht mehr gewährleistet ist.

Um solche Situationen zu vermeiden, muss eine Zugangs- und Nutzungskontrolle durchgeführt werden, beides Aufgaben des Dienstgütemanagements. Bevor eine durch Diffserv qualitätsgesicherte Kommunikation beginnen kann, muss deshalb zuerst beim Dienstgütemanagment eine Reservierung vorgenommen werden, um eine „Freischaltung“ des eigenen Datenstroms zu erreichen. Die entwickelte Managementarchitektur erlaubt dabei Ende-zu-Ende-Reservierungen über mehrere Domänen bzw. Autonome Systeme hinweg für den gesamten Datenpfad vom Sender bis zum Empfänger.

Die entwickelte Managementinstanz namens DSDM (Diffserv Domain Manager) entspricht in ihren Aufgaben einem Bandwidth Broker. Sie setzt damit das Konzept des domänenzentralen Dienstgütemanagements um (vgl. Abbildung 3.1). Diese Designentscheidung bedingt, dass die an der Signalisierung zur Etablierung einer Reservierung beteiligten Knoten nicht jene sind, über die letztlich der Datenpfad der einzurichtenden Reservierung verläuft. Infolgedessen folgen die Signalisierungsnachrichten auch nicht dem späteren Datenpfad, was als pfadentkoppelte Signalisierung (Path-Decoupled Signaling) bezeichnet wird – im Gegensatz zum Path-Coupled Signaling, wie es beispielsweise RSVP (Ressource Reservation Protocol [BZBH+97]) verwendet. Lediglich auf der Ebene von Diffserv-Domänen verlaufen Signalisierungspfad und späterer Datenpfad parallel.

Der wesentliche Vorteil des zentralen Managements liegt in der durch die Entkoppelung von Signalisierung und Datenpfad gewonnenen Flexibilität. Das Dienstgütemanagement kann dadurch eingeführt und weiterentwickelt werden, ohne dass Modifikationen an allen Routern nötig werden. Darüber hinaus wird eine weitere Belastung der Router im Internet vermieden. Bereits heute müssen im Interdomain-Routing viele der BGP-Routen für kleine Netzwerke herausgefiltert[1] werden, um die Anzahl der Routingänderungen auf eine noch bearbeitbare Größe zu reduzieren. Als Folge der vom Datenpfad entkoppelten Signalisierung kann darüber hinaus der Initiator einer Reservierung nicht nur Sender oder Empfänger, sondern auch eine dritte, an der eigentlichen Datenkommunikation unbeteiligte Partei sein. Nicht nur dies

[1] vgl. BGP-Tabellen auf `http://bgp.potaroo.net/`

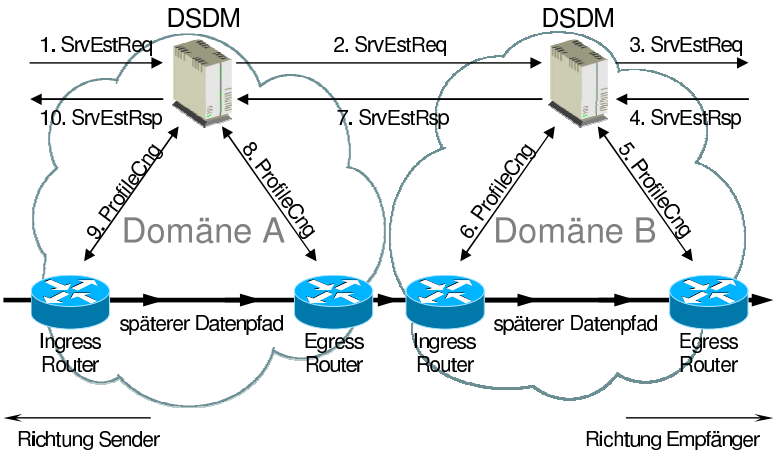


Abbildung 3.2 Abfolge der Signalisierungsnachrichten zum Aufbau einer Reservierung am Beispiel von zwei benachbarten Transitdomänen.

erfordert die Ankopplung an eine AAA-Architektur (Authorization, Authentication, Accounting [LGGV+00]), um Missbrauch zu verhindern und Nutzungsrechte sinnvoll verwalten zu können. Die AAA-Infrastruktur, wegen der benötigten Datenbanken gewöhnlich ebenfalls zentralisiert ausgeführt, lässt sich so auf natürliche Weise mit einer zentralen Managementinstanz verbinden.

Reservierungen sind immer unidirektional. Für eine TCP-Verbindung sind damit immer zwei Reservierungen nötig, eine für jede Richtung. Bidirektionale Reservierungen können jedoch dadurch unterstützt werden, dass diese von der ersten Managementinstanz – jene, an die der Initiator seinen Reservierungswunsch signalisiert hat – in zwei unidirektionale Reservierungen aufgesplittet wird. Anschließend werden diese dann völlig getrennt voneinander weiterbearbeitet. Der Grund für diese Umsetzung liegt in der *Asymmetrie* der Pfade im Internet, verursacht durch die unterschiedliche – asymmetrische – Bewertung von Pfaden auf Ebene der Autonomen Systeme. Verursacht wird dies vor allem durch die finanziellen Interessen der einzelnen Betreiber von Autonomen Systemen mit dem Ziel der Kostenminimierung [Paxs97]. Als Folge der Asymmetrie ist der Aufbau einer Reservierung nur beginnend beim Sender möglich. Der Grund hierfür ist in der Eigenschaft des Protokolls IP zu suchen, dass durch seine zustandslose Paketweiterleitung, basierend lediglich auf den im IP-Kopf eingetragenen Adressen, an jedem Router nur der nächste Router stromabwärts ermittelt werden kann, nicht jedoch der vorhergehende Router, von dem es empfangen werden wird. Entsprechend ist die Bestimmung des zukünftigen Pfads, dem ein Datenstrom durch das Internet folgen wird, nur beginnend beim Sender stromabwärts in Richtung des Empfängers möglich.

Eine Reservierungsanfrage durchläuft so beginnend in der Domäne des Senders stromabwärts jede auf dem Pfad zum Empfänger liegende Domäne (Abbildung 3.2). Dabei überprüft die Managementinstanz jeder einzelnen Domäne, ob entlang des Teilpfades durch ihr eigenes Netz genügend Ressourcen für die neu einzurichtende Reservierung zur Verfügung stehen und keine weiteren Gründe wie eine fehlende Berechtigung dagegen sprechen. Wird die Reservierung akzeptiert, werden die benötigten Ressourcen als belegt gekennzeichnet und die Anfrage an die nächste Domäne stromabwärts weitergeleitet, bis alle Domänen einschließlich der letzten Domäne, in der sich der Empfänger des zukünftigen Datenstroms befindet, die Reservierung akzeptiert haben. Daraufhin passt die Managementinstanz der Empfängerdomäne die Verkehrsprofile ihres Grenzrouters zur Nachbardomäne stromaufwärts entsprechend der hinzugekommenen Reservierung an und bestätigt den Erfolg zur Nachbardomäne stromaufwärts. Diese wiederum konfiguriert ebenso in ihren betroffenen Grenzroutern die Verkehrsprofile um

und signalisiert weiter stromaufwärts. Ist schließlich wieder die Domäne des Senders erreicht, bestätigt deren Managementinstanz dem Initiator der Reservierung die erfolgreiche Einrichtung, woraufhin dieser dem eigentlichen Sender mitteilen kann, dass mit der Übertragung von Daten beginnen kann bzw. dass ab diesem Zeitpunkt die Übertragung qualitätsgesichert ist. Sollte eine Domäne längs des Pfades nicht in der Lage sein, die benötigten Kapazitäten bereitzustellen, wird nicht weiter stromabwärts signalisiert, sondern von ihrer Managementinstanz sofort stromaufwärts die Ablehnung der Reservierung mitgeteilt.

Eine wie eben beschriebene Reservierung bis hinunter auf einzelne Datenströme führt zwangsläufig zu Skalierbarkeitsproblemen, sowohl im Hinblick auf die auszutauschenden Signalisierungsnachrichten und den in Grenzroutern vorzunehmenden Profilanpassungen, als auch hinsichtlich der durch eine Managementinstanz zu haltenden Zustände. Dieser Per-Flow State tritt zwar nicht im Datenpfad sondern nur beim Dienstgütemanagement, also im Kontrollpfad, auf, führt dort jedoch zu denselben schlechten Skalierbarkeitseigenschaften, an der der Internet-weite Einsatz der Integrated Services und seines Signalisierungsprotokolls RSVP gescheitert ist. Um die Skalierbarkeit trotz Unterstützung von expliziten Reservierungen pro Datenstrom Ende zu Ende zu gewährleisten, wurde deshalb im Rahmen des Forschungsvorhabens das Konzept der dynamischen Aggregation [Bles02] entworfen, das im folgenden Abschnitt näher erläutert werden soll.

3.2 Dynamische Aggregation

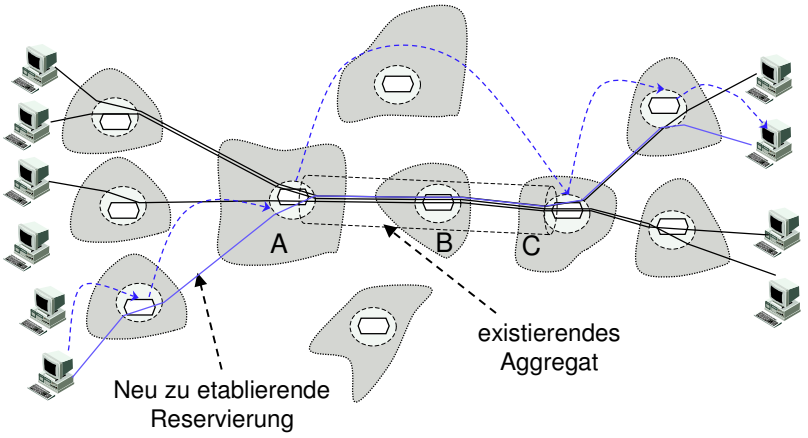
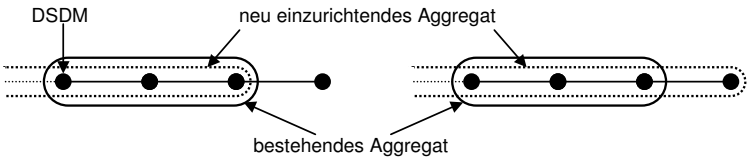
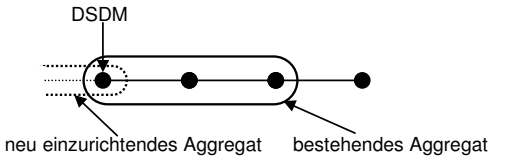


Abbildung 3.3 Aggregation von Reservierungen reduziert die Signalisierungsnachrichten und Zustände in der Kontrollebene.

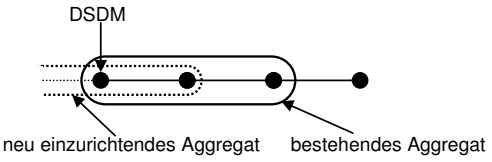
Zur Erläuterung des Prinzips der Aggregation betrachte man Abbildung 3.3. Zu einem Aggregat zusammengefasst werden können jene Reservierungen von Datenströmen, die einen gemeinsamen Teilpfad auf ihrem Weg durch das Netzwerk besitzen. Die Aggregation arbeitet dabei auf Ebene der Diffserv-Domänen bzw. der Autonomen Systeme, d. h. ein Knoten im Graph entspricht einer Domäne mit dem zugehörigen DSDM. Im Beispiel laufen drei Datenströme über die Domänen A, B und C, die zu einem Aggregat von A nach C zusammengefasst wurden. Dadurch sieht die mittlere Domäne nur noch eine Reservierung, die des Aggregats, und nicht mehr die drei in ihm enthaltenen Einzelreservierungen. In Folge dessen hat sich die Anzahl an zu haltenden Zuständen beim DSDM der Domäne B von drei für die Einzelreservierungen auf eine für das Aggregat reduziert. Die DSDMs an den Aggregatenden müssen jedoch neben den Einzelreservierungen noch einen weiteren Zustand, den des Aggregats halten, weshalb die Einrichtung erst ab einem gewissen Schwellenwert sinnvoll ist. Weil sich die meisten Aggregatenden in den Randdomänen finden, wo die Gesamtlast im Vergleich zu den großen Transitdomänen des Internets eher gering ausfällt, führt diese zusätzliche Last dort



(a) Fall 1: Bildung eines längeren Aggregats



(b) Fall 2: Hinzufügen eines weiteren Aggregats



(c) Fall 3: Konfliktfall

Abbildung 3.4 Verschiedene Fälle bei der Aggregatneubildung.

jedoch nicht zu Leistungsengpässen. Vielmehr setzt diese Lastverschiebung die Philosophie von Diffserv fort, Skalierbarkeit durch eine Entlastung des Kernnetzes zu erzielen.

Betrachtet man die Etablierung einer weiteren Reservierung, die gemäß Abbildung 3.3 auch entlang des Pfades von A nach C verläuft, kann diese im besten Fall ohne die Beteiligung von DSDM B eingerichtet werden (gestrichelte Pfeilbogen repräsentieren die Signalisierung), dann nämlich, wenn das Aggregat noch genügend freie Ressourcen besitzt, um die neue Reservierung aufzunehmen. Bei der Einrichtung eines Aggregats sollte demnach etwas mehr reserviert werden als eigentlich für die drei Einzelreservierungen nötig gewesen wäre. Ebenso erfolgt der umgekehrte Vorgang, das Verkleinern der Aggregatkapazität, nicht sofort mit dem Abbau einer enthaltenen Einzelreservierung, sondern hysteretisch verzögert, wenn die ungenutzten Ressourcen im Aggregat einen gewissen Schwellenwert erreichen. Durch diese Form von Überreservierung lässt sich die Aggregatsanpassung von den Änderungen der enthaltenen Reservierungen entkoppeln, und man sogt so für eine erhebliche Reduzierung von Aggregatanpassungsvorgängen, die ja von allen vom Aggregat passierten Transitdomänen bearbeitet werden müssen.

Im optimalen Fall bilden sich im Kernnetz große und stabile Aggregate aus, die nur von Zeit zu Zeit angepasst werden. Dortige Transitdomänen halten nur noch wenige Zustände für die wenigen Aggregate und bearbeiten nur noch wenige Änderungsnachrichten, unberührt davon, dass innerhalb der Aggregate ein ständiges „Kommen und Gehen“ herrscht.

Da Aggregate für Transitdomänen nichts anderes als Einzelreservierungen darstellen, können Aggregate auch geschachtelt werden. Abbildung 3.4 zeigt einige mögliche Fälle und auch einen Konfliktfall. Fall 3.4(c) ist nicht möglich, weil der mittlere DSDM, an dem das neue Aggregat enden soll, keine Kenntnis mehr über die im existierenden Aggregat enthaltenen Einzelreservierungen hat. Jene Einzelreservierungen, die bereits im alten Aggregat zusammengefasst waren und nun auch Teil des neu zu bildenden Aggregats werden sollen, entfallen daher als Kandidaten für das neue Aggregat, wodurch jedoch typischerweise nicht mehr genug Einzelreservierungen vorhanden sind, als dass sich die Neueinrichtung lohnen würde.

Dynamisch heißt die Aggregation, weil die Einrichtung der Aggregate selbständig anhand von Schwellenwerten (Anzahl von Einzelreservierungen, minimale Länge) erfolgt, ihre Kapazität mitwächst und -schrumpft, Reservierungen automatisch bei der Neueinrichtung längerer Ag-

gregate von kürzeren alten in das neue umgeschichtet werden und Aggregate mit der letzten Einzelreservierung auch wieder selbständig entfernt werden.

3.3 Signalisierungsprotokoll

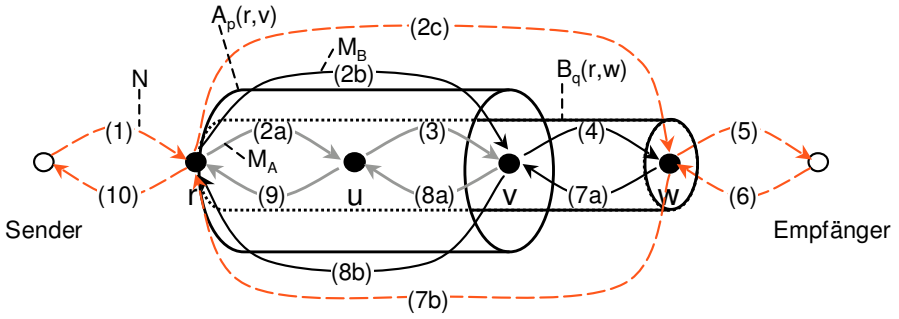


Abbildung 3.5 Änderung zweier Aggregate durch den DSDM im AS r aufgrund einer im längeren Aggregat enthaltenen bzw. neu hinzukommenden Reservierung.

Zusammen mit dem DSDM wurde das Signalisierungsprotokoll DMSP (Domain Manager Signaling Protocol) entwickelt. Es ist ein binär kodiertes Protokoll, dass zur weiteren Beschleunigung des Reservierungsaufbaus mittels Weiterleitungs- und Antwortwartebedingung voneinander abhängige Reservierungsvorgänge so weit wie möglich parallelisiert.

Zur Verdeutlichung der Arbeitsweise der Wartebedingungen in Nachrichten ist in Abbildung 3.5 eine Situation illustriert, in der ineinander geschachtelte Aggregate erhöht werden müssen. Nach Erhalt einer Reservierungsanfrage N im AS r (1), möchte der dort zuständige DSDM die Reservierung in das Aggregat $B_q(r, w)$ integrieren, dessen Kapazität dazu aber erhöht werden muss. Weil das Aggregat selbst in dem kürzeren Aggregat $A_p(r, v)$ enthalten ist, muss auch geprüft werden, ob dessen Kapazität ebenfalls erhöht werden muss, was im Beispiel der Fall sei. Die jeweiligen Signalisierungsvorgänge werden jedoch praktisch gleichzeitig gestartet: die Erhöhung von Aggregat $A_p(r, v)$ durch Nachricht M_A (2a), die Erhöhung von Aggregat $B_q(r, w)$ durch Nachricht M_B (2b) sowie die Einrichtung der in $B_q(r, w)$ zu aggregierenden Reservierung durch Nachricht N (2c). Die richtige Bearbeitungsreihenfolge wird nun durch die folgenden Bedingungen gewährleistet:

- Der Reservierungsanfrage N enthält eine Weiterleitungswartebedingung auf Nachricht M_B im AS w
- Die Aggregatanpassungsanfrage M_B enthält eine Weiterleitungswartebedingung auf Nachricht M_A im AS v sowie eine Antwortwartebedingung auf die Antwort zu Anfrage N
- Die Aggregatanpassungsanfrage M_A enthält eine Antwortwartebedingung auf die Antwort zur Anfrage M_B

In Abbildung 3.6 ist ein Weg-Zeit-Diagramm dargestellt, das ebenfalls einen möglichen Vorgang zur Einrichtung der Reservierung in Verbindung mit der Erhöhung zweier Aggregate aus dem vorigen Beispiel darstellt. Aus Darstellungsgründen wurden jedoch die in Abbildung 3.5 nahezu gleichzeitig stattfindenden Vorgänge (2a)-(2c), sowie (7a)/(7b) und (8a)/(8b) zeitlich weiter voneinander entfernt dargestellt und zur besseren Verdeutlichung der Wirkungsweise der eingesetzten Wartebedingungen zusätzlich innerhalb jeder Gruppe in ihrer Reihenfolge vertauscht, also in Abbildung 3.6 in der Reihenfolge 1, 2c, 2b, 2a, 3, 4, 5, 6, 7b, 7a, 8b, 8a,

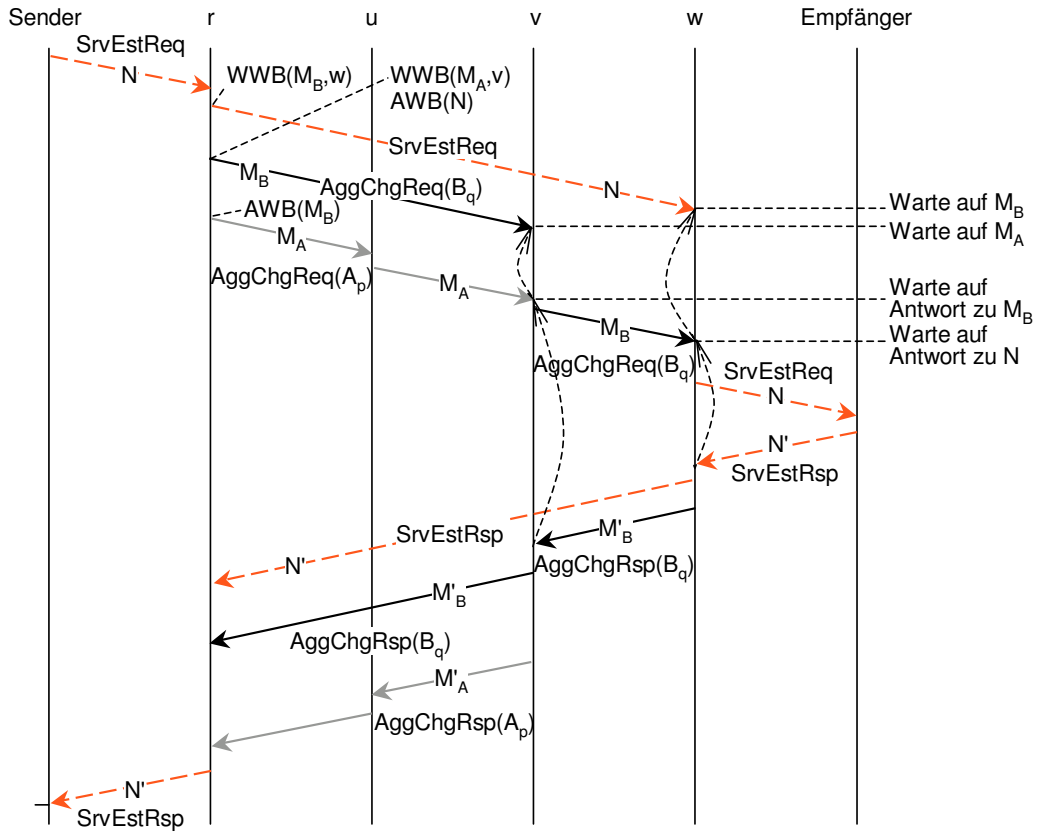


Abbildung 3.6 Weg-Zeit-Diagramm zur Änderung zweier geschachtelter Aggregate durch den DSDM im AS r.

9 und 10) dargestellt. Die vertikalen gestrichelten Pfeilbögen deuten die Erfüllung der Wartebedingungen durch andere Nachrichten an, so dass der zugehörige Signalisierungsvorgang am Beginn des Pfeilbogens fortgesetzt werden kann.

DMSP wird nicht nur auf der Netzwerk-Netzwerk-Schnittstelle, sondern auch an der Benutzer-Netzwerk-Schnittstelle eingesetzt. Das Protokoll für die Benutzer-Benutzer-Signalisierung ist dagegen applikationsspezifisch. Die in diesem Forschungsvorhaben hierfür entworfenen Erweiterungen von HTTP (Hypertext Transfer Protocol) und HTML (Hypertext Markup Language) werden in Abschnitt 4.2 vorgestellt.

3.4 Prototyp DSDM

Die prototypische Implementierung einer integrierten Ressourcenverwaltung für Diffserv-Netze wurde in der Diplomarbeit von Jochen Katz [Katz00] begonnen und im Rahmen von DiDelBi durch Stefan Jansen [Jans03] vollständig überarbeitet, insbesondere, um die Weiterentwicklungen des Signalisierungsprotokolls DMSP zu integrieren.

4. Integration von Dienstgüte

Mit dem DSDM und seinem Signalisierungsprotokoll DMSP steht eine geeignete Dienstgüteinfrastruktur auf Basis von Diffserv bereit, es fehlen jedoch angepasste Anwendungen, um sie nutzen zu können. Darüber hinaus können sich bei dem Versuch der Integration von Dienstgütefunktionalität in Anwendungen neue Anforderungen an das Management herauskristallisieren und, wie hier geschehen, zu seiner Weiterentwicklung beitragen helfen.

Idealerweise ist jede Applikation, die von einer qualitätsgesicherten Kommunikation profitieren kann, selbst in der Lage, den für ihre Bedürfnisse passenden Dienst auszuwählen und beim Dienstgütemanagement anzufordern. Solange jedoch das Dienstgütemanagement nicht weitgehend geklärt ist und vor allem keine Dienste standardisiert sind, ist eine Modifizierung von Anwendungen auf breiter Basis kaum sinnvoll. Es wurde deshalb nach einer Möglichkeit gesucht, die Nutzung von Dienstgüte für möglichst zahlreiche Anwendungen zu ermöglichen, ohne diese letztlich modifizieren zu müssen. Die Wahl viel auf das World Wide Web (WWW), da als nicht nur eine allgegenwärtige und vertraute Benutzerschnittstelle darstellt, sondern zudem die interessante Eigenschaft besitzt, über den Webbrowser weitere Anwendungen starten zu können. Obwohl im Rahmen des Forschungsvorhabens nur Webbrowser- und -server um Funktionen zur Initiierung von Dienstgüte erweitert wurden [DTWZ03], kann so ebenfalls die Kommunikation einer über den Browser gestarteten Anwendung, also insbesondere auch eine zur Nutzung von digitalen Bibliotheken benötigte Anwendung, qualitätsgesichert stattfinden.

4.1 Webbasierte Initiierung

Zum Verständnis des grundlegenden Ablaufs sei folgendes Szenario (vgl. Abbildung 4.1) gegeben: Ein Bibliotheksbesucher nutzt die digitale Bibliothek über seinen Webbrowser. Gewisse Dokumente/Anwendungen mögen von einer qualitätsgesicherten Kommunikation mit der Bibliothek profitieren. Browser und Webserver tauschen zu diesem Zweck Informationen über die benötigte Dienstgüte und die Wünsche des Besuchers aus und veranlassen nach der Auswahl eines geeigneten Dienstes beim Dienstgütemanagement dessen Reservierung. Anschließend wird vom Browser die für das Dokument benötigte Anwendung (in der Abbildung die Zielanwendung) gestartet, deren Kommunikation durch die vorhergehende Reservierung

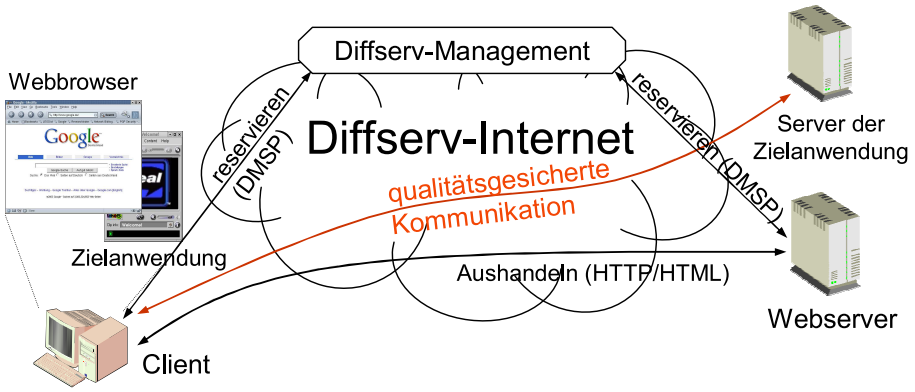


Abbildung 4.1 Initiierung von Dienstgüte über das World Wide Web.

von Diffserv unterstützt mit hoher Qualität stattfinden kann. Nachdem die Verbindung zwischen Zielanwendung und -server durch einen der beiden Kommunikationspartner beendet wurde, wird durch den Browser und/oder den Webserver das Management aufgefordert, die Reservierung wieder abzubauen.

Im Vergleich zum "herkömmlichen" Start einer Anwendung über den Browser sind folgende Punkte zu ergänzen, um eine qualitätsgesicherte Kommunikation initiieren zu können:

- Zwischen Webserver und Browser muss die geeignete Dienstgüte ausgehandelt werden. Der folgende Abschnitt 4.2 stellt die verschiedenen Varianten hierzu vor und beschreibt die entworfenen Ergänzungen von HTML und HTTP.
- Browser und/oder Server müssen in der Lage sein, entsprechend dieser Parameter beim Dienstgütemanagement eine Reservierung durchzuführen. Wie bereits in Abschnitt 3.3 erwähnt, wird hierfür das eigens hierfür entwickelte DMSP eingesetzt.
- Die Zielanwendung muss server- und/oder clientseitig überwacht werden. Zum einen muss das Ende der Kommunikation bestimmt werden können, damit die Reservierung ebenfalls beendet werden kann. Zum anderen sollte die Reservierung möglichst exakt der tatsächlichen Kommunikationsverbindung entsprechen, um eine Nutzung durch Dritte zu verhindern. Auf die Hintergründe hierzu und die gewählte Umsetzung wird in Abschnitt 4.3 näher eingegangen.

4.2 Signalisierung

Folgt ein Benutzer einem in der HTML-Seite eingebetteten Hyperlink zu dem gewünschten Dokument, so erhält der Browser nicht sofort das eigentliche Dokument, sondern eine kleine Datei, die lediglich das eigentliche Dokument referenziert und die hier als "Vorschaltdatei" bezeichnet werden soll. Diese Referenz ist dabei in einer für die jeweilige Zielanwendung spezifischen Syntax verfasst und für den Browser meist unverständlich. Der Webserver gibt im Kopf der HTTP-Nachricht, die die Vorschaltdatei transportiert, deren Mime-Type (Multipurpose Internet Mail Extensions) an, anhand dem der Browser die geeignete Anwendung bestimmen und starten kann, wobei er ihr die lokal gespeicherte Vorschaltdatei übergibt. Die Vorschaltdatei ist nötig, da ansonsten der Browser das gesamte Dokument herunterladen müsste, um es an die Zielanwendung übergeben zu können. Dies ist nicht immer möglich oder vom Anbieter nicht gewünscht. Ein bekanntes Anwendungsbeispiel hierfür ist das Streaming von Audio- oder Videodaten.

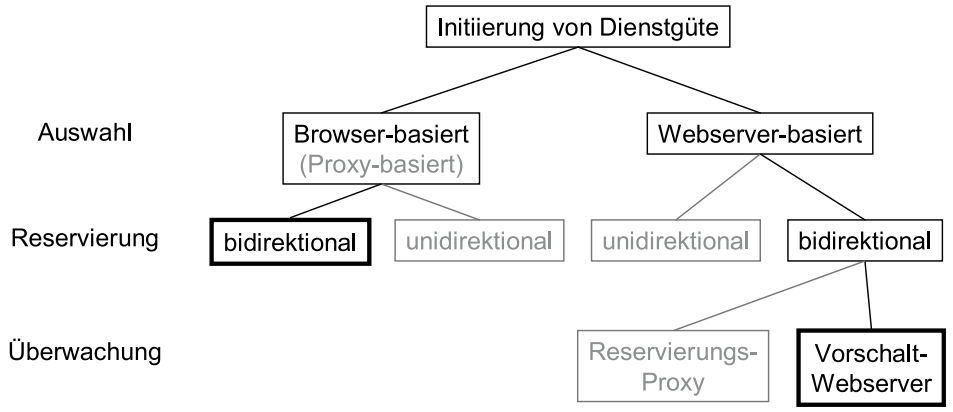


Abbildung 4.2 Mögliche Varianten zur Umsetzung einer webbasierten Dienstgüteinitiierung. Die nicht implementierten Varianten sind grau dargestellt.

Gemäß Abbildung 4.2 lassen sich eine Reihe von Alternativen unterscheiden. Genau wie derzeit die Auswahl einer geeigneten Repräsentation von semantisch äquivalenten Inhalten (verschiedene Sprachen, Grafikformate) durch den Webserver, den Browser oder im Proxy erfolgen kann, was als Server-driven, Agent-driven bzw. Transparent Content Negotiation bezeichnet wird, kann auch die Entscheidung über die zu verwendenden Dienstgüteparameter vom Webserver oder vom Browser getroffen werden. Ein Proxy im Sinne der Transparent Content Negotiation ist hier nicht sinnvoll, da Parameter für eine Kommunikation Ende-zu-Ende ausgehandelt werden sollen und es sich darüber hinaus im allgemeinen um nicht cachebare Inhalte handelt – sei es, aufgrund rechtlicher Aspekte oder wegen eines stringenten Zeitbezugs wie bei interaktiven oder Liveinhalten. Wenn ein Proxy eingesetzt wird, dann ist dieser nicht als Teilnehmer einer Cachehierachie zu sehen, sondern als Stellvertreter des Browsers. Aus Sicht des Netzes sollte er dicht am Browser platziert werden sowie Zugriff auf den Datenpfad der zukünftigen Zielanwendung haben, wie in Abschnitt 4.3 näher erläutert ist. Die Modifikationen, wie sie im Folgenden für den Browser vorgestellt werden, sind dann stattdessen am Proxy vorzunehmen.

Die server- oder browserseitig ausgewählte Dienstgüte muss nun beim Management angefordert werden. Je nach der von der Managementinstanz gebotenen Funktionalität kann die Reservierung gleich bidirektional erfolgen, im Falle der browserbasierten Auswahl sinnvollerweise durch den Browser, im Falle der serverbasierten entsprechend durch den Webserver. Bietet die eigene Mangementinstanz lediglich die unidirektionale Reservierung an, muss jeder, Browser wie Server, für die eigene Senderichtung eine Reservierung anfordern.

Um im ausgewählten Szenario mittels des Webs die benötigte Dienstgüte auszuhandeln, wurden im zugrundeliegenden Protokoll HTTP (für den serverbasierten Ansatz) und in der Seitenbeschreibungssprache HTML (für den browserbasierten Ansatz) einige Erweiterungen vorgenommen, so dass sich Dienstgüteparameter austauschen lassen.

Die wichtigste HTTP-Erweiterung in der serverbasierten Variante ist der HTTP-Header „Client-Address“. Er wird benötigt, um auch dann noch Reservierungen initiieren zu können, wenn die HTTP-Kommunikation über HTTP-Proxys stattfindet. Der Browser teilt hierin dem Webserver die zur Durchführung der Reservierung zwingend erforderliche IP-Adresse der Zielanwendung auf Clientseite mit. Da Zielanwendung und Browser auf demselben Host laufen, initialisiert der Browser Client-Address mit seiner eigenen IP-Adresse oder seinem Hostnamen. Fehlt diese Kopfzeile, beispielsweise wenn ein unmodifizierter Browser verwendet wird, versucht der Webserver, die Clientadresse von der Transportverbindung, über die die HTTP-Anfrage kam, abzuleiten, sofern kein „Via:“-Header vorhanden ist, da in diesem

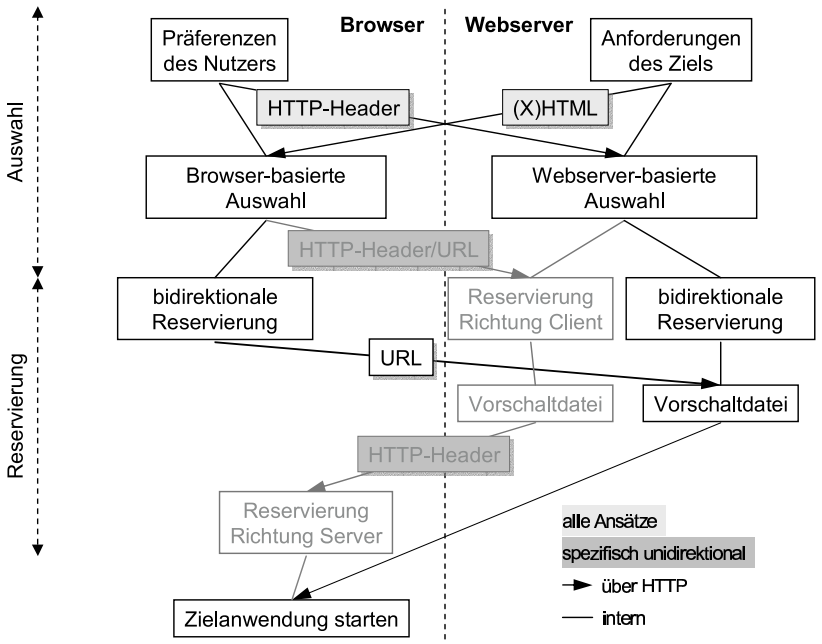


Abbildung 4.3 Zeitlicher Ablauf der vier grundlegenden Varianten zur Dienstgüteinitiierung: browser- bzw. serverbasierte Auswahl der Dienstgüteparameter mit anschließend einer uni- bzw. zwei bidirektionalen Reservierung(en). Grau dargestellt sind die nicht implementierten Varianten mit unidirektionaler Reservierung. Stellen, an denen Dienstgüteparameter ausgestauscht werden, sind hell- bzw. dunkelgrau hinterlegt.

Fall die Anfrage über HTTP-Proxys lief und die so gewonnene IP-Adresse die des letzten Proxys und nicht die der Zielanwendung wäre.

Alternativ kann statt des Browsers auch der erste Proxy nach dem Client modifiziert sein, der dann als Stellvertreter des nicht dienstgütefähigen Browsers die Client-Address-Kopfzeile einfügt (Die Browseradresse ist dem Proxy von der HTTP-Verbindung her bekannt). Zusätzlich muss jedoch der Proxy auch auf dem späteren Datenpfad der Zielanwendung liegen, denn nur so kann er die Kommunikation der Zielanwendung überwachen und deren Ende erkennen.

Neben dem Client-Address-Header sind noch weitere HTTP-Header definiert worden, mit denen der Benutzer bzw. sein Browser Einfluss auf den serverseitigen Auswahlprozess nehmen kann, vergleichbar mit der Funktion von Accept-Headern bei der Content-Negotiation. Die Eigenschaften der Zielanwendung bzgl. der benötigten Dienstgüte verbleibt lokal auf dem Webserver und muss nicht zum Browser übermittelt werden.

Im Gegensatz dazu werden im Falle der browserbasierten Auswahl die Parameter zur Beschreibung der Dienstgüteanforderung mittels HTML-Erweiterungen an den Browser übermittelt, während umgekehrt die Präferenzen des Benutzers lokal im Browser verbleiben können. Die Erweiterung wurde möglichst einfach gehalten und so gestaltet, dass auch nicht modifizierte Browser die veränderten HTML-Quellen problemlos darstellen können. Die Erweiterung besteht aus insgesamt fünf neuen Attributen für das Anchor-Element („a“-Tag):

qosservice: Eine (oder zwei für bidirektionale Reservierungen) Dienstbeschreibun-g(en) in Mime-Type-Syntax mit Haupttyp „qos“ und dem Dienst als Subtyp. Als Parameter sind bisher definiert: Rate(nbereich) (r) (Angabe von Bereichen mittels „-“, mehrere Raten(bereiche) durch „+“ getrennt), min. Verzögerung (c), max. Verzögerung (d) mit Wahrscheinlichkeit ihres Erreichens (p), max. Jitter (j) sowie Multiplikator (m) und

Offset (o), um die Rate einer der beiden Dienstbeschreibungen von der jeweils anderen abhängig machen zu können. Einheiten sind Bit/s bzw. Sekunden, wobei die Parameter zur Beschreibung der gewünschten Verzögerung optional sind, bei der Antwort des Dienstgütemanagements zur Beschreibung des tatsächlich bereitsgestellten Dienstes sind sie jedoch immer enthalten; Ratenbereiche gibt es bei der Antwort nicht. Da der max. Jitter als Differenz von max. und min. Verzögerung definiert ist, dürfen nur jeweils zwei der drei Parameter c, d und j zusammen angegeben werden. Ein Beispiel:

```
qosservice="qos/af;r=4000000-8000000+12000000;c=.24;d=2;p=.97,
            qos/ef;c=.025;o=50000;d=.28;p=.999;j=.04"
```

qoshost: Adresse oder Hostname des Servers der Zielanwendung, falls abweichend vom Host im „href“-Attribut.

qostarget: Eine bereits bestehende Reservierung mit demselben „qostarget“ wie das der neu aufzubauenden wird durch diese neue Reservierung ersetzt anstatt neben der neuen Reservierung bestehen zu bleiben.

qosgroup: Zur Gruppierung mehrerer inhaltlich äquivalenter, sich jedoch bzgl. der benötigten Dienstgüte unterscheidender Links.

qoshref: (Regulärer) Suchausdruck plus Ersetzungsregel – vergleichbar dem Substitutionskommando von „sed“. Sie wird angewendet auf die Dienstbeschreibung (in Mime-Type-Syntax) des tatsächlich vom Dienstgütemanagement reservierten Dienstes und das Ergebnis ersetzt das href-Attribut als Ziel des a-Tags.

Das Attribut „qoshost“ wird benötigt, wenn sich die Vorschaltdatei nicht auf demselben Host befindet wie die Zielanwendung. „qostarget“ wird verwendet, wenn die zu unterstützende Zielanwendung immer nur eine Instanz ihrerselbst laufen lassen kann (typischerweise bei Playern für Audio-/Videostreaming). Mittels „qosgroup“ werden mehrere Links zu einer Gruppe zusammengefasst, die der Browser dann gemeinsam anbieten kann (per Kontextmenü oder Dialogbox). Das „qoshref“ verfolgt das Ziel, dass bei einem mit unterschiedlichen Datenraten kodiertem Dokument nicht mittels Try & Error für jede mögliche Rate ein Reservierungsversuch unternommen wird, bis schließlich einer zum Erfolg führt, sondern dass diese Auswahl im Dienstgütemanagement erfolgt, das versucht, die höchstmögliche Rate zu reservieren. Damit anschließend das zur Reservierung passende Dokument ausgewählt werden kann, lässt sich mittels des „qoshref“-Attributs das Ziel eines Links abhängig von der tatsächlich erfolgten Reservierung machen.

Im Rahmen dieses Forschungsvorhabens wurde anfangs die unidirektionale Reservierung entworfen, da das Management keine bidirektionalen Reservierungen unterstützte. Da jedoch praktisch alle (Unicast-)Kommunikationsbeziehungen in Internet bidirektional sind, ist die Integration in das Management sinnvoll und wurde in diesem Forschungsvorhaben vorgenommen. Entsprechend wurden danach nur die Varianten mit bidirektionaler Reservierung implementiert, die die webbasierte Initiierung ohne Einbußen in der Funktionalität deutlich vereinfachte (vgl. Abbildung 4.3).

Die Lösung, über einen clientseitigen Proxy Dienstgüte zu initiieren, wurde ebenfalls nicht implementiert, da sich die Überwachung noch aufwändiger und unsicherer gestaltet als bei den im folgenden Abschnitt 4.3 erläuterten umgesetzten Varianten. Der Proxy hat keinen Zugriff auf einen der Kommunikationsendpunkte und kann so nur, vergleichbar mit dem Vorgehen bei einer Firewall, anhand der ihn passierenden IP-Pakete auf Verbindungen rückschließen. Während bei TCP Beginn und Ende der Kommunikation durch den Verbindungsauf- und -abbau eindeutig festgestellt werden kann, ist dies bei UDP wegen dessen verbindungslosen

Charakters und insbesondere wegen des damit fehlenden Verbindungsabbaus schwierig. So kann das Ende der Kommunikation einer Zielanwendung nur durch einen Timeout vermutet werden, der nach gewisser Zeit ohne Verkehr den Abbau der Reservierung auslöst.

4.3 Überwachung der Zielanwendung

Am schwierigsten gestaltet sich die Überwachung der Zielanwendung durch Browser bzw. Webserver. Die Überwachung verfolgt drei Ziele:

- Differenzieren verschiedener Anwendungen, die zwischen denselben Endsystemen kommunizieren, um sie gegeneinander abschotten zu können und so Missbrauch von reservierten Ressourcen durch unbefugte Anwendungen auszuschließen.
- Ermöglichen von unterschiedlichen Diensten auch zwischen denselben Endsystemen. Ohne eine Differenzierung ist keine Unterscheidung verschiedener Kommunikationsbeziehungen und damit auch keine gleichzeitige Nutzung unterschiedlicher Dienste möglich.
- Abbau der Reservierung nach dem Ende der entsprechenden Kommunikationsbeziehung.

Während die ersten beiden Punkte optional sind, wenn nicht mehrere Anwendungen parallel zwischen demselben Paar aus Client und Server kommunizieren sollen, macht der letzte Punkt eine Überwachung grundsätzlich notwendig. Zur Klarstellung sei nochmals angemerkt, dass über die beim Aufbau der Reservierung anzugebenden IP-Adressen eine Abgrenzung von Reservierungen verschiedener Client-Server-Paare bereits von vornherein gegeben ist.

Zur Überwachung untersucht der Browser nach dem Starten der Zielanwendung eine gewisse Zeit lang alle geöffneten Sockets (5-Tupel (Quell-IP-Adresse, Quell-Port, Ziel-IP-Adresse, Ziel-Port, Protokoll)) des Systems, ob ein neuer Socket mit passenden IP-Adressen (diese sind ja durch die Reservierung bereits bekannt) geöffnet wird. Wird der Browser fündig, wartet er noch eine weitere kurze Zeit ab, ob weitere Sockets mit demselben Adresspaar auftauchen. Anschließend wird die bestehende Reservierung beim Dienstgütemanagement auf die gefundenen Ports und Protokolle eingeschränkt. Ab hier ist kein Missbrauch durch andere Anwendungen mehr möglich, da Sockets eindeutig sein müssen und damit jeder Socket einmalig ist. Der Browser überwacht anschließend nur noch die gefundenen Sockets, um die Reservierung abzubauen, wenn diese durch die Zielanwendung wieder geschlossen wurden. Für den Fall, dass die Zielanwendung dieselben Ports immer wieder verwendet, kann das Ende der Reservierung nicht (sicher) erkannt werden, so dass eine u. U. unpassende alte Reservierung auch für die neue Kommunikation dieser Anwendung genutzt wird. Mittels „qostarget“ kann hier jedoch dafür gesorgt werden, dass die alte Reservierung durch eine neue ersetzt wird und so immer nur eine Reservierung zu dem gegebenen Server der Zielanwendung besteht. Zudem betrifft dies typischerweise ausschließlich UDP-Ports, da bei TCP wegen des 2·MSL (Maximum Segment Lifetime) andauernden Time_Wait-Zustands eine Wiederverwendung zwar möglich aber clientseitig höchst unüblich ist.

Bei der serverbasierten Variante muss der Webserver Zugriff auf die von der Zielanwendung geöffneten Sockets erlangen. Im allgemeinen kann und soll jedoch nicht angenommen werden, dass sich beide Server auf ein- und denselben Host befinden. Um trotzdem Zugriff zu erlangen, sind zwei Lösungen denkbar: Ein Reservierungsproxy oder ein „Vorschalt“-Webserver. Bei der ersten Lösungsvariante wir die Überwachung der Zielanwendung in einen Proxy ausgelagert, der auf dem Server der Zielanwendung läuft und über den der Webserver alle seine Reservierungsanfragen abwickelt. Diese Variante wurde jedoch nicht implementiert. Alternativ kann

ein „kleiner“ Webserver auf dem Server der Zielanwendung platziert werden, der jedoch nur noch die Vorschaltdateien ausliefert. Die Überwachung der Zielanwendung gestaltet sich ansonsten äquivalent zu der browserbasierten Variante.

Betrachtet man den Ablauf der Überwachung näher, ergeben sich die folgenden inhärenten Nachteile einer generischen Unterstützung, wie sie in diesem Forschungsvorhaben umgesetzt wurde:

- Es können nicht zeitgleich (sondern nur nacheinander) mehrere (Ziel-)Anwendungen gestartet werden, die zwischen denselben zwei IP-Adressen kommunizieren, weil sonst die Reservierungen nicht mehr eindeutig den verschiedenen geöffneten Sockets zugeordnet werden können.
- Alle Sockets einer Anwendung teilen sich die Reservierung. Folglich kann eine Anwendung immer nur einen Dienst nutzen, da kein Wissen über die korrekte Zuordnung der verschiedenen Dienste zu den von der Anwendung geöffneten Sockets besteht.

Damit wird klar, dass die präsentierte Lösung tatsächlich nur als eine Übergangsstrategie gesehen werden kann, die den Weg zur nativen Einbindung von Dienstgüte in jede Applikation bereiten hilft. Den generischen Ansatz jedoch nur deshalb aufzugeben, scheint wenig sinnvoll, da die prinzipielle Funktionsfähigkeit und Brauchbarkeit gegeben ist und damit eine hinreichend flexible Plattform zur Nutzung und zur Fortentwicklung des Dienstgütemanagements bereitsteht.

4.4 Prototypische Implementierung

Die Ergänzung von Dienstgütefunktionalität für den Webbrowser Mozilla ab Version 1.2.1 (bis 1.4b getestet) wurde von Andreas Kunz [Kunz03] vorgenommen, der Webserver Apache in Version 2.0.x (2.0.45 getestet) wurde von Marek Tomczyk modifiziert [Tomc03].

5. Zusammenfassung und Ausblick

Im Rahmen des Forschungsvorhabens DiDelBi wurde eine Dienstgüteunterstützung für das Umfeld digitaler Bibliotheken entworfen und prototypisch implementiert. Dabei konnten wichtige Anforderungen an die bereitzustellenden Dienstklassen einerseits wie an das Dienstgütemanagement andererseits abgeleitet werden. So wurden zwei neue Dienste zur Unterstützung transaktionsbasierten Verkehrs bzw. zur Abkapselung von Hintergrundverkehr entworfen, simulativ untersucht und zur Standardisierung eingereicht. Weiterhin wurden eine Reihe von Erweiterungen in das Diffserv-Management integriert wie die Unterstützung bidirektionaler Reservierungen auch bei asymmetrischen Kommunikationsbeziehungen oder die Trennung von Initiator einer Reservierung als eigenständige Entität unabhängig von Sender und Empfänger. Beim Entwurf des Managements wurde besonderer Wert auf eine skalierbare Lösung gelegt, die nicht nur innerhalb einer Diffserv-Domäne sondern auch internetweit Ende-zu-Ende eine garantierte Dienstgüte bereitstellen kann.

Um diese Infrastruktur für digitale Bibliotheken nutzbar zu machen, wurden die Komponenten des World Wide Web um Funktionen zur Beschreibung, Auswahl und Reservierung von Dienstgüte erweitert mit der Möglichkeit, auch für andere Anwendungen außerhalb des Webs eine qualitätsgesicherte Kommunikation zu initiieren. Die Umsetzung wurde generisch gehalten, um nicht auf bestimmte Anwendungen eingeschränkt zu sein. Dieser Weg wurde eingeschlagen, da durch die noch am Anfang stehenden Standardisierungsbemühungen des Diestgütemanagements auf Basis der Differentiated Services derzeit keine ausreichend feststehenden Schnittstellen bestehen und sich entsprechend eine direkte Integration von Dienstgütefunktionen in die im Umfeld von digitalen Bibliotheken anzutreffenden Anwendungen als nicht sinnvoll darstellt.

Insgesamt präsentiert sich mit den hier entwickelten und prototypisch implementierten Verfahren ein erstes Gesamtkonzept zur Bereitstellung und Nutzung von Dienstgüte für digitale Bibliotheken, dass durch seinen generischen Ansatz auch einer allgemeinen Nutzung im Internet nicht im Wege steht.

6. Literatur

[Bles02] Roland Bless. *Integriertes Management qualitätsbasierter Internetkommunikationsdienste*. Dissertation, Universität Karlsruhe (TH), 2002.

[BlNW02] Roland Bless, Kathleen Nichols und Klaus Wehrle. A Lower Effort Per-Domain Behavior for Differentiated Services. Internet-Draft – `http://doc.tm.uka.de/2002/draft-bless-diffserv-pdb-le-01.txt`, November 2002.

[BZBH+97] Bob Braden, Lixia Zhang, Steve Berson, Shai Herzog und Sugih Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205, September 1997. Proposed Standard.

[DCBB+02] Bruce Davie, Anna Charny, Jon Bennett, Kent Benson, Jean-Yves Le Boudec, Bill Courtney, Shahram Davari, Victor Firoiu und Dimitrios Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). RFC 3246, März 2002. Proposed Standard.

[DTWZ03] Mark Doll, Marek Tomczyk, Klaus Wehrle und Martina Zitterbart. Ansätze für eine Web-basierte Initiierung qualitätsbasierter Kommunikationsdienste. In *KiVS 2003*, Februar 2003.

[HBWW99] Juha Heinanen, Fred Baker, Walter Weiss und John Wroclawski. Assured Forwarding PHB Group. RFC 2597, Juni 1999. Proposed Standard.

[Jans03] Stefan Jansen. Implementierung einer Komponente zur Signalisierung und Verwaltung von Ressourcenanforderungen. Diplomarbeit, Universität Karlsruhe (TH), April 2003.

[Katz00] Jochen Katz. Entwurf und Implementierung einer Managementinstanz für Differentiated-Services-Netze. Diplomarbeit, Universität Karlsruhe (TH), November 2000.

[Küfn03] Tobias Küfner. Entwurf und Evaluierung eines Per-Domain Behaviors zur Unterstützung transaktionsorientierter Kommunikationsbeziehungen. Diplomarbeit, Universität Karlsruhe (TH), 2003.

[Kunz03] Andreas Kunz. Web-basierte Initiierung qualitätsgestützter Kommunikationsdienste – Browser-seitiger Ansatz. Studienarbeit, Universität Karlsruhe (TH), 2003.

[LGGV+00] Cees T. A. M. de Laat, George M. Gross, Leon Gommans, John R. Vollbrecht und David W. Spence. Generic AAA Architecture. RFC 2903, August 2000. Experimental.

[NiCa01] Kathleen Nichols und Brian E. Carpenter. Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification. RFC 3086, April 2001. Informational.

[Paxs97] Vern Paxson. End-to-end routing behavior in the Internet. *IEEE/ACM Transactions on Networking* 5(5), 1997, S. 601–615.

[Tomc03] Marek Tomczyk. Web-basierte Initiierung qualitätsgestützter Kommunikationsdienste – Server-seitiger Ansatz. Studienarbeit, Universität Karlsruhe (TH), 2003.