

Ansätze für eine Web-basierte Initiierung qualitätsbasierter Kommunikationsdienste

Mark Doll, Marek Tomczyk, Klaus Wehrle, Martina Zitterbart

Institut für Telematik, Universität Karlsruhe (TH),
D-76128 Karlsruhe, Deutschland
{doll|tomczyk|wehrle|zit}@tm.uka.de

Zusammenfassung Mit der Differentiated-Services-Architektur werden derzeit die Grundlagen für eine Unterstützung qualitätsbasierter Dienste im Internet gelegt. Jedoch werden die meisten Benutzer mit der Reservierung und Konfiguration dieser Dienste überfordert sein, u. a. mit der Auswahl einer adäquaten Dienstklasse bzw. der passenden Konfigurationsparameter für die jeweilige Kommunikation. Zwar übernehmen Reservierungsprotokolle die Verteilung von Reservierungsinformationen, sie bieten aber keine benutzerfreundliche Auswahl bzw. Aushandlung von Qualitätsparametern an.

In diesem Beitrag werden verschiedene Ansätze vorgestellt, die diese Lücke zwischen Endnutzer und Dienstgütearchitektur schließen. Auf der Basis des World Wide Webs, das sich zunehmend als allgemeine und ubiquitäre Benutzerschnittstelle etabliert, wird eine einfache und benutzerfreundliche Reservierungsmöglichkeit für Anwender angeboten.

1 Strategien zur Auswahl von Dienstgüte

Das heutige Internet wird wesentlich durch das World Wide Web geprägt, das als Standardbenutzerschnittstelle für eine Vielzahl von Diensten und Protokollen weite Verbreitung findet. HTTP als Basis des WWW soll im folgenden als Ausgangspunkt für die Fragestellung dienen, wie sich eine von der Architektur der Differentiated Services (DS) [1] bereitgestellte Dienste auf Anwendungsebene nutzen lassen und benutzerfreundlich konfiguriert werden können.

Es wird vorausgesetzt, dass eine Netzwerkinfrastruktur mit DS-Unterstützung zur Verfügung steht. O.B.d.A. sollen Bandwidth-Broker die Managementebene von DS repräsentieren, denn nur das zu verwendende Reservierungsprotokoll stellt die für die Anwendungsschicht relevante Schnittstelle zur Managementinfrastruktur dar. Im Rahmen der IETF-Arbeitsgruppe NSIS werden hierfür derzeit Anforderungsprofile und Rahmenwerke entwickelt [2].

2 Browser-gestützte Auswahl eines Dienstes

Die Auswahl von Inhalten im WWW beschränkt sich heutzutage im wesentlichen auf zwei Arten und eine Mischform dieser beiden [3]: Bei der Server-seitigen Auswahl liefert ein Server Inhalte in bestimmten Sprachen und/oder Datenformaten nach Auswertung der Accept-Kopfzeilen im HTTP-Request, die sowohl die grundsätzlichen Fähigkeiten des Browsers, als auch die durch den Benutzer am Browser vorgenommenen

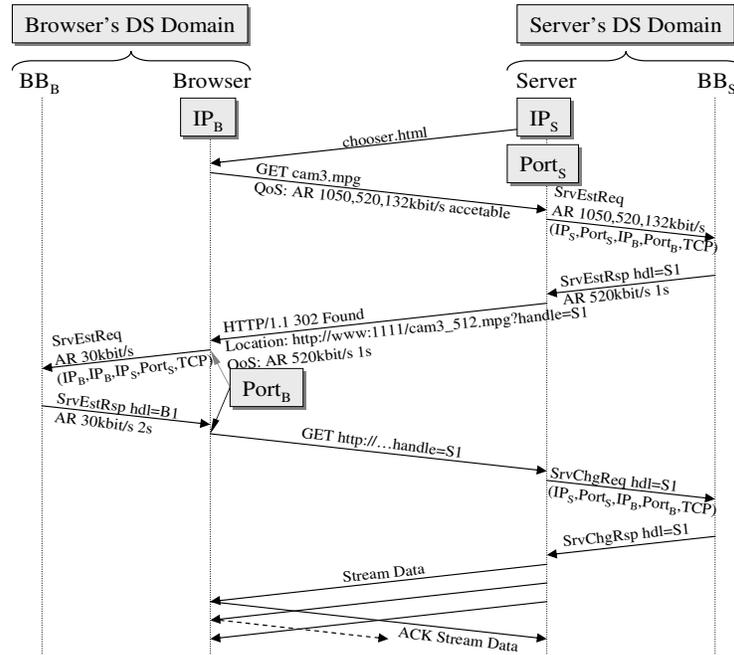


Abbildung 1. Browser mit Dienstgüteunterstützung wählen entsprechend der Voreinstellung und spezieller HTML/XML-Tags einen geeigneten Datenstrom aus.

Einstellungen widerspiegeln. Alles, was hierüber nicht ausgewählt werden kann, bspw. Datenraten, wird üblicherweise durch das Präsentieren einer Webseite mit Alternativen gelöst. Der Benutzer muss daraufhin anhand von menschenverständlichen Beschreibungen aus einer Menge alternativer Links den passendsten auswählen. Bei dieser Browserseitigen Auswahl wäre die Bestimmung der besten Alternative ohne Zutun des Benutzers selbständig durch den Browser möglich, sofern dieser über einen entsprechenden Auswahlalgorithmus verfügt. Die transparente Aushandlung [4] als Mischform der beiden obigen Varianten spielt bei nicht cache-baren Objekten wie bspw. Livestreams keine Rolle und soll nicht weiter betrachtet werden.

Das erste Szenario baut auf dem Browser-basierten Ansatz auf, erweitert ihn jedoch um Komponenten zur Spezifikation bzw. Auswahl der gewünschten Dienstgüte. In einem Fall wird eine modifizierte Browser-Applikation benötigt (Abb. 1); im anderen Fall genügt ein herkömmlicher Browser (Abb. 2).

Der Browser muss beim anspruchsvolleren ersten Ansatz um zusätzliche, für den Benutzer zugängliche, Konfigurationsmöglichkeiten für Dienstgüteparameter erweitert werden. Darüber hinaus sind dienstgütebezogene Daten zu verarbeiten, die der Webserver schickt. Diese Metadaten über verfügbare Datenquellen und Dienstgüteparameter werden automatisch mit den konfigurierten Präferenzen verglichen, woraufhin ein adäquater Datenstrom ausgewählt wird.

Im vorliegenden Szenario möchte ein Benutzer einen vom Webserver angebotenen Stream anschauen. Er erhält von ihm eine Auswahlliste „chooser.html“ mit Hy-

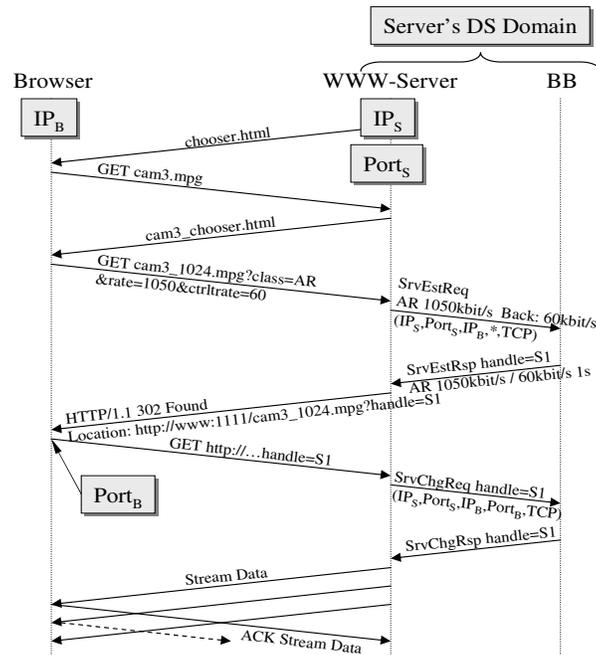


Abbildung 2. Browser ohne Dienstgüteunterstützung burden dem Benutzer die Auswahl der gewünschten Streaming-Qualität auf; Dienstgüteparameter werden URL-codiert.

perlinks zu den verfügbaren Streams. Diese Verweise enthalten, eingebettet in speziellen HTML- bzw. XML-Tags, ausführliche Angaben zu den verfügbaren Datenraten und Dienstgüteparametern. Nur der modifizierte Browser kann die speziellen Tags und Attribute interpretieren und vergleicht diese mit seinen benutzerdefinierten Voreinstellungen. Bei der Auswahl des Links „cam3.mpg“ fügt er zum HTTP-Request URL-codierte Parameter oder spezielle HTTP-Header hinzu, in denen die Anfrage genauer spezifiziert wird (in Abb. 1 z. B. die DS-Dienstklasse „Assured Rate“ und drei Datenraten, deren Reservierung in der angegebenen Reihenfolge angestrebt werden soll). Das HTTP-Extension Framework [5] bietet hier eine geeignete Ausgangsbasis für solche Headererweiterungen. Der gewöhnliche Browser ignoriert diese Tags und sendet seine Anfrage ohne dienstgütespezifische Ergänzungen, so dass keine Inkompatibilitätsprobleme entstehen.

Die Antwort des Webservers hängt vom empfangenen HTTP-Request ab. Im Request des nicht modifizierten Browsers (Abb. 2) fehlen Dienstgüteparameter, so dass der Server noch nicht entscheiden kann, welche Datenrate gewünscht wird. Er generiert eine weitere HTML-Seite, auf der, in für Menschen lesbarer Form, mögliche Alternativen des Films, samt den zugehörigen Links mit URL-codierten Dienstgüteparametern, aufgelistet sind. Nach Auswahl einer Alternative extrahiert der Webserver die Dienstgüteparameter aus der URL und reserviert beim zuständigen Bandwidth-Broker (BB) den gewünschten Dienst mit passender Bandbreite. Der BB bestätigt die gefor-

erten Parameter und garantiert die Einhaltung der Reservierung zwischen Server und Client.

Im Fall des unmodifizierten Browsers muss das Signalisierungsprotokoll zur Ressourcenreservierung sowohl eine Reservierung für die Richtung vom Server zum Client, als auch für die Rückrichtung vornehmen. Die Upstream-Reservierung ist notwendig, damit zum Server zurückgeschickte ACKs und Kontrollpakete ebenfalls qualitätsunterstützt transportiert werden. Der modifizierte Browser ist in der Lage, diese Reservierung selbst vorzunehmen. Eine erfolgreiche Reservierung beantwortet der Server mit der endgültigen URL zum gewünschten Inhalt. Konnte dagegen der BB der Reservierungsanfrage nicht entsprechen, wird eine Fehlerseite generiert und anstelle des Films zum Benutzer gesendet.

Wie auf beiden Abbildungen zu erkennen, werden vom Webserver bzw. vom modifizierten Browser eigentlich zwei Reservierungsanfragen gestellt. Die erste versucht, eine Reservierung einzurichten (*Service Establishment Request*), die bei ungenügenden Ressourcen scheitern kann. Die zweite modifiziert eine bereits bestehende Reservierung (*Service Change Request*), um den Client-Port zu spezifizieren. Da der Client-Port gewöhnlich dynamisch zugeteilt wird, steht er frühestens nach dem „bind“ während des Einrichtens der Streaming-Verbindung zur Verfügung. Die Modifizierungsanfrage ist immer erfolgreich, da die Ressourcen auf allen benötigten Netzabschnitten bereits bei der ersten Anfrage durch den BB reserviert wurden und sich die Route durch die Konkretisierung des Ports nicht ändert.

Bisher wählt der Browser, bzw. der Benutzer, falls es sich um einen nicht modifizierten Browser handelt, anhand der Einstellungen bzw. der Vorlieben einen verfügbaren Datenstrom aus. Ebenso können gemäß der Methode der serverseitigen Auswahl die Präferenzen des Benutzers bei jedem HTTP-Request mittels dienstgütespezifischer Accept-Header mitgeschickt werden, so dass keine in HTML oder XML kodierten Auswahlregeln für die Dienstgüte mehr benötigt werden. Als Nachteil wird der HTTP-Request für alle Anfragen vergrößert.

3 Serviceauswahl mit Proxy-Unterstützung

In den bisher vorgestellten Beispielen wird vorausgesetzt, dass der Anwender zumindest mit einigen technischen Konzepten wie 'Bandbreite' oder 'Reservierung' vertraut ist. Diese Konzepte könnten unerfahrene Anwender überfordern. Überdies werden teilweise modifizierte Browser benötigt.

Ein gangbarer Weg, um den Anwender von komplexen Sachverhalten zu entlasten, liegt im Einsatz von Proxys. Ein Proxy, der für den Benutzer völlig transparent sein kann, kann derart angepasst werden, dass er bestimmte Dienstgüteattribute setzt oder filtert, so dass sie einem vereinbarten Dienstvertrag entsprechen. Er übernimmt auch die Kommunikation mit dem BB.

Das folgende Szenario (Abb. 3) stützt sich auf einen Proxy und verwendet die serverseitige Dienstausswahl, ließe sich aber genauso gut mit der Browser-seitigen Auswahl anwenden. Als Variante seien in diesem Szenario Web- und Streamingserver getrennt. Der Player bleibt weiterhin als Plugin in den Browser integriert. Neben dem zuvor erwähnten Proxy sind wieder zwei BB beteiligt: einer initiiert die Downstream-

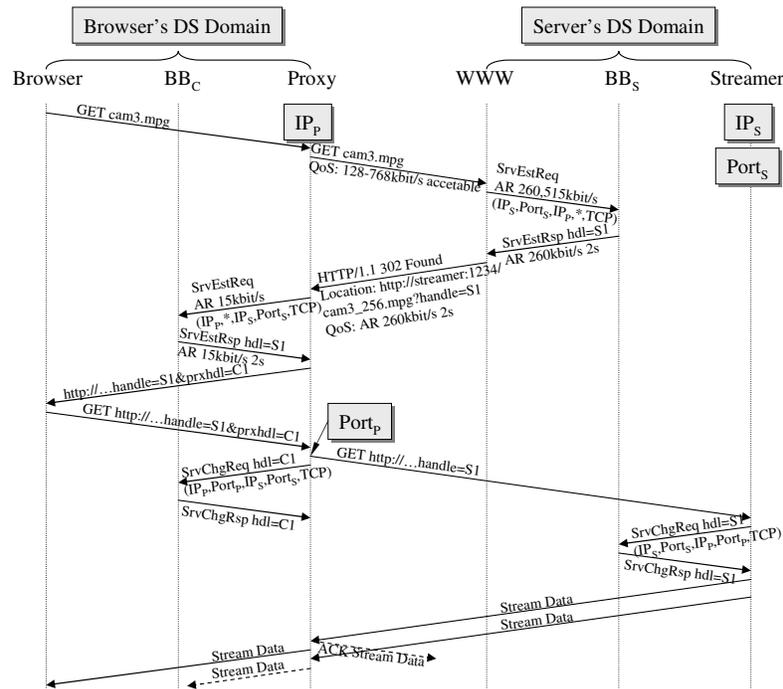


Abbildung 3. Proxy-unterstützte serverseitige Auswahl.

Reservierung, der andere jene für die ACKs in Gegenrichtung. Die BB der Transitdomänen nehmen nur indirekt über die dargestellten BB der Quell- und Zieldomänen an der Kommunikation teil und sind deshalb nicht dargestellt.

Beim Anfordern des Streams „cam3.mpg“ wird die Anfrage vom Proxy-Server abgefangen und dort um zusätzliche Accept-Header zur Angabe von Bandbreiten ergänzt. Die modifizierte Anfrage veranlasst den Webserver, verfügbare Varianten, entsprechend der Bandbreitenvorgaben im Request, auszuwählen. Alle für eine Reservierung in Frage kommenden Datenraten werden um einen Prozentsatz erhöht, um die evtl. während des Streaming-Vorgangs auftretenden Sendewiederholungen durch TCP kompensieren zu können („*Overprovisioning*“). Die endgültige Reservierungsanfrage an den BB beinhaltet neben möglichen Datenraten und gewünschtem DS-Dienst auch den Socket ($IP_S, Port_S, IP_P, *, TCP$), wobei der Port des Proxy-Servers für den späteren Empfang des Streams noch nicht bekannt ist und zunächst unspezifiziert bleiben muss.

Der BB kann die höhere Datenrate nicht garantieren, bestätigt aber eine Reservierung über 260 kbit/s. Die Bestätigung beinhaltet eine Reservierungskennung „S1“, die zum späteren Zeitpunkt für eine Reservierungsänderung oder den Abbruch der Reservierung genutzt werden kann. Zusätzlich wird ein Hinweis zur Optimierung des Puffermanagements mitgeschickt. In unserem Beispiel steht die angeforderte Rate als gemittelter Wert über zwei Sekunden zur Verfügung, so dass die Anwendung, die später die Daten entgegennehmen soll, in der Lage sein sollte, durch einen Puffer entsprechender Größe die Ratschwankungen auszugleichen. Diese Informationen werden,

zusammen mit der zu den eigentlichen Daten zeigenden URL, vom Webserver zum Proxy geschickt.

Eine ähnliche Reservierung wie jene vom Webserver muss durch den Proxy auch für die Rückrichtung vorgenommen werden, damit der reibungslose Transport der ACKs im Upstream sichergestellt ist. Nach einer erfolgreichen Reservierung für den Upstream kann die vom Server empfangene URL mitsamt der Reservierungskennung „C1“ des Proxys an den Browser weitergereicht werden, der daraufhin wieder über den Proxy eine Verbindung zum Streaming-Server aufbaut. Der Proxy bekommt mit der URL seine Reservierungskennung und vervollständigt die Reservierung für den Upstream, nachdem er den Socket für die Verbindung zum Streaming-Server geöffnet hat, da nun der bis dahin unbekannt Port zur Verfügung steht. Der Streaming-Server erhält den Request von diesem Port des Proxys und vervollständigt seinerseits die Downstream-Reservierung.

Hiermit ist zwar die Kommunikation zwischen Proxy und Server durch die DS-Infrastruktur abgesichert, nicht jedoch der Abschnitt vom Proxy zum Benutzer. Hier wäre entweder durch Overprovisioning oder durch eine weitere vom Proxy initiierte Reservierung möglichen Paketverlusten vorzubeugen.

Im Fehlerfall, beispielsweise beim Fehlschlagen der Upstream-Reservierung oder beim Absturz des Browsers, müssen durch geeignete Maßnahmen ungenutzte Reservierungen nach einer gewissen Zeit abgebaut werden. Insofern lässt sich nicht vermeiden, dass nun Proxy und Webserver, wenn auch nur in begrenztem Umfang, Zustände verbindungsübergreifend speichern müssen.

4 Zusammenfassung und Vergleich

Es wurden unterschiedliche Vorgehensweisen für die Anforderung von DS-basierter Dienstgüteunterstützung im World Wide Web betrachtet. Die typische Struktur einer Client-Server-Interaktion im Web macht die auf der Benutzerseite empfundene Dienstqualität hauptsächlich von der Dienstgüte der Serverantwort abhängig. Aus diesem Grund muss der Anwender mit Eingriffsmechanismen für die Beeinflussung der Dienstgüte des Downstreams instrumentalisiert werden. Ein herkömmlicher Browser kann zusammen mit einem dienstgüteunterstützenden Webserver mittels URL-codierter Parameter eine bereitstehende DS-Infrastruktur nutzen, wenn auch mit Einbußen im Komfort durch die zwischengeschalteten Auswahldialoge. Ein modifizierter Browser hat den Nachteil der notwendigen Anpassung, stellt jedoch die elegantere Lösung bei gleichzeitig gesteigertem Komfort dar. Eine Proxy-basierte Lösung vermeidet eine Browser-Modifikation und macht die Dienstgüteaushandlung vollständig transparent, so dass auch unbedarfte Anwender eine optimierte Dienstgüteunterstützung erhalten können.

Literatur

1. S. Blake et al.: An Architecture for Differentiated Services. RFC 2475, IETF, 1998
2. IETF-Arbeitsgruppe NSIS, <http://www.ietf.org/html.charters/nsis-charter.html>
3. R. Fielding et al.: Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, IETF, 1999
4. K. Holtman, A. Mutz: Transparent Content Negotiation. RFC 2295, IETF, 1998
5. H. Nielsen, et al.: An HTTP Extension Framework. RFC 2774, IETF, 2000