

Cooperative QoS Management

Stefan Fischer

**International University, School of Information Technology
D-76646 Bruchsal**

Hermann de Meer

**Columbia University, Center for Telecommunications Research
New York, NY 10027**

Zusammenfassung

Die Übertragung und Darstellung kontinuierlicher Datenströme wie Audio und Video in verteilten Multimediasystemen verlangt ein effizientes Management der verwendeten Ressourcen, wenn eine zeitlich korrekte Darstellung der Datenströme beim Empfänger erforderlich ist. Zu diesem Zweck wurde eine Reihe von *QoS-Managementsystemen* entwickelt. In diesem Artikel wird zunächst aufgezeigt, daß eine grundlegende Eigenschaft dieser Systeme, nämlich die Ende-zu-Ende-Verhandlung der QoS-Parameter, dazu führt, daß sie eher ungeeignet für eine immer wichtiger werdende Klasse von multimedialen Anwendungen sind: den Massenanwendungen, bei denen eine große Zahl von Empfängern gleichzeitig teilnimmt. Als Lösungsmöglichkeit wird das neue *Cooperative QoS Management* beschrieben, das auf der Verwendung von Agenten im Netz als Stellvertreter der Informationsanbieter basiert.

1 Einführung

Beim Design verteilter multimedialer Anwendungen ist die Frage der Dienstgüte (*Quality of Service, QoS*) ein zentrales Thema. Dies liegt vor allem an den relativ hohen Anforderungen, die zeitabhängige digitale Medien wie Video und Audio an die Übertragungs- und Verarbeitungsleistung von Netzwerken und Endsystemen stellen. Müssen sich die multimedialen Anwendungen die Systemressourcen, die für die Erzielung einer bestimmten Dienstgüte verantwortlich sind, miteinander teilen, so ist der auch im Internet verwendete *Best-Effort-Ansatz* besonders bei hoher Auslastung nicht mehr sinnvoll anwendbar; vielmehr wird eine effiziente Verwaltung und Reservierung dieser Ressourcen unabdingbar für eine reibungslose Funktionsweise des gesamten Systems. Sämtliche Maßnahmen, die mit der Ressourcenverwaltung zusammenhängen, kann man unter dem Begriff *QoS Management* zusammenfassen. Dazu gehören neben der eigentlichen Reservierung von Ressourcen u.a.:

- *QoS Negotiation bzw. Dienstgüteverhandlung*: Benutzer handeln untereinander und mit dem Kommunikationsdienstbringer aus, welche Qualitätsmerkmale die Übertragung

eines bestimmten Datenstroms aufweisen soll. Das Verhandlungsergebnis hängt wesentlich von den verfügbaren Ressourcen in den Netzen und Endsystemen sowie dem zu zahlenden Preis ab. Dienstgüteverhandlungen sind notwendig, wenn den Benutzern ein deterministischer oder statistischer Dienst zur Verfügung gestellt werden soll, auf dessen Qualität sie sich voll bzw. entsprechend den statistischen Parametern verlassen können. Ohne beiderseitiges Einverständnis über Qualität und zugehörige Kosten besteht von keiner Seite eine Verpflichtung zu einer Leistung.

- *QoS Monitoring bzw. Dienstgüteüberwachung*: Sowohl Benutzer (oder Anwendung) wie auch der Kommunikationsdienstbringer werden Mittel einsetzen, um die aktuell erbrachte Dienstgüte zu überwachen, also zu messen und mit den ausgehandelten Werten zu vergleichen. Beide können dann entsprechende Maßnahmen einleiten, wenn es zu Abweichungen kommt.
- *QoS Adaptation bzw. Dienstgüteanpassung*: Aufgrund plötzlicher Problemsituationen im Netz kann es passieren, daß Dienstbringer auf der Basis ihrer augenblicklichen Konfiguration die dem Benutzer garantierte Qualität nicht mehr erbringen können. Ziel der Dienstgüteanpassung ist es, so auf die Umweltveränderung zu reagieren, daß die aktuelle Qualität gehalten werden kann. Mögliche Maßnahmen können beispielsweise sein:
 1. Änderung des Pfades, über den die Daten geleitet werden und damit Ausnutzung freier Ressourcen in anderen Netzbereichen.
 2. Ausnutzung möglicher Toleranzgrenzen bei der Dienstgüte anderer Ströme und damit Verschiebung der Ressourcen von einem Strom zu einem anderen.
 3. Verschärfung von QoS-Anforderungen für andere Komponenten auf dem existierenden Pfad, was eine Verringerung der Anforderungen an die den Fehler verursachende Komponente erlaubt.

Daneben könnte auch die Anwendung selbst eine Anpassung durchführen. Stellt sie z.B. fest, daß sich über die aktuelle Verbindung kein Video übertragen läßt, reagiert sie selbständig und stellt nur ein Standbild dar. Die frei werdenden Ressourcen können dann besser für wichtigere Medien wie etwa einen Audiostrom verwendet werden.

- *QoS Renegotiation bzw. Dienstgütereuverhandlung*: Kann der Dienstbringer trotz Adaptation die garantierte Qualität nicht mehr erbringen, so wird eine Neuverhandlung mit dem Benutzer notwendig. Entsprechend kann auch ein Benutzer eine Neuverhandlung initiieren, wenn er beispielsweise mit der aktuell ausgewählten Qualität nicht mehr zufrieden ist oder die Kosten für die Verbindung reduzieren will.

Bereits Ende der 80er und Anfang der 90er Jahre wurden im Umfeld der neu aufkommenden Hochgeschwindigkeitsnetze erste Ansätze zum QoS-Management auf der Transportebene entwickelt [11]. Erst in jüngerer Zeit wurden die Betrachtungen dann erweitert auf mehrere Schichten des Kommunikationssystems bzw. schließlich eine Ende-zu-Ende-Dienstgüte: nur wenn die Übertragungs- und Darstellungsqualität vom Erzeuger der Datenströme bis zu den Sinnesorganen des menschlichen Empfängers aufrecht erhalten werden kann, wird eine aus Benutzersicht akzeptable Dienstgüte erreicht. Unter dieser Prämisse wurden eine Reihe von QoS-Managementsystemen entwickelt. Prominenteste Vertreter sind das QoS-Modell HeiRAT des IBM ENCs in Heidelberg [16], die Quality-of-Service-Architecture (QoS-A) der Universität Lancaster [2] und die Tenet-Architektur des ICSI in Berkeley [4]. Eine aktuelle Zusammenfassung relevanter Ansätze findet sich in [1].

Diese Systeme sind auf Multimediaanwendungen ausgerichtet, die auf Eins-zu-eins-Kommunikationsbeziehungen beruhen, bei denen also typischerweise zwei menschliche Benutzer oder ein menschlicher Benutzer mit einem multimedialen Dokumentenspeicher kommunizieren. So wird z.B. in [10] ein News-on-Demand-System beschrieben, bei dem sich Benutzer einloggen und die neuesten Nachrichten als multimediale Dokumente ansehen können. Man spricht hier von einer *Präsentationsanwendung* (die Daten werden von einem Speichermedium zum Ziel übertragen) im Gegensatz zu einer *Live-Anwendung*, bei der die multimedialen Datenströme sofort nach ihrer Erzeugung zum Ziel übertragen werden. Um die Dienstqualität Ende-zu-Ende garantieren zu können, müssen in diesen Systemen auch Ende-zu-Ende-Verhandlungen zwischen den Benutzern durchgeführt werden.

Betrachtet man nun die Klasse der *multimedialen Massenanwendungen*, also Anwendungen mit m Sendern und n Empfängern, wobei zumindest n typischerweise sehr groß wird, dann ist festzustellen, daß der Ende-zu-Ende-Ansatz nicht mehr effizient einsetzbar ist: Solche Anwendungen zeichnen sich dadurch aus, daß alle Empfänger gleichzeitig einen (oder mehrere) von der Quelle ausgesandten Datenstrom empfangen. Ein Beispiel für eine solche Anwendung wäre die Live-Übertragung eines Fußballspiels. Da nicht alle Empfänger die gleiche Qualität verarbeiten können, muß der Sender verschiedene Qualitätsstufen anbieten¹. Es ist nun leicht vorstellbar, daß eine massenhafte Ende-zu-Ende-Verhandlung der Dienstgüte zwischen dem Sender und allen Empfängern, die ja wahrscheinlich fast gleichzeitig stattfinden wird, sehr schnell zu einer Überlastung des Senders als auch der Leitungen in dessen Umgebung führt.

Aufgrund dieser offensichtlichen Schwierigkeiten wurde ein neuer Ansatz zum QoS-Management entwickelt, dessen Basisidee darin besteht, die QoS-Verhandlungen soweit wie möglich zu dezentralisieren und auf lokaler oder regionaler Ebene durchzuführen. Dazu wird eine verteilte Managementarchitektur eingeführt. Da die verteilten Komponenten des Systems zwar autonom agieren, aber miteinander kooperieren, wurde dieses Verfahren *Cooperative QoS Management* genannt. Dieser Artikel stellt die Grundzüge des neuen Verfahrens vor.

Abschnitt 2 beschreibt zunächst die generelle Architektur des neuen Verfahrens. In Abschnitt 3 werden anschließend die unterstützten QoS-Managementfunktionen detailliert dargestellt. Abschnitt 4 geht auf Strategieaspekte ein, also auf die Frage, wie und wann bestimmte Aktionen bzw. Managementfunktionen ausgeführt werden. Abschnitt 5 diskutiert Implementierungsfragen, und Abschnitt 6 gibt einen kurzen Ausblick auf zukünftige Arbeiten.

2 Architektur des Cooperative QoS-Management

Cooperative QoS-Management basiert auf der Idee, die Kommunikation zur Dienstaushandlung zwischen Sender und Empfängern zu reduzieren und sie durch lokale oder regionale Kommunikation zu ersetzen. Als Ergebnis wird die Überlastung des Senders bzw. der Umgebung des Senders vermieden. Dazu muß es in der Umgebung der Empfänger jeweils

¹. Dies kann z.B. durch gleichzeitiges Senden mehrerer Datenströme verschiedener Qualität oder durch hierarchische Kodierung [14] eines einzigen Datenstroms erfolgen.

einen *Stellvertreter* des Senders geben, der die Verhandlungen durchführt bzw. andere QoS-Managementfunktionen initiiert. Diese Instanz wird im folgenden *QoS-Agent* genannt, da sie autonom agiert. Ein solcher QoS-Agent wird auf jedem Netzknoten und allen beteiligten Endsystem installiert. Sobald sich der erste Empfänger bei einem Agenten anmeldet, um Daten von einem Sender zu empfangen, holt sich dieser Agent die notwendigen Informationen über verfügbare Datenströme und auswählbare Qualitäten von seinem Nachbaragenten in Richtung Quelle. Dies kann unter Umständen auch schon der Agent auf dem Senderrechner sein. Meldet sich ein weiterer Empfänger bei einem bereits aktiven Agenten an, dann ist die Abfrage bzgl. zur Verfügung stehender Ströme nicht mehr nötig; vielmehr kann sofort mit der eigentlichen Verhandlung begonnen werden.

Das Verfahren nutzt in erheblichem Maße die Eigenschaften der *Multicast-Übertragungstechnik* aus. Im Gegensatz zu typischen Präsentationsanwendungen, bei denen ja eine 1:1-Kommunikationsbeziehung zwischen den Partnern besteht, kann bei Massenanwendungen die Eigenschaft ausgenutzt werden, daß viele Benutzer denselben Datenstrom empfangen. Wird dieser Strom per Multicast übertragen, dann bedeutet dies, daß die Daten vom Sender aus nur einmal verschickt werden; bei einer Unicast-Übertragung müßte der Sender die Daten bei n Empfängern auch n -mal versenden. Bei Multicast wird der Datenstrom nur dupliziert, wenn dies notwendig ist, also an einem Netzknoten, über den mehrere Empfänger nur über unterschiedliche Leitungen erreicht werden können. Wird der Datenstrom in m Qualitäten angeboten, dann müssen (je nach Kodierung) maximal m Multicast-Verbindungen aufgebaut werden. Ein wesentliches Vorgehen von Cooperative QoS Management ist es nun, den Wert m möglichst klein zu halten, also möglichst wenige Qualitäten übertragen zu müssen. Dadurch können Ressourcen eingespart werden, die dann anderen Anwendungen zur Nutzung zur Verfügung stehen.

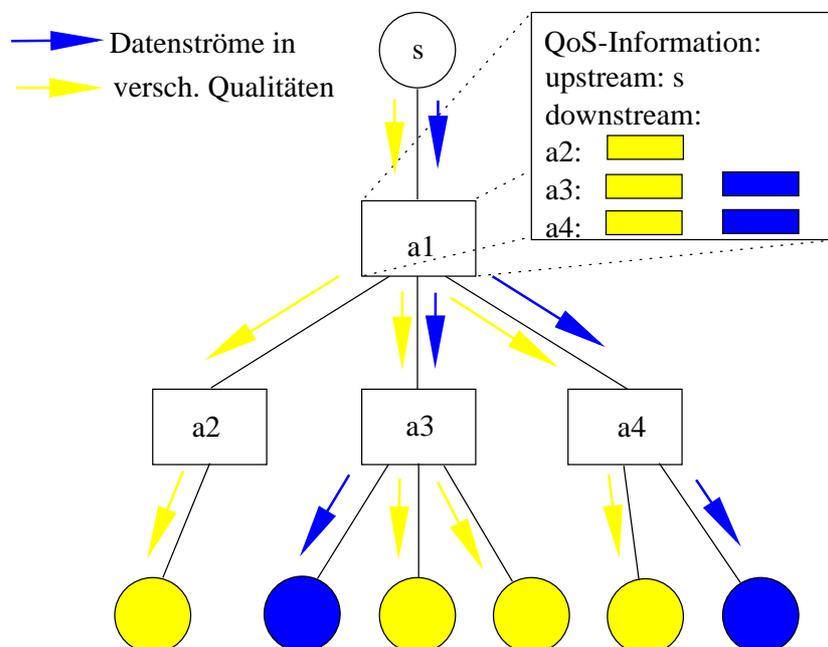


ABBILDUNG 1. Agentenarchitektur im Cooperative QoS Management

Kommunikation in Cooperative QoS Management findet immer nur zwischen benachbarten Agenten statt, so daß es normalerweise zu keiner überregionalen Verbreitung von QoS-

Nachrichten kommt. Der Begriff "benachbart" bezieht sich hier auf die Position der Agenten im vom Sender ausgehenden Multicast-Baum². Agenten erhalten Informationen über die aktuell verfügbaren bzw. übertragenen Qualitäten von ihren im Baum näher an der Wurzel befindlichen Nachbarn. Sie verhandeln mit den Blattknoten, die durch die Endsystem-Agenten gebildet werden. Abbildung 1 zeigt diese Architektur. Jeder Agent hält fest, an welchen "downstream" liegenden Netzknoten er welche Ströme (in einer bestimmten Qualität) weitergibt und welches sein "Upstream"-Agent ist. Das Bild stellt diese Informationen beispielhaft für den Agenten a1 dar.

Die Kommunikation zwischen Agenten erlaubt aber nicht nur die Durchführung lokaler Verhandlungen. Auch die anderen QoS-Funktionen wie Dienstgüteeanpassung und -neuverhandlung werden unterstützt. Der folgende Abschnitt beschreibt detailliert die einzelnen Management-Funktionen.

3 QoS-Management-Funktionen beim Cooperative QoS Management

Mit Hilfe dieser auf Agenten basierenden Architektur können nun die oben beschriebenen QoS-Managementfunktionen realisiert werden. Dieser Abschnitt geht auf die Punkte Dienstgüteeanpassung, -anpassung und -neuverhandlung ein.

3.1 Dienstgüteeanpassungen

Dienstgüteeanpassungen müssen aufgenommen werden, wenn ein neuer Empfänger zu einer laufenden Anwendung hinzukommt, also in den bestehenden Multicast-Baum eingefügt werden soll. Dieser Multicast-Join-Prozeß läuft wie folgt ab:

1. Die Information, welche Anwendungen zur Zeit aktiv sind, ist für potentielle Empfänger über einen Verzeichnisdienst verfügbar. Sobald der neue Empfänger seinen Teilnahmewunsch an einer Anwendung mittels der entsprechenden Sitzungsadresse an den QoS-Agenten weitergegeben hat, kontaktiert dieser den nächstgelegenen QoS-Agenten in Richtung der Quelle. Zur Suche nach diesem Upstream-Agenten wird der Multicast-Join-Algorithmus eingesetzt, der bestimmt, an welcher Stelle des existierenden Baumes der neue Knoten angehängt wird. Ist der Kontakt hergestellt, sendet der Upstream-Agent sämtliche Informationen bzgl. der verfügbaren Datenströme einschließlich der mit der Übertragung verbundenen Kosten an den lokalen Endsystemagenten. Diese Aktion kann auch dazu führen, daß weitere QoS-Agenten auf denjenigen Routern installiert werden, die auf dem Pfad zwischen lokalem und Upstream-Agenten liegen.
2. Der lokale Endsystem-Agent stellt eine Liste aller verfügbaren Datenströme der Sitzung zusammen, wobei er aus der von dem Upstream-Agenten erhaltenen Liste diejenigen ausfiltert, die aufgrund technischer Einschränkungen nicht unterstützt werden können. Die Liste wird an den Benutzer übergeben, der dann eine Reihe von Strömen auswählt

². Ein Multicast-Baum besteht aus dem Senderknoten als Wurzel sowie allen Links und Knoten, über die die zur Multicast-Verbindung gehörenden Daten verschickt werden.

und die gewünschte Qualität selektiert. Außerdem kann er angeben, welche weiteren Qualitäten akzeptabel sind. Diese Information wird vom Managementsystem für die Dienstgütee Anpassung und -neuverhandlung genutzt.

- Über den Verzeichnisdienst und die Information durch seine Nachbaragenten besitzt der lokale Agent die Information darüber, wie die Datenströme empfangen werden können. Dieses Wissen umfaßt beispielsweise notwendige Multicast-Adressen oder Filter, die in einem Ressourcenreservierungsprotokoll gesetzt werden müssen. Er übergibt diese Informationen an die für die Übertragung zuständigen Protokolle (z.B. Ressourcenreservierungs- und Gruppenmanagementprotokolle). Daraufhin wird der Join-Algorithmus erneut durchgeführt, um den Empfänger endgültig in den Multicast-Baum einzufügen. Entlang des neuen Pfades müssen entsprechende Ressourcenreservierungen vorgenommen werden; auf dem schon existierenden Pfad müssen Reservierungen ergänzt werden, wenn die gewünschte Qualität dort bisher noch nicht unterstützt wurde. Sollte eine Reservierung nicht möglich sein, dann muß entweder ein neuer Pfad gesucht oder eine neue Verhandlungsrunde initiiert werden.

3.2 Dienstgütee Anpassung

Dienstgütee Anpassung tritt auf, wenn das System die ausgehandelte Qualität nicht mehr unterstützen kann, aber auch, wenn Empfänger die Gruppe verlassen oder neue Empfänger hinzukommen. Die beiden letzten Fälle bieten die Möglichkeit einer Optimierung der Ressourcennutzung, indem z.B. der Multicast-Baum umkonfiguriert wird. Im Kontext dieses Artikels ist besonders der erste Fall von Interesse.

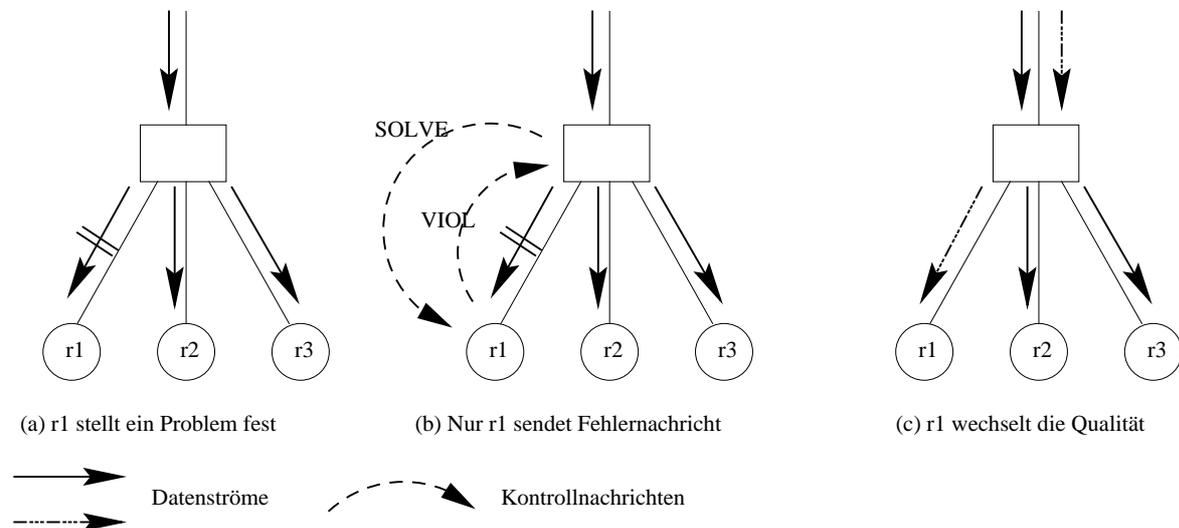


ABBILDUNG 2. Entdeckung eines QoS-Problems nahe beim Empfänger

Da Empfänger üblicherweise ständig die Qualität der eintreffenden Datenströme kontrollieren (mittels der Funktion *QoS Monitoring*), können sie Qualitätsverletzungen schnell feststellen. In diesem Fall sendet der lokale QoS-Agent eine Fehlernachricht (VIOL-Nachricht) an seinen benachbarten Upstream-Agenten. Letzterer wartet nun eine bestimmte Zeit ab, ob er von anderen Empfängern ebenfalls solche Nachrichten erhält. Ist dies nicht der Fall, befindet sich die Quelle des Problems mit hoher Wahrscheinlichkeit beim Empfänger bzw. bei der Verbindung zwischen Empfänger und Upstream-Agent. Daher antwortet der Upstream-Agent

mit einer Nachricht mit der Aufforderung an den Downstream-Agenten, sich selbst um dieses Problem zu kümmern (SOLVE-Nachricht). Eine mögliche Lösung bestünde dann etwa darin, daß der Empfänger eine niedrigere Qualität auswählt, da der Link die ursprüngliche Auswahl nicht unterstützen kann. Dieser Fall ist in Abbildung 2 dargestellt.

In der dargestellten Situation stellt Empfänger *r1* eine Verletzung der ausgehandelten Qualität fest (Teilabbildung (a)) und sendet eine Fehlernachricht an seinen Upstream-Agenten (b). Da dieser keine weitere Fehlernachricht empfängt, antwortet er nach einer bestimmten Zeit mit einer SOLVE-Nachricht (b). Als Ergebnis schaltet *r1* nach einer Neuverhandlung auf eine andere Qualität um (c).

Empfängt dagegen der Upstream-Agent mehrere Fehlernachrichten, dann sendet er selbst eine Fehlernachricht an seinen eigenen Upstream-Agenten, der nun seinerseits auf weitere Fehlernachrichten wartet. Durch dieses Verfahren läßt sich von der Seite der QoS-Agenten her die Fehlerquelle innerhalb des Multicast-Baumes eingrenzen. Sobald die entsprechende Stelle gefunden ist, kann z.B. eine teilweise Rekonfiguration des Multicast-Baumes durchgeführt werden, d.h., der QoS-Agent an der betroffenen Stellen fordert die Routing-Komponente auf, einen neuen Weg zu suchen, der die gewünschte Qualität unterstützt und den aktuell gestörten Link vermeidet. Diese Lösungsmöglichkeit ist in Abbildung 3 dargestellt.

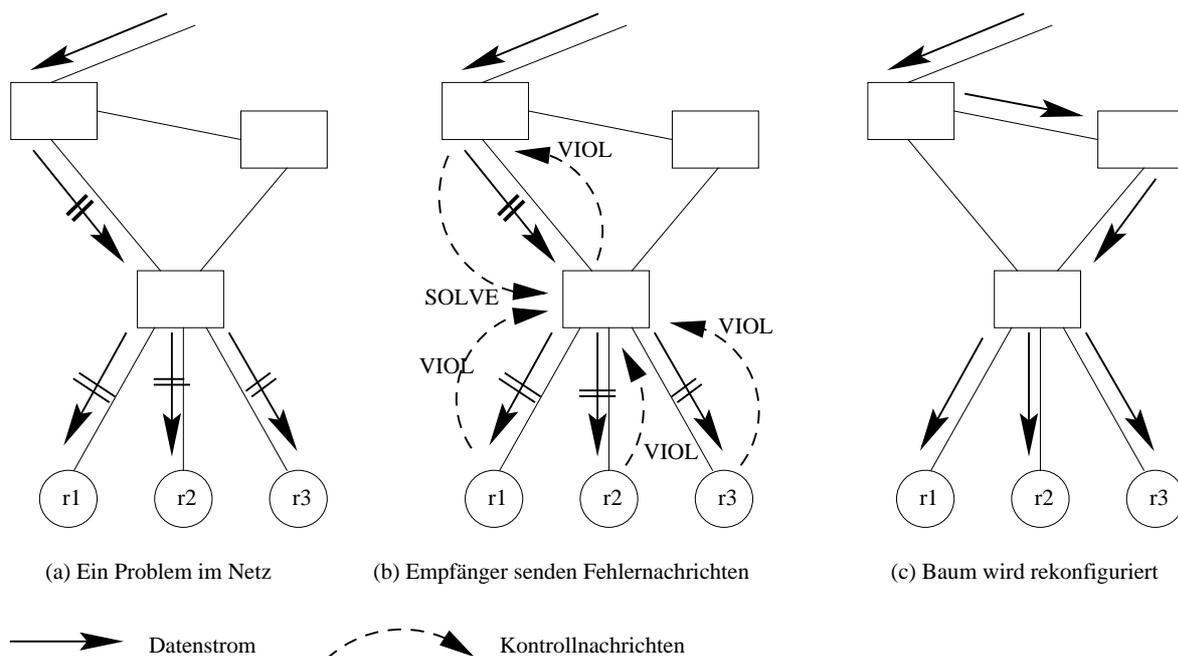


ABBILDUNG 3. Eine Qualitätsverletzung im Netz

In Teilabbildung (a) tritt ein Fehler irgendwo im Netz auf, der dazu führt, daß sämtliche Empfänger Qualitätseinbußen erleiden. Alle drei Empfänger im Beispiel senden daraufhin Fehlermeldungen an ihren Upstream-Agenten (b), der aufgrund der Menge der eintreffenden Meldungen nun ebenfalls eine Fehlernachricht generiert und an seinen eigenen Upstream-Agenten schickt (b). Da dieser keine weiteren Meldungen erhält, schickt er eine SOLVE-Nachricht zurück (b). Zur Lösung des Problems sucht nun der (idealerweise) QoS-basierte

Multicast-Routing-Algorithmus nach einem neuen Weg, über den dann die Daten übertragen werden (c). Ein Problem mit dieser Vorgehensweise ergibt sich bei degenerierten Multicast-Bäumen mit nur einem abwärts gerichteten Zweig bei einem Agenten, da dann bereits eine einzige Fehlermeldung das Weiterleiten der Fehlermeldung nach oben auslöst. Dadurch wird unter Umständen die Fehlerquelle übersehen. Aus diesem Grund führen die einzelnen Stationen selbständig ein Monitoring durch. Details zu dieser Lösung finden sich in [7].

3.3 Dienstgüteneuverhandlung

Insbesondere bei der Neuverhandlung ergibt sich eine interessante neue Möglichkeit zur Optimierung des Ressourcenverbrauchs im Multicast-Baum: Bei jeweils individuell durchgeführten Verhandlungen zwischen Empfänger und Agent kann es schnell zu einer unerwünschten Ressourcenverteilung kommen, da sich Empfänger nicht daran orientieren können, wie sich andere Empfänger in ihrer Umgebung verhalten. Ein Extrembeispiel: der Sender bietet einen Video in drei Qualitäten an, und drei benachbarte Empfänger wählen jeweils eine dieser Qualitäten aus. Dann müssen auf dem gesamten Weg vom Sender zu den Empfängern Ressourcen für alle drei Ströme reserviert werden, so daß der Vorteil der Multicast-Übertragung völlig wegfällt.

Die lokalen Agenten wissen jedoch, welche ihrer "Kunden" welche Datenströme in welcher Qualität empfangen. Damit sind sie in der Lage, *Vorschläge für einen Qualitätswechsel* zu generieren und an einzelne Empfänger zu senden. Sie informieren damit den Empfänger über die aktuell günstigerweise zu verwendenden Dienstgütern für die Ströme, die er empfängt. Der Empfänger kann dann entscheiden, ob er den Vorschlag akzeptiert und die Qualität wechselt. Tut er dies, und ist er der einzige oder letzte Empfänger einer bestimmten Qualität in einem lokalen Bereich, dann können anschließend die für den vorher empfangenen Strom reservierten Ressourcen freigegeben werden, da der Strom nicht mehr in diese Region übertragen werden muß. Üblicherweise wird mit einem Akzeptieren ein monetärer Vorteil für den Empfänger verbunden sein, denn sonst bestünde für ihn wenig Anlaß, die Qualität zu wechseln. Da das Gesamtsystem durch das Einsparen von Ressourcen kostengünstiger arbeiten kann, können kooperierende Empfänger einen Anteil des entstehenden Gewinns als Belohnung erhalten.

Ein Beispiel für dieses Verfahren zeigt Abbildung 4. In Bild (a) ist ein Sender dargestellt, der einen Video in zwei verschiedenen Qualitäten verschickt. Im unteren Teilbaum gibt es nur einen Empfänger ($r4$) für Qualität 2. Daher wird Agent $a3$ nun aktiv und schickt ihm einen Vorschlag zum Qualitätswechsel. Bild (b) zeigt die Situation, die eintritt, falls $r4$ den Vorschlag akzeptiert. Qualität 2 muß nun auf dem Link von $a3$ zu $r4$ nicht mehr unterstützt werden. Damit wird sie auch auf dem Link von $a1$ nach $a3$ nicht mehr benötigt. Entsprechend können nun Ressourcen freigegeben werden. Prinzipiell kann jetzt auch Agent $a1$ aktiv werden, denn er sendet Qualität 2 nur noch an $a2$. Es kann also sinnvoll sein, einen Vorschlag zum Wechsel zu Qualität 1 an $a2$ zu schicken, der dann seinerseits entscheiden muß, ob er diesen Vorschlag an seine Empfänger (in diesem Fall nur $r6$) weitergibt. Es zeigt sich, daß nicht nur direkte Nachbarn von Endsystemagenten aktiv werden können, sondern auch Netzagenten, die sich näher an der Wurzel des Multicast-Baumes befinden. Im optimalen Fall läßt sich der gesamte Unterbaum optimieren, so daß dorthin nur noch eine Qualität übertragen werden muß.

Auf eine detaillierte und formale Beschreibung der Protokolle, die die Ausführung der oben beschriebenen Management-Funktionen unterstützen, soll an dieser Stelle verzichtet werden. Nähere Informationen zu den verwendeten Nachrichtenformaten und Protokollautomaten finden sich in [9].

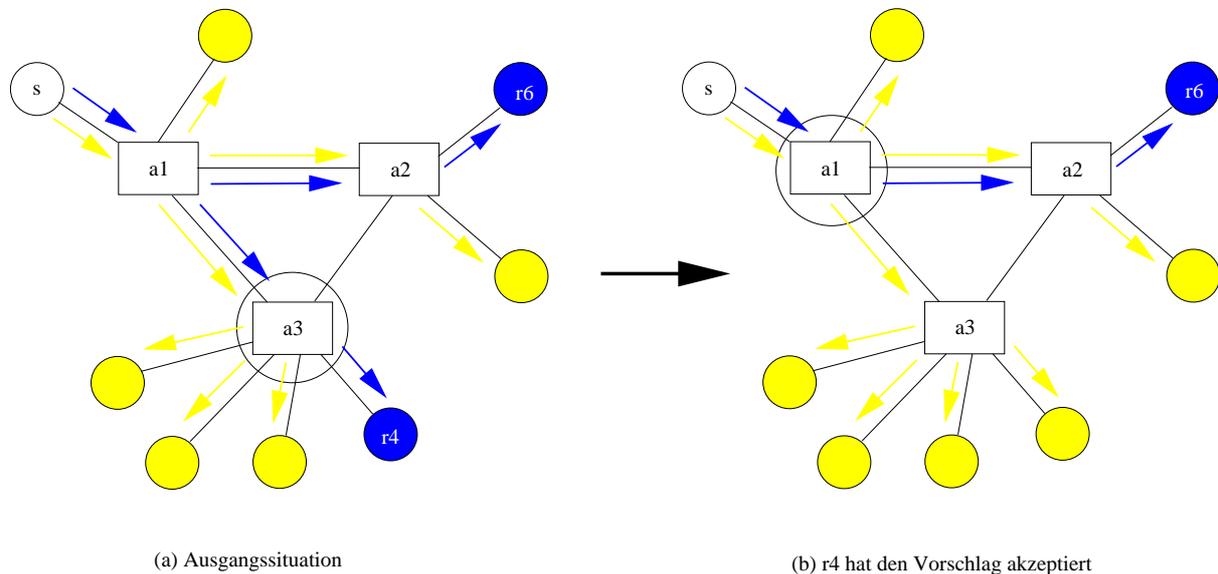


ABBILDUNG 4. QoS Neuverhandlung am Beispiel

4 Agentenstrategien

Die oben umrissenen Protokolle zwischen Agenten legen fest, welche Möglichkeiten zur Handlung einem Agenten jeweils zur Verfügung stehen. Nun stellt sich die Frage, *wann* ein Agent aktiv wird, also zu welchem Zeitpunkt er beispielsweise einen Vorschlag zum Qualitätswechsel generiert. Es muß also festgelegt werden, welche *Strategie* ein Agent anwendet. Dabei können Strategien auf leicht nachvollziehbaren Ideen wie in Abbildung 4 dargestellt basieren, indem abgezählt wird, wieviele Empfänger eine bestimmte Qualität empfangen. Sobald diese Zahl einen bestimmten Wert unterschreitet, wird der Agent aktiv und bittet die restlichen Empfänger, ihre aktuelle Qualität zu wechseln.

Oft kann es jedoch auch interessant sein, das Verhalten der Umgebung (z.B. anderer Agenten) oder das wahrscheinliche zukünftige Verhalten des Systems miteinzubeziehen. Aus diesem Grund wurden zwei neue Verfahren zur Strategiefindung entwickelt bzw. für Cooperative QoS Management angewendet.

4.1 Quality of Operation

Bei diesem Ansatz wird der aktuelle Zustand des Systems durch eine Kennzahl charakterisiert, die sich ergibt aus dem Wert der Verwendung von Ressourcen für bestimmte Ströme, den Kosten, die sich aus einer möglichen Qualitätsverletzung ergeben, sowie dem Wert freier und damit für möglicherweise in Kürze neu eintreffende Ströme verfügbare

Ressourcen. Die Kennzahl, bei der es sich um eine für die Multicast-Situation modifizierte Variante eines Ansatzes aus [13] handelt, berechnet sich wie folgt:

$$QoS = \sum_{j \in \text{streams}} \left(\alpha_j A_j - \sum_{i \in QoS} \delta_{ji} D_{ji} \right) + \sum_{t \in \text{streamtypes}} \beta_t B_t$$

mit den Parametern

- A_j , ein Maß für den Wert der Ressourcen (Bandbreite), die für Strom j reserviert wurden,
- B_t , ein Maß für den Wert der noch freien Ressourcen, die für Ströme des Typs t verwendet werden können,
- D_{ji} , ein Maß für die Kosten, die eine Qualitätsverletzung des QoS-Parameterwerts i für einen Strom j verursacht. Diese Parameter drücken den Unterschied zwischen tatsächlichen und ausgehandelten Parameterwerten aus. Wenn ein bestimmter Parameterwert nicht wie ausgehandelt eingehalten werden kann, wird ein Benutzer weniger bezahlen, so daß sich der durch die Übertragung des Stroms ergebende Gewinn verringert.
- α_j , β_t und δ_{ji} sind Steuerungsparameter, deren Wert dynamisch modifiziert werden kann:
 - ◆ α_j wird verwendet, um den Gewinn zu charakterisieren, der sich aus einer Übertragung des Stromes j ergibt; der Wert dieses Parameters wird sich typischerweise proportional zur Zahl der ausgehenden Verbindungen des Multicastbaumes für Strom j verhalten.
 - ◆ β_t beschreibt die Wichtigkeit des augenblicklichen Systemzustands, d.h. den Wert freier Ressourcen, die eingesetzt werden können, um weitere Ströme des Typs t zuzulassen.
 - ◆ δ_{ji} schließlich gibt die Wichtigkeit eines bestimmten QoS-Parameters i für den Strom j an.

Für jede mögliche Aktion, also z. B. das Freigeben von Ressourcen, weil ein Empfänger einen Vorschlag zum Qualitätswechsel akzeptiert hat, wird nun mit den sich aus der Operation ergebenden Parameterwerten eine neue Kennzahl berechnet, die den durch die Aktion erreichten Zustand kennzeichnet. Ist dieser neue Wert größer als der alte, dann sollte die Aktion durchgeführt werden, anderenfalls nicht. Gibt es mehrere Handlungsalternativen, ist die Operation mit der höchsten Wertigkeit auszuführen. Details zu diesem Konzept finden sich in [7,5].

4.2 Controlled Stochastic Petri Nets (COSTPN)

Mit Hilfe dieser neuen Variante von stochastischen Petrinetzen wird zunächst das gesamte System modelliert, wobei stochastische Schwankungen, die in jedem realen System auftreten, sehr gut erfaßt werden können. Die Neuerung in dieser Variante besteht in der Verfügbarkeit sogenannter *Rekonfigurationstransitionen*, die willkürlich geschaltet werden können. Das Schalten einer solchen Transition modelliert die Entscheidung eines Managers, das System in einen neuen Zustand zu überführen und damit besser an aktuelle Umweltbedingungen anzupassen. Es erfolgt, wenn sich ein höherer Gewinn erzielen läßt als wenn das System im alten Zustand verbliebe. Mit Hilfe numerischer Algorithmen lassen sich für ein gegebenes

COSTPN *Strategien* berechnen, die für eine bestimmte Laufzeit einen maximalen Gewinn erzeugen. Eine Strategie besteht dabei im wesentlichen aus einer Reihe von Empfehlungen, die angeben, zu welchem Zeitpunkt welche Rekonfigurationstransition geschaltet werden sollte. Diese Strategien werden nun an einen Agenten gegeben, so daß sein Verhalten entsprechend kontrolliert werden kann. Die Idee der Anwendung von COSTPN für QoS Management im allgemeinen wird in [3] beschrieben, die Anwendung speziell im Rahmen des Cooperative QoS Management kann in [5] detailliert nachgelesen werden.

5 Implementierungsaspekte

Von den oben vorgestellten Konzepten konnte bereits ein Teil in einem Prototypen des Gesamtsystems realisiert werden. In diesem Abschnitt soll zunächst das Implementierungsmodell für das Gesamtsystem beschrieben werden. Anschließend wird auf Besonderheiten beim Design der Benutzerschnittstelle auf der Empfängerseite eingegangen. Den Abschluß bildet die Vorstellung eines Anwendungsprototypen, der zeigt, wie die Konzepte des Cooperative QoS Management ihren Niederschlag in einer Teleteaching-Anwendung finden.

5.1 Das Implementierungsmodell für Cooperative QoS Management

Bei den QoS-Agenten handelt es sich um aktive Komponenten, die über das gesamte Netzwerk verteilt sind. Sie kommunizieren miteinander sowie mit diversen lokalen Komponenten, die auf den einzelnen Netzknoten verfügbar sind. Am besten wird eine solche Struktur durch sogenannte *Active Networks* [15] unterstützt. In Active Networks ist es möglich, kleine Programme dynamisch auf einen Netzknoten zu laden und bei bestimmten Anlässen auszuführen. Typischerweise ist ein solcher Anlaß das Eintreffen eines Datenpaketes, aber es ist auch möglich, solche Codestücke im Hintergrund auszuführen und sie zu aktivieren, wenn bestimmte Bedingungen erfüllt sind. Auf diese Art und Weise funktionieren auch die QoS-Agenten: sie überprüfen periodisch, ob es günstig oder notwendig ist, bestimmte QoS-Managementfunktionen auszuführen. Entscheidungen werden aufgrund der im vorherigen Abschnitt diskutierten Strategiemodelle getroffen.

Aktionen von Agenten umfassen das Versenden von Nachrichten an benachbarte Agenten entsprechend den Inter-Agenten-Protokollen aus Abschnitt 3, aber auch die Kommunikation mit lokalen Einheiten. Im hier entwickelten Modell stehen als lokale Einheiten das Ressourcenreservierungsprotokoll RSVP [17] sowie eine QoS-basierte Routing-Komponente zur Verfügung. QoS-Agenten erhalten Informationen von diesen Komponenten, aber sie geben auch Anweisungen. Ein typisches Beispiel für eine Agenten-Komponenten-Kommunikation ist die Aufforderung eines Agenten im Rahmen eines QoS-Adaptation-Prozesses an die Routing-Einheit, einen neuen Weg zu finden, der die QoS-Anforderungen erfüllt, da der alte Weg nicht mehr die gewünschte Dienstgüte erbringen kann.

Abbildung 5 zeigt das Gesamtimplementierungsmodell für Cooperative QoS Management. Die lokalen Komponenten RSVP und QoS-Routing sind fest auf den Knoten installiert, während die QoS-Agenten nur bei Bedarf geladen werden. Weitere Informationen zum Implementierungsmodell finden sich in [6].

5.2 Anwendungen und Cooperative QoS Management

Neben dem Agentenmodell stellt sich die Frage, inwiefern der Entwurf von Applikationen durch das neue QoS Management-Verfahren beeinflusst wird. Genauere Untersuchungen [8] ergaben, daß verglichen mit anderen QoS-Managementsystemen, also z.B. mit dem schon genannten Montrealer System für präsentationale Anwendungen [10], nur wenige zusätzliche Anforderungen an die Anwendung zu stellen waren. Neben verschiedenen kleineren Änderungen wird insbesondere die Benutzerschnittstelle zum QoS-Management beeinflusst. Es muß berücksichtigt werden, daß der lokale QoS-Agent die von Netzagenten generierten Vorschläge an den Benutzer weitergeben kann. Entsprechend muß der Benutzer dann die Möglichkeit haben, auf einen solchen Vorschlag zu reagieren. Außerdem wurde eine explizite Kostenangabe eingeführt, die es dem Benutzer erlaubt, Datenströme auf der Basis von Qualität und Kosten auszuwählen.

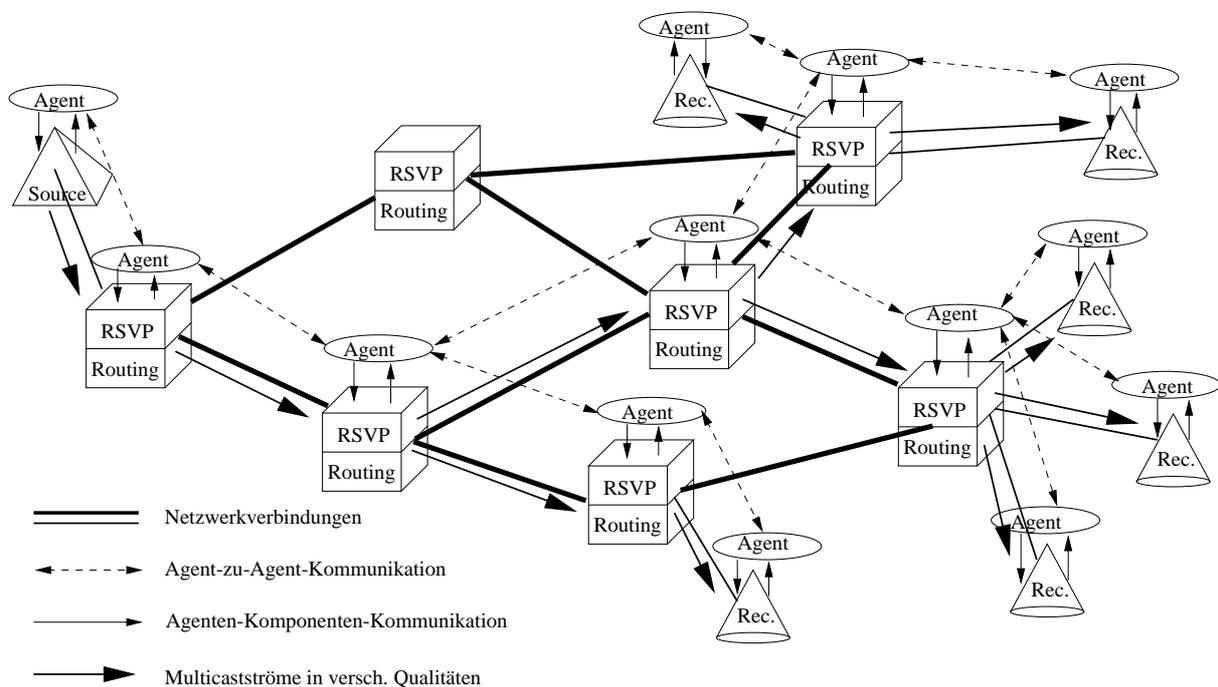


ABBILDUNG 5. Implementierungsmodell für Cooperative QoS Management

Im ursprünglichen Montrealer QoS Management-System wurde bereits eine QoS-Benutzerschnittstelle auf der Basis von X-Windows und Motif implementiert. Die Schnittstelle wurde nun entsprechend der genannten Anforderungen modifiziert und für den Gebrauch durch die neuen QoS-Agenten vorbereitet [8]. Abbildung 6 zeigt diese Schnittstelle in einer Situation, in der dem Benutzer drei Medien in jeweils unterschiedlichen Qualitäten angeboten werden. Er kann nun die gewünschte sowie weitere akzeptable Qualitäten angeben. Außerdem kann er festlegen, ob er die Möglichkeit der QoS-Neuverhandlungen nutzen will, also mögliche Angebote des QoS-Systems zum Qualitätswechsel angezeigt bekommen möchte.

Streams	Selected	Acceptable	Description	Current Cost in \$
Teacher Video	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Colored Big-screen 25 fps	7
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Colored Small-screen 25 fps	5
	<input type="checkbox"/>	<input type="checkbox"/>	B&W Small-screen 16 fps	2
Teacher Audio	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CD Quality	2
	<input type="checkbox"/>	<input type="checkbox"/>	Telephone Quality	1
Student Audio	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CD Quality	2
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Telephone Quality	1

I would like to receive special rate/quality offers during the session!

Submit Reset Fields

ABBILDUNG 6. Benutzerschnittstelle für die QoS-Auswahl

5.3 Entwicklung eines Prototypen für eine Teleteaching-Applikation

Abschließend wurde ein Prototyp einer Applikation erstellt, um die Grundidee des Cooperative QoS Management an einem Beispiel aus Anwendungssicht zu demonstrieren. Basierend auf den MBone-Tools [12] *vic* zur Multicast-Übertragung von Video und *vat* für Audio sowie der neuen QoS Managementschnittstelle wurde eine Teleteaching-Applikation zusammengestellt. Diese Anwendung wird im Rahmen des großen kanadischen Multimedia-Projekts "Broadband Services" eingesetzt und derzeit noch weiterentwickelt.

Abbildung 7 zeigt die Grundstruktur der Anwendung. Ein Dozent spricht über ein bestimmtes Thema. Audio und Video seines Vortrags werden über das Netz übertragen und können von den virtuell teilnehmenden Studenten empfangen werden. Studenten können ebenfalls aktiv werden, indem sie z.B. Fragen stellen. Allerdings wird von den Studenten nur ein Audiostrom gesendet. Bei seinem Vortrag kann der Dozent zusätzlich auf in einer Datenbank abgelegte multimediale Dokumente zugreifen und deren Übertragung an die Zuhörer veranlassen.

Als Basisfunktionen des Cooperative QoS Managements wurden QoS Verhandlung und Neuverhandlung eingebaut. In der neuen Benutzerschnittstelle werden eintreffende Vorschläge zum Qualitätswechsel eingeblendet. Akzeptiert der Benutzer einen solchen Vorschlag, dann wird das entsprechende gerade laufende MBone-Tool abgebrochen und mit der Multicast-Adresse für den Strom der neuen Qualität erneut gestartet. Damit ist der Qualitätswechsel durchgeführt. Die dazu notwendige Softwarearchitektur eines Endsystem-Knotens zeigt Abbildung 8. Der QoS-Agent erhält Informationen vom QoS-Monitor bzgl. der aktuellen Qualität, von seinem Upstream-Agenten bzgl. angebotener Datenströme und von

der QoS-Benutzerschnittstelle bzgl. eventuell neuer Wünsche oder Antworten des Benutzers auf Anfragen des Agenten. Falls nötig, kann der Agent das Stoppen eines Mbone-Tools und den Neustart mit einer neuen Multicast-Adresse veranlassen. Details zur Teleteaching-Applikation finden sich in [8].

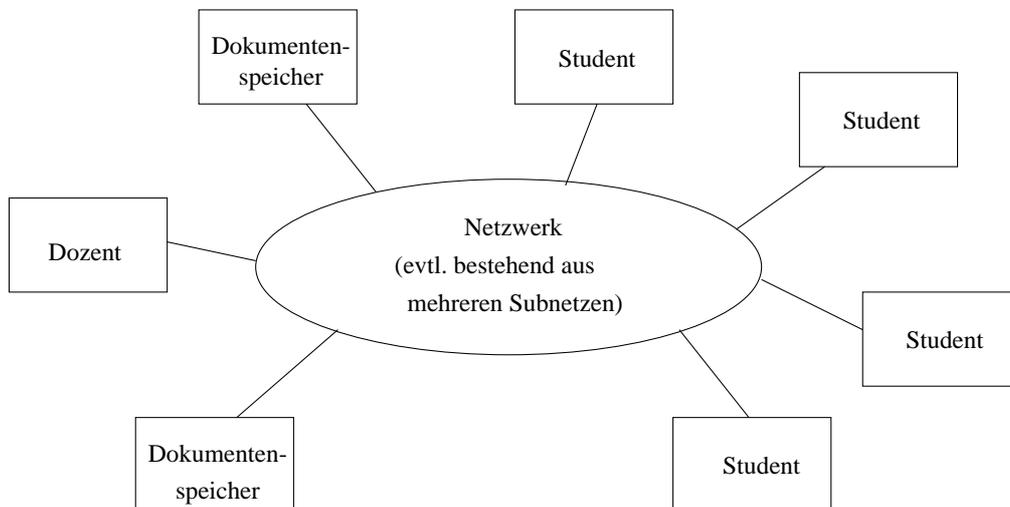


ABBILDUNG 7. Architektur der Teleteaching-Anwendung

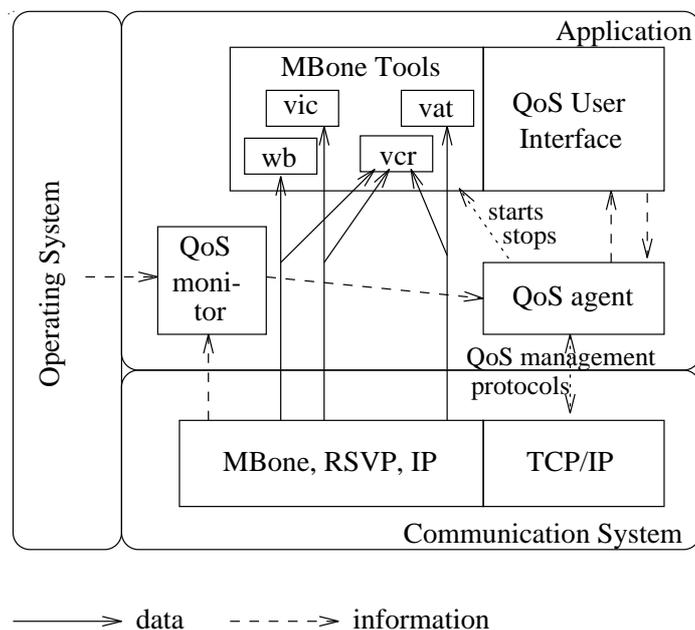


ABBILDUNG 8. Softwarestruktur eines Endsystems bei der Teleteaching-Anwendung

6 Zusammenfassung und Ausblick

Der vorliegende Artikel beschreibt ein neues System zur Unterstützung von QoS-Management, das auf der Verwendung von QoS-Agenten im Netz beruht. Aufgrund seiner

Struktur unterstützt dieses Cooperative QoS Management nicht nur die Erbringung eines einmal ausgehandelten QoS; vielmehr versucht es gleichzeitig, für eine optimale Ausnutzung der für die Multimediakommunikation verwendeten Ressourcen zu sorgen. Eine zentrale Eigenschaft des Verfahrens besteht darin, daß nur Nachbarknoten Nachrichten austauschen. Auf diese Weise wird die Überlastung bestimmter Netzknoten, also beispielsweise eines Senders vermieden, wodurch sich sehr gute Skalierungseigenschaften ergeben. Es wurden überblicksartig die Architektur des Verfahrens und detailliert die einzelnen unterstützten Managementfunktionen beschrieben. Außerdem wurde ein Rahmen für die Implementierung des Verfahrens vorgegeben.

Neben der weiteren Realisierung von QoS-Managementfunktionen und der Umsetzung des Gesamtsystems auf einem Active Network sind insbesondere die Fragen der Strategiefindung eines Agenten von besonderem Interesse. In weiteren Arbeiten soll vor allem das Modell der Controlled Stochastic Petri Nets weiter untersucht werden. Auch die quantitative Auswirkung der Verwendung unterschiedlicher Video- und Audiokodierungsverfahren wird Gegenstand zukünftiger Untersuchungen sein.

Literatur

- [1] C. Aurrecochea, A. Campbell, und L. Hauw. A Survey of QoS Architectures. *Multimedia Systems Journal, Special Issue on QoS Architectures*, Mai 1998.
- [2] A. Campbell, G. Coulson, F. Garcia, und D. Hutchison. A Continuous Media Transport and Orchestration Service. In *Proceedings ACM SIGCOMM'92*. ACM, 1992.
- [3] H. de Meer und S. Fischer. Controlled stochastic petri nets for multimedia qos management. In *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (Tagungsband MMB'97)*, Freiberg, Deutschland. VDE-Verlag, Sept. 1997.
- [4] D. Ferrari und D. C. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368-377, 1990.
- [5] S. Fischer und H. de Meer. Decision Support in Cooperative QoS Management. In R. Friedrich und E. Knightley, Hrsg., *6th International Workshop on Quality of Service, Napa, USA*, Mai 1998.
- [6] S. Fischer und H. de Meer. Using Cooperation in QoS Selection to Reduce Service Cost. In M. Merz and W. Lamersdorf, Hrsg., *Proc. of Trends in Electronic Commerce'98, Hamburg, Deutschland*. dpunkt-Verlag, Juni 1998.
- [7] S. Fischer, A. Hafid, G. v. Bochmann, und H. de Meer. Cooperative QoS Management in Multimedia Applications. In *4th IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS'97)*, Ottawa, Canada, S. 303-310. IEEE Computer Society Press, Juni 1997.
- [8] S. Fischer, M. Salem, und G. von Bochmann. Application Design for Cooperative QoS Management. In A. Campbell and K. Nahrstedt, Hrsg., *Proc. of the Fifth IFIP International Workshop on Quality of Service, New York, USA*, S. 191-194, Mai 1997.
- [9] A. Hafid und S. Fischer. A multi-agent architecture for cooperative qos management. In *Management of Multimedia Networks and Services (Tagungsband MMNS'97)*, S. 41-54. Chapman & Hall, 1998.

- [10] A. Hafid und G. von Bochmann. An Approach to QoS Management in Distributed MM Applications: Design and an Implementation. *Multimedia Tools and Applications Journal*, erscheint im Sommer'98.
- [11] D. Hehmann, M. Salmony, und H. J. Stüttgen. Transport services for multimedia application on broadband networks. *Computer Communications*, 13(4), 1990.
- [12] V. Kumar. *MBone - Interactive Multimedia on the Internet*. New Riders Publishing, Indianapolis, Indiana, 1996.
- [13] J. E. Neves, L. B. de Almeida, und M. J. Leitao. ATM Call Control by Neural Networks. In J. Alspector, Hrsg., *Proc. of the 1st Intern. Workshop on Applications of Neural Networks to Telecommunication*, S. 210-217, 1993.
- [14] N. Shacham. Multipoint communication by hierarchically encoded data. In *IEEE Infocom'92*, S. 2107-2114, 1992.
- [15] D. Tennenhouse et al. A Survey of Active Networks Research. *IEEE Communications Magazine*, 35(1):80-86, Jan. 1997.
- [16] C. Vogt, L. Wolf, R. G. Herrtwich, und H. Wittig. HeiRAT - Quality of Service Management for Distributed Multimedia Systems. *Multimedia Systems Journal*, Mai 1998.
- [17] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource Reservation Protocol. *IEEE Network*, 7(5), Sept. 1993.