

Quality of Experience von mobilen Multimediaanwendungen

Shelley Buchinger

shelley.buchinger@univie.ac.at

Abstract: Der Erfolg von mobilen Multimedia Services hängt von sehr unterschiedlichen Faktoren, wie zum Beispiel der Verfügbarkeit interessanter Inhalte, spezieller Eigenschaften der mobilen Geräte, der User Akzeptanz im Allgemeinen und natürlich von technischen Aspekten ab. Das Ziel dieser Dissertation besteht darin, Strategien für offene Probleme im Bereich mobiler Multimedia Services mit dem Fokus auf die Nutzerperspektive vorzuschlagen. Der hier gewählte Ansatz wird üblicherweise als Quality of Experience (QoE) bezeichnet und bildet die subjektive Erfahrung der Nutzer eines Ende-zu-Ende Systems ab. Auf der technischen Seite des Gebietes, wird zum Beispiel die Reduzierung der Serverbandbreite genauer überlegt und untersucht, da diese meistens den Flaschenhals eines Systems darstellt.

1 Problemstellung

Im Speziellen wird in dieser Arbeit überlegt, wie die mit Hilfe eines sogenannten Video-on-Demand-Services die zur Verfügung gestellten Filme am vorteilhaftesten an die Kunden verteilt werden können. Der einfachste Ansatz, den Nutzern die angefragten Videos zu liefern, besteht darin, jedem User den von ihm angeforderten Film einzeln mit *unicast* über einen virtuellen Kanal von b bit/s zu übertragen. Unter der Annahme, dass λ Kunden pro Sekunde Videos der Länge L anfragen, kann die Serverlast mit λL Kanälen bezeichnet werden, wobei jeder Kunde genau einen Kanal erhält. So ein System ist schwer skalierbar, da Videos viel Speicherkapazität benötigen und daher auf Festplatten gespeichert werden müssen. Die Zugriffszeiten von Festplatten sind allerdings von erheblich längerer Dauer und daher stellt die Serverbandbreite bei einer hohen Anzahl von Nutzern den Flaschenhals des Systems dar. In dieser Dissertation wird ein Verfahren entwickelt mit welchem die Serverbandbreite am effektivsten reduziert werden kann. Dabei wird besonderer Wert darauf gelegt, die vom Nutzer wahrgenommene Qualität nicht zu beeinträchtigen.

Bisher wurden bereits einige Methoden zur Reduzierung der verwendeten Serverbandbreite von Video-on-Demand-Systemen vorgeschlagen. Eine Auswahl der wichtigsten Vertreter wird in folgender Liste angegeben.

Batching: Anfragen, welche innerhalb eines fixen Zeitraums den Server erreichen, werden gebündelt. Das spart zwar Bandbreite, führt allerdings zu Wartezeiten am Anfang der Übertragung. Die Dauer der Wartezeiten hängt von der Länge des gewählten fixen Zeitraums ab [DSS94, BCLMM05].

Piggybacking: Die Datenrate neuer Anfragen wird beschleunigt und die älterer Anfragen wird verlangsamt. Somit können mehrere Übertragungen gebündelt werden. Leider kann mit dieser Methode nur wenig Serverbandbreite gespart werden, da die Datenrate um maximal 5% verändert werden kann [AWY96].

Periodic Broadcasting: Das Video wird in kleine Abschnitte unterteilt. Diese werden fortlaufend an alle Nutzer ausgestrahlt, wobei Abschnitte des Videobeginns häufiger gesendet werden. Trotz allem führt dieser Ansatz zu Wartezeiten am Anfang der Übertragung [KS03, Hu01, Bha05].

Patching: Der Hauptteil des Videos wird mit Hilfe eines Multicast-Kanals gebündelt ausgestrahlt. Der Anfang des Videos wird gleichzeitig je nach Bedarf an jeden Nutzer einzeln gesendet. Die notwendige Serverbandbreite hängt davon ab, wie weit die gebündelte Übertragung bereits fortgeschritten ist [HCS98].

Da Patching keine Wartezeiten verursacht, wurde dieser Ansatz oftmals weiterentwickelt. Einige Patchingmethoden wurden vorgeschlagen, jedoch keine erreichte die in der Theorie aufgestellten minimalen Anforderungen von $B^* = \ln(\lambda L + 1)$ Kanälen an die Serverbandbreite. Diese Patchingidee wurde in der Vergangenheit auch mit einer Cache-basierten Video-Verteilung kombiniert.

2 Ergebnisse der Arbeit

In dieser Dissertation wurde ein neues Verfahren zur Verteilung angefragter Filme eines Videos-on-Demand-Services, das *hierarchisches Patching* [HB08] genannt wird, entwickelt und vorgestellt. Es beruht auf der Annahme, dass die Kunden in der Lage sind, mehrere Kanäle gleichzeitig zu empfangen. Die zentrale Idee besteht darin, auch die Anfänge zu bündeln und über Multicastkanäle zu verteilen. Außerdem soll weiterhin Serverbandbreite gespart werden, indem einzelne Videoteile nicht sofort zum Zeitpunkt der Anfrage, sondern erst zu dem Zeitpunkt ausgestrahlt werden, an dem ein Nutzer diese Daten für eine Videodarstellung ohne Stockungen benötigt.

In der Abbildung 1 wird ein Beispiel gebracht, welches den Leser dabei unterstützen soll, die Abfolge von hierarchischem Patching nachzuvollziehen und dessen Sinn zu verstehen. Nutzer c_1 stellt zum Zeitpunkt 0 seine Anfrage an den Server und erzeugt somit einen Multicastkanal der Länge L . Der zweite Kunde c_2 fragt zum Zeitpunkt 2 das selbe Video an. Er kann sich über den vom Nutzer c_1 erzeugten Multicastkanal das Video vom Zeitpunkt 2 bis L herunterladen. Er benötigt allerdings wie beim klassischen Patching den versäumten Anfang über einen separaten Kanal, welcher Patch genannt wird. Dieser ist beim hierarchischem Patching wieder ein Multicastkanal und wird in diesem Beispiel mit $p_{2,1}$ bezeichnet. Nutzer c_3 kommt zum Zeitpunkt 3 an und hat die ersten drei Zeiteinheiten verpasst. Die zweite Zeiteinheit erhält er, indem er sich patch $p_{2,1}$ anschließt. Dem Nutzer c_2 fehlen also die erste und die dritte Zeiteinheit. Es müssen also zwei neue Patches $p_{3,1}$ und $p_{3,2}$ erzeugt werden. Der Kunde benötigt allerdings nicht beide Kanäle sofort. Patch $p_{3,2}$ wird erst in der Zukunft gebraucht, und dessen Übertragung beginnt daher erst zum Zeitpunkt 5. Die Anfrage des Kunden c_4 erreicht den Server zum Zeitpunkt

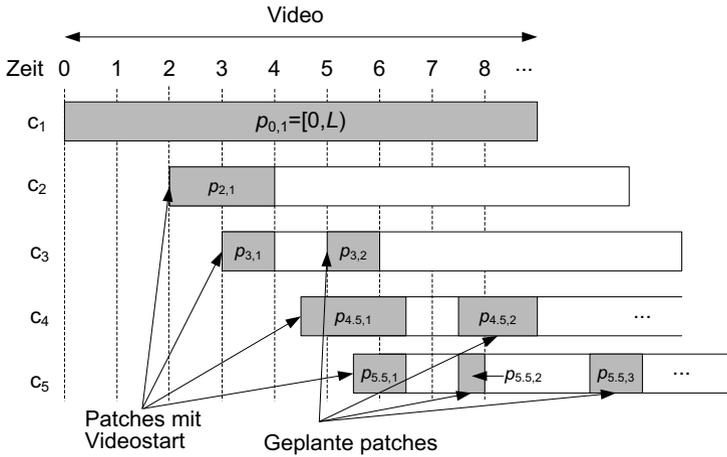


Abbildung 1: Hierarchisches Patching, ein Beispiel

4.5. Zu diesem Zeitpunkt ist nur der erste Patch $p_{0,1}$ verfügbar. Da die Übertragung von Patch $p_{3,2}$ in die Zukunft zum Zeitpunkt 5 gesetzt wurde, hat nun der Kunde c_4 noch immer die Möglichkeit, sich diesem anzuschließen. Die Serverbandbreite wird mit Hilfe von hierarchischem Patching reduziert, weil alle Patches Multicastkanäle sind und nur dann übertragen werden, wenn die Daten zum erstem Mal benötigt werden.

In dieser Dissertation konnte analytisch bewiesen werden, dass die Serverbandbreite unter Anwendung von hierarchischem Patching auf das bereits bekannte Minimum von $B^* = \ln(\lambda L + 1)$ reduziert werden kann. Außerdem werden in dieser Dissertation Ergebnisse von Simulationen präsentiert, welche dazu dienen, die Ersparnis der Bandbreite unter Verwendung von hierarchischem Patching im Vergleich zu anderen Patchingmethoden zu demonstrieren.

Wie bereits erwähnt wurde, wird bei hierarchischem Patching angenommen, dass ein Nutzer mehrere Kanäle gleichzeitig empfangen kann. In seltenen Fällen kann es in der anwendungsnahen Version des vorgeschlagenen Patching-Verfahrens, welches batched hierarchisches Patching genannt wird, vorkommen, dass diese Anzahl die Downloadkapazität des Kunden überschreitet wird. Um dies zu verhindern, wird ein spezielles Kodierungsverfahren vorgeschlagen. Es basiert auf der Beobachtung, dass der Anfang eines Videos mit hierarchischem Patching viel öfter gesendet wird als der Rest. Diese ursprünglich experimentell erkannte Eigenschaft konnte im Laufe der Arbeit analytisch bewiesen werden.

Es wurde die Wahrscheinlichkeit, dass der Zeitpunkt $x, 0 \leq x \leq L$ des Videos übertragen wird, gesucht. Dies wird durch die Dichtefunktion $f_{\lambda L}(x)$ dargestellt, welche wie folgt ausgedrückt wurde:

$$f_{\lambda L}(x) = \frac{1}{B^*} \frac{\lambda L}{\lambda L x + L}, \quad x \in [0, L], \tag{1}$$

wobei B^* die bereits oben erwähnten minimalen Anforderungen an die Serverbandbreite,

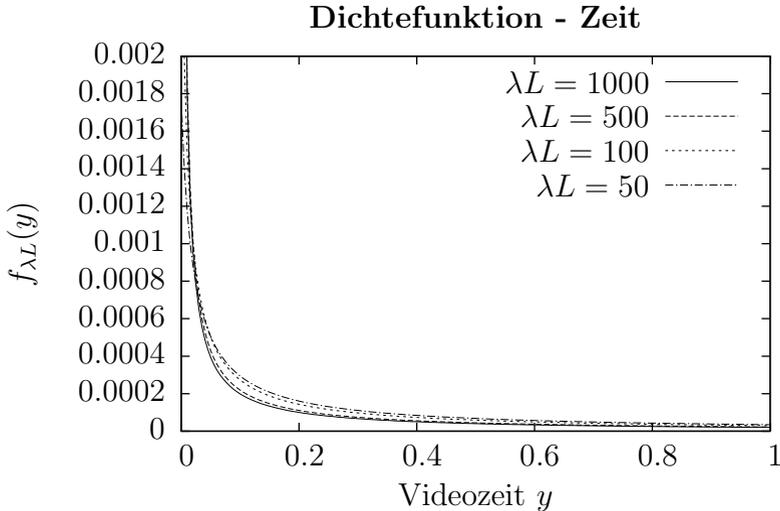


Abbildung 2: Die Dichtefunktion $f_{\lambda L}(x)$ gibt die Wahrscheinlichkeit an, dass der Zeitpunkt x , $0 \leq x \leq L$ des Videos übertragen wird.

welche mit hierarchischem Patching erreicht werden können, darstellt.

$$B^* = \ln(\lambda L + 1).$$

In Abbildung 2 kann man deutlich erkennen, dass die Werte der Dichtefunktion $f_{\lambda L}(x)$, wenn x gegen 0 geht, stark ansteigen. Es kann also behauptet werden, dass der Anfang viel öfter gesendet wird als der Rest.

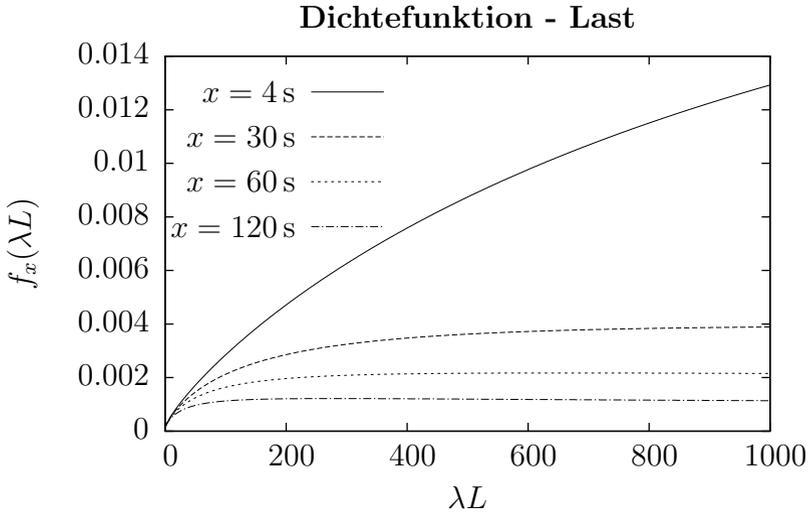
Außerdem konnte ein interessanter Zusammenhang zwischen $f_{\lambda L}(x)$ und der Menge an Kundenanfragen λL hergestellt werden. Um dies heraus zu kehren, wurde f als Funktion von λL neu definiert:

$$f_x(\lambda L) = \frac{1}{\ln(\lambda L + 1)} \frac{\lambda L}{\lambda L x + L}.$$

In Abbildung 3 wird die Wahrscheinlichkeit, dass die Zeitpunkte $x = 4, 30, 60$, und 120 s in Abhängigkeit von der Last λL für ein Video von zwei Stunden Länge übertragen werden, dargestellt. Die Werte von $f_{\lambda L}(x)$ steigen stark für kleine λL , was für das hiesige Vorhaben gut ist. Und wie erwartet ist die Steigung für kleine x besonders ausgeprägt.

Wenn man sich nun dazu entschließt, den Anfang eines Videos mit deutlich geringerer Bitrate zu kodieren als den Rest, dann kann unter Verwendung von hierarchischem Patching eine entscheidende Menge an Serverbandbreite eingespart werden. Diese Strategie wird mit *Low Start* [BH10] bezeichnet.

Wenn $0 \leq \alpha, \beta \leq 1$ und b die Kodierungsbitrate des Filmes darstellt, dann kann die Low

Abbildung 3: $P(x = 4, 30, 60, \text{ and } 120 \text{ s is transmitted})$

Start-Serverbandbreite $\hat{B}(\alpha, \beta)$ folgendermaßen aufgeschrieben werden:

$$\begin{aligned} \hat{B}(\alpha, \beta) &= \alpha b \ln(\lambda \beta L + 1) + b [\ln(\lambda L + 1) - \ln(\lambda \beta L + 1)] \\ &= b \ln(\lambda L + 1) - (b - \alpha b) \ln(\lambda \beta L + 1), \end{aligned} \quad (2)$$

wobei βL die Länge des Anfangs und αb die reduzierte Bitrate darstellen. $\hat{B}(\alpha, \beta)$ ist also auf jeden Fall kleiner als die ursprüngliche minimale Serverbandbreite $B = b \ln(\lambda L + 1)$.

In vielen Fällen ist dies auf natürliche Weise der Fall, da zu Beginn eines Filmes das Firmenlogo und/oder Credits angezeigt werden, welche eine deutlich geringere Bitrate bei der Kodierung benötigen als die meisten Videoszenen. Dies ist jedoch nicht immer der Fall. Vor allem moderne Filme beginnen nun immer häufiger mit einer Actionszene, um den Zuschauer gleich zu Beginn ins Geschehen eintauchen zu lassen und um ihn zu fesseln. In einem solchen Fall könnte sich die Reduzierung der Bitrate sehr negativ auf die Nutzerzufriedenheit auswirken.

Um sicher zu gehen, dass der vorgeschlagene Low Start-Ansatz die vom Kunden wahrgenommene Qualität auch dann nicht beeinträchtigt, wenn ein Film mit einer schwierig zu kodierenden Szene beginnt, wurde eine Nutzerstudie durchgeführt. Um sinnvolle Werte für α und β , also für die Länge des Anfangs und für das Ausmaß der Reduzierung der Bitrate zu wählen, wurden folgende Überlegungen angestellt:

Das Ziel von Low Start besteht darin, Serverbandbreite zu sparen. Daher wird als erstes darauf geachtet, α und β so zu wählen, dass ein brauchbares Ergebnis erzielt werden kann. Dafür wird die Low Start-Bandbreitenfunktion $\hat{B}(\alpha, \beta)$ normiert, indem durch die Serverbandbreite ohne Anwendung von Low Start B , also wenn $\alpha = 1$ und $\beta = 0$

sind, dividiert wird.

$$\begin{aligned}
 \bar{B}(\alpha, \beta) &:= \frac{\hat{B}(\alpha, \beta)}{\hat{B}(1, 0)} \\
 &= \frac{\hat{B}(1, 0) - b(1 - \alpha) \ln(\beta \lambda L + 1)}{\hat{B}(1, 0)} \\
 &= 1 - \frac{(1 - \alpha) \ln(\lambda L \beta + 1)}{\ln(\lambda L + 1)}, \tag{3}
 \end{aligned}$$

wobei

$$\hat{B}(1, 0) = b \ln(\lambda L + 1).$$

Jetzt kann zum Beispiel eine Reduzierung der Serverbandbreite auf $R = 0.8$ festgelegt werden. Grafisch kann dieser Fall als Schnitt zwischen der Ebene mit der Höhe 0.8 mit der von $\bar{B}(\alpha, \beta)$ erzeugten Oberfläche dargestellt werden. Das Ergebnis kann in Abbildung 4 beobachtet werden, wobei $\lambda L = 1000$ gewählt wurde, da es nur notwendig ist, die Serverbandbreite zu reduzieren, wenn viele Anfragen das System erreichen.

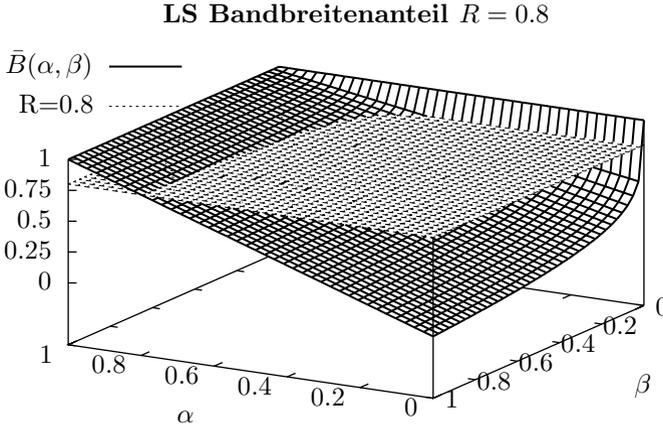


Abbildung 4: Bandbreite $\bar{B}(\alpha, \beta)$, Ebene $R = 0.8$

Wenn nun $\bar{B}(\alpha, \beta) = R$ gesetzt wird, kann β durch α folgendermaßen ausgedrückt werden:

$$\beta(\alpha) = \frac{(\lambda L + 1)^{\frac{1-R}{1-\alpha}} - 1}{\lambda L}. \tag{4}$$

Jetzt müssen nur noch Werte für α bestimmt werden und das Experiment kann aufgebaut werden. Die Auswahl der Bitraten für den Anfang der Videos richtete sich nach der

Einschätzung der vom Nutzer wahrgenommenen Qualität. Drei Qualitätsklassen wurden dafür definiert:

- Stufe 1: perfekt, keine erkennbaren Fehler (380 kbit/s, $\alpha = 1$)
- Stufe 2: gut, enthält aber Störungen (285 kbit/s, $\alpha = 0.75$)
- Stufe 3: noch akzeptabel, auch wenn das Video sichtbare Fehler enthält (247 kbit/s, $\alpha = 0.65$)

Bei einer gewünschten Serverbandbreitenreduzierung von $R = 0.8$ wurden zwei 30 Minuten lange Videos, $v_{0.65}^{0.8}$ und $v_{0.75}^{0.8}$, mit den in Tabelle 1 beschriebenen Eigenschaften erzeugt: Der obere Index von v steht für die erwünschte Bandbreitenreduzierung und der

Tabelle 1: Kodierungsparameter für Testvideos $R = 0.8$.

Video	Low Start	Bitrate kbit/s	Restdauer	Bitrate kbit/s
$v_{0.65}^{0.8}$	91 s	247	28 min 29 s	380
$v_{0.75}^{0.8}$	450 s	285	22 min 30 s	380

untere für α . Wenn R stärker reduziert werden soll, also auf 0.7, dann ergibt das ein Video, das mit $v_{0.65}^{0.7}$ bezeichnet wird und dessen Eigenschaften in Tabelle 2 angeführt werden. In diesem Fall kann kein zweites Video erzeugt werden, da das Ziel $R = 0.7$ mit der Qualitätsstufe 2 nicht einmal dann erreicht werden kann, wenn das gesamte Video mit $\alpha = 0.75$ kodiert wird.

Tabelle 2: Kodierungsparameter für Testvideos: $R = 0.7$.

Video	Low Start	Bitrate kbit/s	Restdauer	Bitrate kbit/s
$v_{0.65}^{0.7}$	670 s	247	18 min 50 s	380

Diese Videos wurden nun sowohl mit objektiven Methoden der Videoqualitätsmessung [DAO08], welche als Software zur Verfügung standen, als auch subjektiv mit Hilfe einer Nutzerstudie [ITU02] untersucht. Alle angewandten Methoden lieferten das gleiche Ergebnis:

Es konnte festgestellt werden, dass die Nutzer einen kurzen Anfang mit geringerer Bitrate gegenüber einer leichteren und dafür länger andauernden Störung bevorzugen.

Außerdem stellten die Nutzer keine Qualitätseinbußen bei Videos mit kurzen Anfängen fest. Man muss allerdings erwähnen, dass das Ergebnis der Nutzerstudie nicht in allen Fällen signifikant war. Daher galt es weitere Argumente zu finden, welche dieses Ergebnis bestätigen können.

Nach weiteren Beobachtungen konnte festgestellt werden, dass die Dateigrößen der Filme sich nach den gewählten Eigenschaften der Anfänge richteten. Genauer gesprochen, stellte

sich heraus, dass die Dateigröße von einem Video mit kurzem Low Start und geringerer Bitrate größer ist als die eines längeren und dafür moderateren Anfangs. Diese Eigenschaft würde eine stichfeste Begründung für das Ergebnis der objektiven Qualitätsmessung und der Nutzerstudie liefern. Sie bedeutet nämlich, dass Videos mit kurzen und schlechten Anfängen insgesamt mehr Bitrate zur Verfügung steht, als solchen mit besseren, aber dafür längeren Anfängen. Dies muss sich förmlich positiv auf die Gesamtqualität der Videos auswirken.

Daher galt es einen analytischen Beweis zu finden, der diese Beobachtung bestätigt und das Ergebnis der Studie untermauert. Die Dateigröße $s(\alpha)$ für eine gegebene Bitrate b , Videolänge L , Last λL und einer gewünschten Bandbreitenreduzierung R kann ausgedrückt werden als:

$$s(\alpha) = \beta(\alpha)L \cdot \alpha b + (1 - \beta(\alpha))L \cdot b, \quad (5)$$

wobei $\beta(\alpha)$ durch (4) gegeben ist.

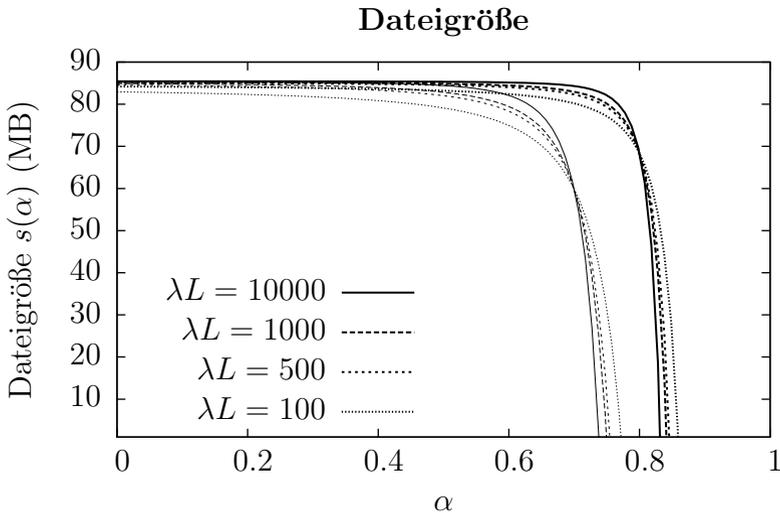


Abbildung 5: Video file size $s(\alpha)$, $R = 0.8$, $R = 0.7$

Das Verhalten der Funktion $s(\alpha)$ kann in Abbildung 5 beobachtet werden. Hier kann nun festgestellt werden, dass die Dateigröße mit steigendem α sinkt. Demnach ist die gesamte Datei kleiner, je höher die Bitrate des Anfangs ist und in Folge auch je länger dieser dauert.

Insgesamt konnte nun gezeigt werden, dass ein kurzer Low Start mit geringerer Bitrate einem längeren zu bevorzugen ist.

Um die Qualität von Video-on-Demand-Services speziell für Mobiles zu verbessern, wurde eine weitere Nutzerstudie zur Bestimmung der optimalen Framerate für verschiedene Handy-TV Inhaltstypen durchgeführt. Es stellte sich heraus, dass statt 25 oder 30 Bildern

pro Sekunde für Inhalte wie Nachrichtensendungen und Comics die Einstellung von 5 Bildern pro Sekunde bevorzugt wird. Die optimale Einstellung für Inhalte mit mehr Bewegung, wie zum Beispiel einem Fußballmatch, liegt auch nur bei 10 Bildern pro Sekunde.

3 Methodischer Ansatz der Arbeit

Das Spezielle dieser Arbeit besteht in der Kombination drei sehr unterschiedlicher Forschungsmethoden. Es wurde für die Berechnung der notwendigen Serverbandbreite ein analytisches Modell entwickelt. Eine Simulation wurde programmiert, um das vorgeschlagene Verfahren mit bereits existierenden Methoden besser vergleichen zu können. Zuletzt wurden Nutzerstudien durchgeführt, um die realen Auswirkungen der Kodierungsmethode "Low Start" auf die Nutzerzufriedenheit zu messen und die optimale Framerate zu bestimmen.

4 Anwendungsbereiche der Ergebnisse

Die Ergebnisse dieser Dissertation können einerseits für eine effizientere und adäquatere Kodierung von Videos im mobilen Bereich verwendet werden. Zum Beispiel wird Low Start bereits von IPTV-Anbietern angewendet. Die Nutzerzufriedenheit könnte gesteigert werden, wenn die Framerate von Videos, die am Handy konsumiert werden sollen, bei der Kodierung stark reduziert wird. Andererseits können die Ergebnisse für eine effiziente Medienverteilung eines Video-on-Demand-Systems verwendet werden.

5 Relevanz der Ergebnisse für die Medienproduktion oder -distribution

Für die AV-Medienproduktion ist das Low Start-Konzept sehr wichtig. Mit dem Wissen im Hinterkopf, dass ein Videobeginn, der wenig Bitrate benötigt, für die Verteilung der Medien von Vorteil ist, wird man eventuell Anfangsszenen, die sehr aufwändig zu kodieren sind, eher vermeiden. Natürlich ist es wichtig, den Zuseher gleich zu Beginn mit einem interessanten Inhalt zu fesseln. Dies sollte zumindest während der ersten Sekunden nicht mit einer sehr bewegten, aktiven Szene, wie zum Beispiel einer Explosion, geschehen. Auch die Erkenntnis, dass die Handy-TV-Nutzer eine geringere Bitrate bevorzugen, sollte auf die Parametereinstellungen bei der Kodierung einen Einfluss haben. Für die Bereitstellung eines Video-on-Demand-Systems könnte bei der AV-Mediendistribution überprüft werden, ob der aktuell verwendete Ansatz verbessert werden könnte.

Literatur

- [AWY96] C.C. Aggarwal, J.L. Wolf und P.S. Yu. On Optimal Piggyback Merging Policies for Video-On-Demand Systems. In *Proc. 1996 ACM SIGMETRICS Conf. Measurement and Modeling of Computer Systems*, 1996.
- [BCLMM05] G. Boggia, P. Camarda, L-Mazzeo und M. Mongiello. Performance of Batching Schemes for Multimedia-on-Demand Services. *IEEE Transactions on Multimedia*, 7(5):920–931, 2005.
- [BH10] Shelley Buchinger und Helmut Hlavacs. Optimal server bandwidth for mobile video on demand. *Annales des Télécommunications*, 65(1-2):31–46, 2010.
- [Bha05] M.S. Bhatia. Broadcasting Protocols. In *Proceeding of Parallel and Distributed Computing and Networks (PDCN 2005)*, Innsbruck, Austria, Feb. 2005.
- [DAO08] D.Vatolin, A.Moskvin und O.Petrov. MSU Video Quality Measurement Tool. http://compression.ru/video/quality_measure/video_measurement_tool_en.html, December 2008.
- [DSS94] A. Dan, D. Sitaram und P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In *ACM Multimedia*, Seiten 391–398, Okt. 1994.
- [HB08] Helmut Hlavacs und Shelley Buchinger. Hierarchical Video Patching with Optimal Server bandwidth. *ACM Transactions on Multimedia Computing Communications and Applications (TOMCCAP)*, 4(1), 2008.
- [HCS98] K.A. Hua, Y. Cai und S. Sheu. Patching: A Multicast Technique for True Video-on-Demand Services. In *ACM Multimedia'98*, Seiten 191–200, Bristol, UK, Sept. 1998.
- [Hu01] A. Hu. Video-on-Demand Broadcasting Protocols: A Comprehensive Study. In *Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, Jgg. 1, Seiten 508–517, Anchorage, AK, USA, 2001.
- [ITU02] ITU-R BT.500-11. Methodology for the subjective assessment of the quality of television pictures. ITU-R BT.500-11, 2002.
- [KS03] T. Kameda und R. Sun. Survey on VOD broadcasting schemes. Bericht, School of Computing Science, SFU, August 2003.



Shelley Buchinger absolvierte 1999 ihr Mathematik Diplom-2008 ihr Mathematik/Informatikmanagement Lehramts- und 2010 ihr Doktoratsstudium der technischen Wissenschaften an der Universität Wien. Im Rahmen des Erasmusprogramms übersiedelte sie nach Rom, wo im Anschluss an ihr erstes Studium in der Industrie ihre berufliche Karriere begann. Heute ist sie als post-doc Assistentin in der Forschungsgruppe Entertainment Computing der Universität Wien tätig, wo sie ihre Arbeit auf die Themengebiete QoE, mobile Multimediaanwendungen, Nutzer-generierte Inhalte und hybride Qualitätsmodelle konzentriert. Im März 2011 wurde Shelley Buchinger der Doc.Award 2010, eine Auszeichnung der Stadt Wien für ihre wissenschaftliche Tätigkeit im Rahmen der Dissertation verliehen.